**Honey File System for Insider Threat**

**1Priya M , 2Mrs Revathi**

1Student Department of Computer Science,Rathinam College of Arts and Science,Coimbatore,Tamilnadu,India

2Assistant Professor , Department of Computer Science,Rathinam College of Arts and Science,Coimbatore,Tamilnadu,India

 **ABSTRACT**

 Insider threats pose a significant risk to organizational security, often bypassing traditional perimeter-based defenses. This paper proposes a novel Honey File System (HFS) designed to detect, deter, and analyze insider threats by leveraging deceptive technologies. The system integrates strategically placed honey files—decoy documents that appear legitimate but contain traceable, non-sensitive content—within an organization’s file infrastructure. When accessed, modified, or exfiltrated, these files trigger real-time alerts and logging mechanisms, enabling early detection of malicious intent. The HFS not only aids in identifying unauthorized access patterns but also serves as a forensic tool for post-incident analysis. By combining deception with active monitoring, the Honey File System offers a lightweight, non-intrusive, and cost-effective approach to strengthening internal security posture. Experimental evaluation demonstrates its effectiveness in capturing anomalous behaviors without disrupting normal user operations, making it a practical enhancement to existing cybersecurity frameworks.

 **INTRODUCTION**

 Organizations now have to deal with a growing number of security dangers current digital environment, both from inside their own trusted networks and from outside entities. Insider threats are a serious risk to data integrity, confidentiality, and overall cybersecurity, regardless of whether they are intentional or unintentional. Insiders with lawful access can execute subtle, unauthorised operations that are difficult for traditional security procedures to identify.

Honey File Systems provides a creative and proactive solution to this problem. A deception-based security technique called a "honey file system" deliberately introduces phoney files or directories, or "honeyfiles," into a network. Potential insider threats are drawn to these files because they seem to contain important or sensitive information. Unauthorized access or alteration of these files may be reported as questionable activity, setting off alarms.

 These files can be manipulated or accessed without authorisation, which can be reported as suspicious activity to set off alarms and aid in early discovery.

The design and implementation of a honey file system with the express purpose of identifying and reducing insider risks is examined in this study. By combining behavioural monitoring and deception technologies, the system serves as an early warning system, assisting organisations in spotting possible dangers before actual harm is done.

 **3.METHODOLOGY**

 **Design of the System**

|  |  |  |
| --- | --- | --- |
|  **Web Interface** |   |   |



**(Left Panel) Sidebar:**

**includes links for navigation such as:**

The dashboard

File Manager for Honey

User Behaviour

Warnings

Clear labels and simple iconography

"Active Alerts"

"Total Honey Files"

"Suspicious Accesses"

**Lists or Tables:**

displaying the history of alerts or recent file access logs

File names, user IDs, timestamps, and severity levels are all stored in columns.

**Style of Design:**
 **Colour Scheme:** Accent colours (often red or orange for alerts) on a neutral white or grey backdrop

**Typography:** readable, clear sans-serif fonts

**Visual Hierarchy**: The dashboard's layout prioritises the visibility of important events, such as notifications.

**Functional Emphasis:**

 **Designed to help security administrators:**

Track the interactions of the fake files

Recognise questionable activity quickly

Effectively access user activity logs

By distributing fake files, or "honey files," among systems under observation, the proposed Honey File System (HFS) is intended to identify, evaluate, and lessen insider risks. The basic tenet is that, in most cases, authorised users shouldn't access or alter these data. Any contact with a honey file is regarded as questionable and recorded for additional examination.
 **Parts:**

**Honey File Generator**: Produces realistic-looking spoof documents automatically (e.g., confidential reports, phoney credentials).

**Deployment Manager:** Distributes honey files strategically over cloud environments or user-accessible directories.
 A small daemon or background process that records access events to honey files is called a monitoring agent.

**Alert and Logging System:** Notifies administrators of possible threats by integrating with internal dashboards or SIEM.

**Honey File Creation Strategy:** Use socially engineered filenames (e.g., "Admin\_Passwords.txt," "HR\_Salaries\_2024.xlsx").

**File Types:** To make your content more credible, use widely used file formats like PDFs, DOCX, XLSX, and TXT.

**Content Simulation:** Use scripts or data generating tools to insert syntactically accurate but fake information into files.

**Watermarking:** Use metadata or concealed tracking information (such canary tokens) to verify unapproved distribution.

1. **Environment for the Deployment Model:** The HFS is set up in a regulated, enterprise-style setting with authentic user profiles and access controls.

**Placement Strategy**: Files are arranged according to role and privilege level in high-interest directories (such as shared drives, downloads, and home directories).

**Access Control:** To guarantee that the intended users may open the files, they are given permissive access privileges.

**4. Tracking and Recording Access Triggers:** Event logs are triggered by any read, write, open, or delete operations on honey files.

**Logging Metadata:** Records the timestamp, file accessed, username, hostname, and action taken.

**Behaviour profiling**: Monitors recurring patterns of access, links them to user roles, and highlights irregularities.
2. **Metrics for Evaluation**
 The following indicators are monitored to assess the HFS's efficacy:

The ratio of real insider threats found to false positives is known as detection accuracy.

 The amount of time between file access and alert creation.

**False Positive Rate:** Inaccurately reported incidents.

**User Transparency:** The extent to which users are not aware that there is a honey file.

**System overhead:** Effect on monitored endpoints' performance.
3. **Validation via Experiments**
 Configure the testbed by simulating an enterprise network with different roles (financial, HR, developers, and administrators).

**Adversarial Scenarios:** Model insider threat actions (such as stealing credentials, abusing privileges, or gaining access out of curiosity).

**Comparative Analysis:** Evaluate performance against alternative techniques for detecting insider threats (optional).

**User Study (optional):** Determine if users are able to differentiate between authentic and honey files.

**RESULTS AND DISCUSSION**

**1. System Implementation Results**

#### **The Honey File System was successfully implemented using Python with the watchdog library. The system achieved the following key functionalities:**

|  |  |
| --- | --- |
| **Feature** | **Result Description** |
| **Honey File Creation** | **Created 3 decoy files with realistic filenames and bait content.** |
| **File Access Monitoring** | **Detected file modifications and deletions in real-time.** |
| **Alert Logging** | **Logged all suspicious interactions into alerts.log.** |
| **Real-Time Detection** | **Triggered alerts almost instantly (< 1s delay) upon file access.** |
| **Ease of Deployment** | **Lightweight setup with no external dependencies.** |

**Discussion**

**Accuracy of Honey File System for Insider Threat**

**False Positive and True Positive Results**

During the testing phase, the system reliably generated alarms and precisely recorded them when simulated insider activities were conducted by accessing or altering honey files. These accurate identifications, which are categorised as true positives, help the system discover possible insider threats.

On the other hand, very few false alerts (false positives) were observed during typical usage scenarios by authorised users. This is due to the fact that honey files were purposefully positioned in directories that ordinary users would not visit while performing their daily duties. As a result, accidental access was rare, which improved the system’s overall accuracy.

**Factors Affecting Accuracy Several factors impact the accuracy of the system:** The placement and naming of honey files The ability of the monitoring tool to capture all types of file access events Whether alerts are triggered and logged correctly The absence of user identity tracking, which limits the system’s forensic depth Future Improvements for Accuracy To enhance accuracy, future versions of the system could incorporate: OS-level access auditing tools for deeper monitoring Integration with user authentication systems to associate access events with specific users Behavioral analytics to further reduce false positives

1. **CONCLUSION**

The Honey File System developed in this project provides an effective and proactive approach to **detecting insider threats** within an organization's digital environment. By deploying strategically crafted **honeyfiles** and monitoring access to them, the system can **identify unauthorized or suspicious activity** from internal users who may otherwise go unnoticed by traditional security measures. The project successfully demonstrates how **deception-based security techniques** can enhance internal threat detection without impacting normal user operations. This approach not only strengthens overall cybersecurity posture but also provides valuable logs and insights for further **incident response and forensic analysis**. Moving forward, the system can be scaled and integrated with broader security infrastructure to provide continuous monitoring and protection against evolving insider threats

**REFERENCES**

C. Bowen, S. Hershkop, S. J. Stolfo. 2009. "Decoy Document Deployment for Effective Insider Threat Detection." Columbia University Technical Report. <https://www.cs.columbia.edu/>

A. Salem, S. Hershkop, S. Stolfo. 2008. "A Survey of Insider Attack Detection Research." Insider Attack and Cyber Security. Springer. <https://link.springer.com/>

D. Spitzner. 2003. "Honeypots: Catching the Insider Threat." In Proceedings of the 19th Annual Computer Security Applications Conference. <https://ieeexplore.ieee.org/>

K. Voris, J. McCoy, M. Bishop, R. Levitt. 2013. "Lost: Modelling Cyber-Deception in a Game-Theoretic Framework." New Security Paradigms Workshop. <https://dl.acm.org/>

J. Kim, A. Kopylov, P. B. Durgin. 2019. "Honey Files and Honeytokens for Insider Threat Detection." International Journal of Cybersecurity Intelligence & Cybercrime. <https://vc.bridgew.edu/ijcic/>

M. Almseidin, M. Alzubi, S. Kovacs, M. Alkasassbeh. 2017. "Evaluation of Machine Learning Algorithms for Intrusion Detection System." Procedia Computer Science. <https://www.sciencedirect.com/>

T. Peng, C. Leckie, K. Ramamohanarao. 2007. "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems." ACM Computing Surveys. <https://dl.acm.org/>

C. Gates, C. Taylor. 2007. "Challenging the Anomaly Detection Paradigm: A Provocative Discussion." NSPW. <https://dl.acm.org/>

P. Mell, T. Grance. 2011. "The NIST Definition of Cloud Computing." NIST Special Publication. <https://nvlpubs.nist.gov/>

N. Virvilis, D. Gritzalis. 2013. "The Big Four – What We Did Wrong in Advanced Persistent Threat Detection?" IEEE ISSE. <https://ieeexplore.ieee.org/>