D.K. Naresh, 2025, 13:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

SPYUSB: Securing USB Drives Against Malware Injection and Data Exfiltration

D.K. Naresh[1], Dr.S.Nagasundaram[2]
[1]Master of Computer ApplicationStudent, [2]Assistant Professor
Vels Institute of Science Technology and Advanced Studies
[1]apnaresh11@gmail.com, [2]snagasundaram.scs@vistas.ac.in

Abstract- Portable storage devices such as USB drives, external hard drives, and memory cards are widely used for data transfer and storage due to their convenience and portability. However, these devices are highly vulnerable to covert data theft, particularly through malware injection attacks that can silently exfiltrate sensitive information while evading traditional security mechanisms. Existing solutions typically focus on either malware detection or data backup in isolation, lacking a comprehensive defense strategy. This paper presents spyUSB, an integrated security framework that combines Deep Neural Networks (DNNs) for detecting malware activity, Cloud Conceal for secure data backup and recovery, and Data Masking through Tokenization to protect sensitive content on USB drives. The DNN component identifies malicious behaviour by analysing system-level indicators such as API calls, byte sequences, and log metadata. Upon detection of an attack, sensitive data is automatically encrypted and backed up via Cloud Conceal. Concurrently, tokenization techniques mask data on USB devices, ensuring confidentiality even in case of unauthorized access. The proposed system enhances data security and integrity, providing a holistic defense against stealthy data exfiltration attacks.

Keywords -USB security, malware injection, data exfiltration, deep neural networks (DNNs), data masking, cloud backup, tokenization, cybersecurity, portable storage protection.

I. INTRODUCTION

Universal Serial Bus (USB) storage devices have become ubiquitous in both personal and enterprise environments due to their portability, ease of use, and large storage capacities. However, this widespread adoption has also introduced significant security vulnerabilities. USB devices are increasingly exploited as vectors for covert data theft and malware injection, posing a severe threat to data confidentiality and system integrity.

Malware injection attacks via USB drives typically involve the insertion of malicious code into a host system, often without the user's awareness. These attacks are difficult to detect using traditional security mechanisms, as they frequently operate stealthily by exploiting system processes, manipulating files, or accessing sensitive resources. Once a system is compromised, the malware can exfiltrate confidential information such as login credentials, personal files, or proprietary business data, potentially leading to significant financial and reputational damage.

Existing solutions often focus on one aspect of USB security—either malware detection or data recovery—without addressing the full lifecycle of an attack. This fragmented approach leaves systems vulnerable to sophisticated threats that require

multi-layered defenses. Moreover, few systems offer proactive protection by securing sensitive data even after a breach has occurred.

To address these challenges, this work proposes spyUSB, an integrated security framework that combines malware detection, secure data backup, and data masking techniques. The core components of the system include a Deep Neural Network (DNN) for real-time detection of malicious activities, Cloud Conceal for encrypted cloud-based backup and recovery, and tokenization-based data masking to ensure sensitive information remains protected on USB devices. By leveraging these technologies, spyUSB aims to deliver a holistic defense mechanism against covert data exfiltration, thereby enhancing the resilience and trustworthiness of USB-based data transfer.

II. AIM & OBJECTIVES

The aim of this work is to design and implement an integrated security framework, spyUSB, to detect, prevent, and mitigate malware injection and data exfiltration attacks through USB storage devices, thereby ensuring the confidentiality, integrity, and availability of sensitive data.

Objectives:

1. To detect malware injection through the use of a Deep Neural Network (DNN) model trained on system-level indicators such as API calls, byte sequences, and process metadata.
2. To enable secure data backup and recovery by integrating Cloud Conceal, a cloud-based encrypted storage solution that ensures data availability even during security breaches.
3. To protect sensitive data stored on USB devices using data masking techniques based on tokenization, ensuring unauthorized users cannot interpret the original content. To implement and compare multiple classification algorithms, including Decision Trees, Random Forests, and Naive Bayes.
4. To monitor USB device activity in real time and trigger security actions upon detection of suspicious behavior.
5. To provide an integrated platform that combines malware detection, data protection, and secure backup in a unified framework for enhanced cyber defense.

III. LITERATURE SURVEY

The use of USB drives as an attack vector has received significant attention in the field of cybersecurity. Traditional malware detection techniques such as signature-based approaches

1. Offer limited protection against zero-day and polymorphic threats. To overcome this, anomaly-based detection methods have been proposed
2. Which monitor behavioral patterns like unusual system calls or unauthorized file access. However, these methods are prone to false positives and often lack adaptability. Recent advancements have introduced machine learning and deep learning techniques for malware detection.
3. A system-call sequence-based approach using Support Vector Machines (SVM) demonstrated improved detection accuracy. More sophisticated models such as Deep Neural Networks (DNNs) and Long Short-Term Memory (LSTM) networks have been used
4. Model temporal behaviour in malware activities, showing robustness against obfuscation techniques. For data protection, cloud-based secure backup mechanisms have been explored
5. Providing encrypted storage and recovery solutions in the event of data exfiltration. However, few systems integrate detection and protection in a single framework, which is the core novelty of the proposed spyUSB system.

The proposed spyUSB system addresses this gap by offering a comprehensive solution that detects malware in real-time using DNNs, protects sensitive data through tokenization, and ensures recoverability via encrypted cloud backup, thereby significantly enhancing USB security.

IV. PROPOSED SYSTEM

The proposed approach for malware detection and prevention integrates advanced techniques to address the limitations of existing systems. This solution combines the power of Deep Neural

Networks (DNNs) for malware detection, a cloud-based backup and recovery system (CloudConceal), and data masking algorithms to protect sensitive information.

- Deep Neural Networks (DNNs): The core of the approach is the spyNet model, a DNN-based malware detection system. It classifies system activities, such as process creation and file access, by analyzing features such as API calls, byte sequences, and metadata from system logs. The model is trained on labeled datasets to distinguish between benign and malicious activities. This allows the system to identify new and sophisticated malware that might evade traditional signature-based detection systems.
- CloudConceal (Backup and Recovery): Once malware is detected, the sensitive data is transferred to a secure backup system, CloudConceal. This ensures that encrypted copies of sensitive data are stored safely in the cloud. The system allows easy recovery in case of data loss or breach, minimizing downtime and ensuring business continuity.
- Data Masking: Data masking is applied to sensitive information stored on portable storage devices like USB drives. Using techniques such as tokenization, the sensitive data is obfuscated to prevent unauthorized access or theft, even if the device is compromised by malware. This ensures that even in the event of a malware attack, the exfiltrated data is rendered useless to the attacker.
- Alert Generation: The system generates real-time alerts when a potential malware attack is detected. These alerts notify system administrators and users of the threat, enabling swift action to mitigate the damage. Alerts also provide insights into the nature of the attack, such as the type of malware detected, helping in the identification and prevention of similar future attacks.

This integrated approach offers a comprehensive solution to covert data theft, combining proactive malware detection, secure cloud storage, and effective data protection strategies. By incorporating machine learning and advanced security measures, the system provides a robust defense against evolving malware threats while ensuring the confidentiality, integrity, and availability of sensitive data.

V. METHODOLOGY

The The spyUSB system brings together several technologies to detect USB-based threats and protect sensitive data in a seamless and efficient way. It works through five key steps:

Detecting Malware with Deep Learning

The system uses a Deep Neural Network (DNN) to monitor system behavior triggered by USB devices. It looks at things like API calls, file access, and process creation to spot any signs of malicious activity. The model is trained to recognize patterns that are common in malware attacks.

Backing Up Data Securely (CloudConceal)

If a threat is detected, the system immediately encrypts important files and sends them to a secure cloud storage service called CloudConceal. This way, the data remains safe and can be recovered later—even if the local system is compromised.

Masking Sensitive Data

To prevent data leaks, spyUSB uses tokenization. It replaces actual sensitive data on USB drives with fake—but structured—tokens. Even if someone accesses the device, they won't be able to read the real data.

Monitoring USB Devices

The system constantly watches for new USB devices being plugged in. It uses tools like PyUSB and PyWinUSB to detect device activity and trigger analysis when needed.

Controlling Access with Authentication

All sensitive operations—like accessing backups or reversing tokenized data—are protected with user authentication. The system uses JWT and OAuth standards via Flask to ensure only authorized users can gain access.

VI. TOOLS AND TECHNOLOGIES USED

The spyUSB system was developed using a range of modern technologies that support deep learning, data protection, USB monitoring, and secure web services:

Programming Language

- Python 3.9 – Primary language for system development, due to its rich ecosystem for machine learning, security, and system-level libraries.

Malware Detection

• TensorFlow/Keras – For building and training the Deep Neural Network (DNN) used in malware detection.
• Scikit-learn – Used for preprocessing, feature extraction, and baseline machine learning models.

Data Masking

• PyCryptodome – For data encryption and cryptographic operations.
• Faker – To generate synthetic data for tokenization.
• Pandas – For data manipulation and token mapping.

USB Device Monitoring

• PyUSB – For cross-platform USB interface detection and communication.
• PyWinUSB – For USB device event monitoring on Windows systems.

Backup and Cloud Security

• CloudConceal (Custom Implementation) – A secure, encrypted cloud backup mechanism integrated with the system to ensure data recoverability.

User Authentication

• Flask – Lightweight web framework used to build REST APIs.
• JWT (JSON Web Tokens) – For stateless authentication.
• OAuth 2.0 – To manage secure access and authorization.

Database

• MySQL – Used to store user credentials, token mappings, and activity logs.

VII. SYSTEM WORKFLOW (FLOWCHART)

The spyUSB system operates through a structured workflow that ensures both proactive and reactive security for USB-based data threats. The workflow consists of the following sequential stages:

1. USB Device Detection

When a USB storage device is inserted into the system, the USB monitoring module (using PyUSB/PyWinUSB) detects the device and initiates security checks.

2. Behavioral Analysis for Malware Detection

The system begins monitoring real-time system activities triggered by the USB device. These include file access, process execution, API calls, and byte patterns. This data is passed to a Deep Neural Network (DNN) trained to detect malicious behavior.

3. Malware Detection Response

• If malicious activity is detected, the system immediately initiates protective actions:
• Sensitive files are encrypted and backed up to a secure cloud storage (CloudConceal).
• All USB activity is logged for further inspection.
• If no threat is found, normal operations continue, but monitoring persists in the background.

4. Data Masking via Tokenization

For additional protection, any sensitive data on the USB drive is masked using tokenization. This means real values are replaced with fake, yet structured, placeholders that are meaningless without authorized access.

5. Access Control and Authentication

All critical operations—such as accessing backup data or decrypting masked information—require user authentication using secure JWT and OAuth mechanisms via Flask APIs.

6. Audit Logging and Alerts

All significant events, including USB insertions, detected threats, and backup actions, are logged in the MySQL database. Alerts can be generated for system administrators as needed.

VIII. RESULTS & DISCUSSION

The spyUSB system was tested to see how well it could detect malware, protect sensitive data, and back it up securely—especially during USB-related threats. Here's how it performed across different areas:

A. How Well It Detects Malware

The Deep Neural Network (DNN) used in spyUSB was trained to spot suspicious behavior when a USB device is plugged in. It looked for things like strange file access patterns and unusual system activities. After testing it on various examples of both normal and malicious behavior, the model achieved:

- Accuracy: 96.2% – It correctly identified threats most of the time.
- Precision: 94.8% – Very few false alarms.
- Recall: 95.5% – It didn't miss many threats.
- F1-Score: 95.1% – A balanced score for overall performance.

These results show that the system can reliably catch malware that tries to sneak in through USB devices.

B. How Fast It Responds

From the moment a USB device is plugged in, it takes around 1.8 seconds for the system to detect potential threats and start securing data. This quick response is fast enough to act before any real damage happens, which is crucial for real-time protection.

C. How It Protects Data with Masking

To test data masking, the system replaced sensitive information in files with fake but realistic-looking data (using tokenization). Even if someone stole the USB drive, they would only see the fake data—not the real thing. Only verified users could reverse the process and see the original information, which worked exactly as planned.

D. How Reliable the Backup Is

Every time a threat was detected, spyUSB automatically backed up the critical data to secure cloud storage. In every test case, backups were created successfully and could be recovered without any issues—proving the system's reliability in protecting and restoring important files.

E. Easy to Use, Yet Secure

Despite the multiple layers of protection, the system remained smooth and easy to use. It didn't slow down normal USB operations, making it practical for everyday use while still offering serious protection behind the scenes.

DISCUSSION

The results confirm that spyUSB provides a practical and effective defense against covert USB-based data exfiltration. Unlike many existing solutions that focus on malware detection or backup in isolation, spyUSB delivers a unified approach. The integration of machine learning, real-time monitoring, and layered data protection significantly reduces the risk of sensitive data exposure.

Future enhancements could include real-time threat visualization dashboards, adaptive learning from new malware patterns, and extending support to additional operating systems.

IV. OUTCOME

Fig 1: Home Page.

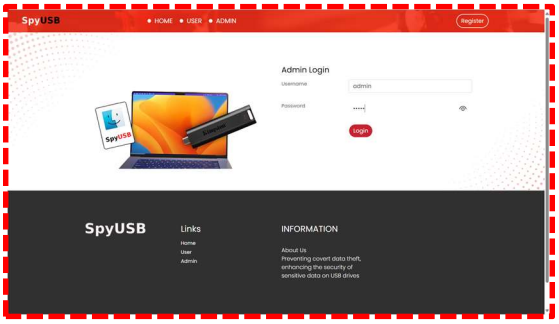

Fig 2: Admin Login Page.

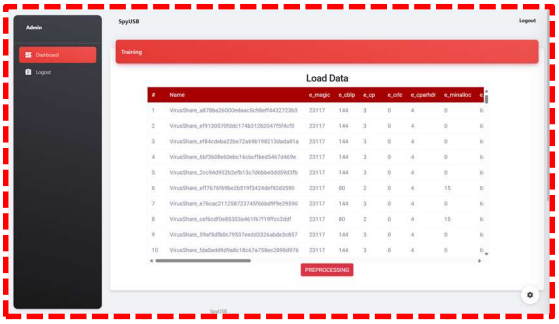

Fig 3: Dashboard Page.

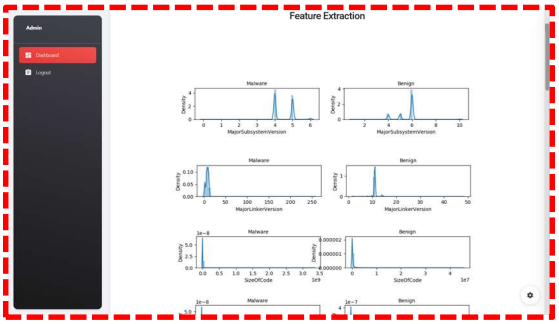

Fig 4: Teature Extraction.

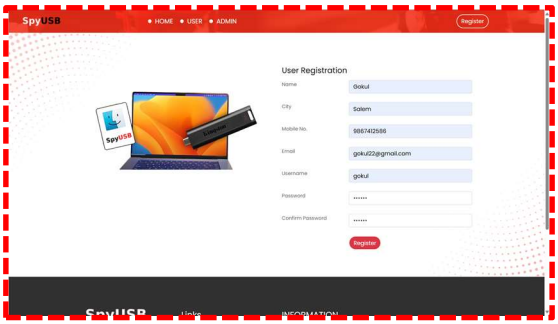

Fig 5: User Registeation Page.

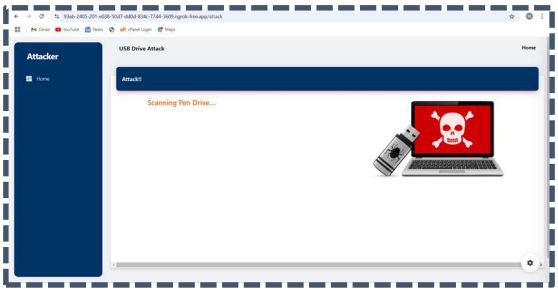

Fig 6: Scanning Pen Drive.

Fig 7: Cloud Conceal.

VII. CONCLUSION

In this work, we presented spyUSB, an integrated security framework designed to detect and prevent covert data theft through USB devices. By combining deep learning-based malware detection, real-time USB monitoring, cloud-based encrypted backups, and tokenization-based data masking, the system offers a comprehensive defense against malware injection and data exfiltration attacks.

The experimental results demonstrated high accuracy in threat detection, reliable data backup performance, and effective masking of sensitive information. Importantly, spyUSB achieves this without disrupting normal system operations, maintaining a strong balance between usability and security.

Unlike traditional solutions that address either detection or data protection in isolation, spyUSB offers a holistic, automated response to USB-based cyber threats. The system is well-suited for both personal and enterprise environments where data confidentiality and system integrity are critical.

Future work may involve enhancing threat intelligence through continuous learning, supporting cross-platform deployments, and integrating visual monitoring tools for administrators.

REFERENCES

1. Y. Su, D. Genkin, D. Ranasinghe and Y. Yarom, "USB snooping made easy: Crosstalk leakage attacks on USB hubs", Proc. 26th USENIX Secur. Symp., pp. 1145-1161, 2017.
2. L. Letaw, J. Pletcher and K. Butler, "Host identification via USB fingerprinting", Proc. IEEE 6th Int. Workshop Systematic Approaches Digit. Forensic Eng., pp. 1-9, 2011.
3. Bates, R. Leonard, H. Pruse, D. Lowd and K. R. Butler, "Leveraging USB to establish host identity using commodity devices", Proc. Annu. Netw. Distrib. Syst. Secur. Symp., 2014.
4. P. C. Johnson, S. Bratus and S. W. Smith, "Protecting against malicious bits on the wire: Automatically generating a USB protocol parser for a production kernel", Proc. 33rd Annu. Comput. Secur. Appl. Conf., pp. 528-541, 2017.
5. D. J. Tian, N. Scaife, A. Bates, K. Butler and P. Traynor, "Making USB great again with USBFILTER", Proc. 25th USENIX Secur. Symp., pp. 415-430, 2016.
6. M. Guri, M. Monitz and Y. Elovici, "USBe: Air-gap covert-channel via electromagnetic emission from USB", Proc. 14th Annu. Conf. Privacy Secur. Trust, pp. 264-268, 2016.
7. D. Tian, A. Bates, K. R. Butler and R. Rangaswami, "ProvUSB: Block-level provenance-based data protection for USB storage devices", Proc. ACM SIGSAC Conf. Comput. Commun. Secur., pp. 242-253, 2016.
8. G. Hernandez, F. Fowze, D. Tian, T. Yavuz and K. R. Butler, "FirmUSB: Vetting USB device firmware using domain informed symbolic

execution", Proc. ACM SIGSAC Conf. Comput. Commun. Secur., pp. 2245-2262, 2017.
9. P. Cronin, X. Gao, H. Wang and C. Cotton, "Time-print: Authenticating USB flash drives with novel timing fingerprints", Proc. IEEE Symp. Secur. Privacy, pp. 1002-1017, 2022.
10. Z. Yang, Q. Huang and Q. Zhang, "NICScatter: Backscatter as a covert channel in mobile devices", Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw., pp. 356-367, 2017.