WEB INFORMATION EXTRACTION SYSTEM TO SENSE INFORMATION LEAKAGE

H.M.N.B. Herath

(128212J)

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science, Specialized in Computer & Network Security

Department of Computer Science & Engineering

University of Moratuwa
Sri Lanka

March 2017

DECLARATION

“I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

Name: H.M.N.B. Herath

The above candidate has carried out research for the Masters Dissertation under my supervision.

Signature of the Supervisor: Date:

Name of the supervisor: Dr. H.M. N. Dilum Bandara

ABSTRACT

The smell of a data breach does not necessarily come from within; often it begins with a discovery of sensitive information posted on the Internet, notably in a PasteBin site. Data leaks related to LinkedIn, Spotify, DropBox, Morgan Stanley, and Sri Lankan organizations emphasize the importance of having an early detection mechanism for data leakages. Sensitive information leaked into such websites varies from Personally Identifiable Information (PII) such as login credentials and credit card data to top-secret military data. Once the information is published online, they will be proliferated among different popular sources notably in social media. Early detection of information leakages and cyber security incidents enable immediate removal of breached content and prompt incident response. Such an early detection platform is vital for a nation, an organization, as well as for the individuals to keep track of the data breaches related to their sensitive data.

The proposed system, namely LeakHawk, is an early detection platform that monitors sensitive data leakages and evidence of hacking attacks in PasteBin sites. Proposed solution periodically pools a given list of sources such as pastebin.com for new content. LeakHawk uses both regular expression and machine-learning based text classification techniques to analyze the content to predict the sensitivity label for an identified data leak. Automated extraction of granular-level details for each incident significantly reduces the manual intervention of analyzing the content. Furthermore, early detection through automation gives more time to attend to containment procedures immediately. As a proof of concept, an instance of LeakHawk is developed which monitors pastebin.com for sensitive data. The performance evaluation showed that the solution can maximize the recall and minimize the false-alarm rate for different non-structured data feeds.

Keywords: Data leakage monitoring, PasteBin monitoring, Sensitive data leakages, Text classification

ACKNOWLEDGEMENT

I would like to dedicate my sincere gratitude to my supervisor Dr. Dilum Bandara for his dedicated support for the success of this research. His guidance and constructive comments provided me a great value, also an encouraging background to complete this dissertation.

Further, I would like to thank the TechCERT staff, especially Prof. Gihan Dias, Dr. Shantha Fernando and Mr. Dileepa Lathsara, for their continuous encouragement provided throughout the project.

I am indeed very much grateful to Dr. Suranga Senevirathna for providing me with the fundamental knowledge which helped a lot in developing this system.

I would like to thank the entire academic and the non-academic staff of the Department of Computer Science and Engineering for their kindness extended to me in every aspect.

Last but not least, I thank my family and all my friends who supported me for the success of this piece of work.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
APT	Advanced Persistent Threat
ARFF	Attribute-Relation File Format
AUP	Acceptable Use Policy
BIN	Bank Identification Number
CC	Credit Card
CEP	Complex Event Processing
CERTs	Computer Emergency Response Teams
CID	Card Identification Number
CF	Configuration Files
CSIRTs	Computer Security Incident Response Team
CVC	Card Verification Code
CVV	Card Verification Value
DA	DNS Attack
DB	Database Dump
DLP	Data Leakage Prevention
DNS	Domain Name System
EO	E-mail Only
EC	E-mail Conversation
IRC	Internet Relay Chat
IUO	Internal Use Only
JSON	JSON JavaScript Object Notation
NCSC	National Cyber Security Centre
NER	Named Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
PAN	Primary Account Numbers
PHI	Personal Health Information
PII	Personally Identifiable Information
PIN	Personal Identification Number
PK	Private Key
POC	Proof of Concept
RSS	Rich Site Summary
SIEM	Security Information and Event Management
UC	User Credentials

1. INTRODUCTION

1.1. Background and Motivation

Digital information, owned by organizations is growing at an exponential rate. Such digital data storages may contain intriguing information of many forms such as military secrets, trade secrets, Personal Health Information (PHI), and Personally Identifiable Information (PII). Assuring the security requirements of such data, while maintaining the convenience and freedom of access, is truly a tedious task.

Data breaches have become an epidemic. In a world where digital information owned by governments is frequently targeted under the information warfare and Hacktivist movements on the rise, business entities targeted in pursuit of competitive advantages and the individuals targeted in Spear phishing attacks over personal rivalries, data breaches are becoming inevitable. While the establishment of preventive controls such as Data Leakage Prevention (DLP) systems along with procedural controls is the best strategy to counter such attacks, availability of detective controls is also vital in defense in depth.

A significant subset of the data breaches has been the incidents where the entire or a portion of the content is published back on the Internet to expose the data breach. The primary motive for such exposures is to damage the reputation of the data owners. These types of data breaches are mostly exposed via text sharing websites commonly known as PasteBin applications or social media sites such as Facebook, Twitter, and LinkedIn. Pastebin.com [1] (hereinafter referred to as pastebin) is the most well-known text sharing site on the Internet. Hacker communities and Hacktivists frequently misuse pastebin to publish stolen data and evidence of attacks. Usually, the targeted entities are unaware about the data breaches until the evidence is posted in pastebin. More than hundreds of data leaks and attack announcements on Sri Lankan organizations including telecommunication companies, hosting providers, and educational Institutes were exposed via pastebin during the last couple of years [2], [3] (see Appendix A).

Early detection of data leakages and evidence of attacks is a primary detective control employed in effective incident response. It is one of the major concerns for the entities that manage the Cyber Security in an organization. To address the problem of early detection of data leakages in PasteBin applications and other online channels, the need of an efficient and effective monitoring platform is stressed. While covering an ample breadth of origins of data leakage sources on the Internet, such a platform should effectively automate the security incident identification and classification of data being retrieved to reduce the manual intervention. It also should be customizable to cater the data leakage detection requirements of an individual to national-level mass data breaches.

1.1.1. Importance of Early Detection

Early discovery of information leakages and security incidents allows for immediate response and reduction of proliferation of damage. At the national scale, Computer Emergency Response Teams (CERTs), Computer Security Incident Response Team (CSIRTs), National Cyber Security Centers (NCSC), and similar entities are formed to detect, prevent, coordinate and warn the authorities regarding cyber security incidents, data breaches, etc. [4]. Smaller-scale units are established in the organizational context with the same objective, which is responsible for the protection of data possessed by the organization. Monitoring for data breaches and containment of such incidents are integral parts of their jobs. As the data leakage disclosures via PasteBin sources and other social media feeds are on the rise, diverse manual methodologies are employed by those organizations while spending the significant amount of time and effort.

Following set of scenarios explains the utmost importance of the existence of an early detection platform which significantly affects the proliferation of damage and effectiveness of incident response.

Scenario one: A chain of pastes on PasteBin on a series of attacks targeting a set of banks in Sri Lanka

The monitoring platform will immediately notify each affected bank about the security incidents to initiate incident response plan to contain the incident. Furthermore, upon the prediction of a targeted attack on the banking industry of Sri Lanka, other subscribed banks, and financial organizations are duly warned to be vigilant of hacking attempts on their external facing components managed under their authority.

Scenario two: A dump of user email and password pairs posted on pastebin

The system will identify the user base affected by the incident and immediately inform them to change their credentials and limit the proliferation of damage. As the reuse of password is too mainstream, a slight delay in responding would cause an unrecoverable loss for the individuals.

Scenario three: A credit card dump is posted in PasteBin with CVV2 and other sensitive data

The system matches the Bank Identification Number (BIN) of the credit card numbers and identifies the issuing banks of the breached accounts. The issuing bank can initiate their contingency plans and incident response plans accordingly. Furthermore, the bank can remove the content with immediate effect by reporting the incident to the website administrators.

In a similar scenario where an individual who is using *LeakHawk* for his/her personal use, can immediately get notified about the data breach as a personal notification. He/she can inform their issuing banks to prohibit the breached Credit Card from making further transactions.

1.1.2. Identification and Classification of Data Breaches

A successful monitoring platform for detecting security information leakages and hacking incidents will require a comprehensive incident verification strategy. With a volume of information being fed into such a system, it should be potent enough to identify all the relevant security events (maximum *recall*) while minimizing the number of false alarms (maximum *precision*).

Once a potential data breach or a hacking attack is identified, the system should be able to classify the incident based on the severity. It should automate the manual verification process to some extent and assist the administrators in initiating the appropriate containment procedures on a timely basis.

1.2. Problem Statement

This research attempts to address the following problem:

In the event of a data leakage, how to identify and classify/rank such incidents while maximizing recall and minimizing false positives?

In a situation where sensitive information belongs to a particular entity is leaked onto the Internet, there should be a mechanism to identify them promptly. Further, it should analyze the content and classify it based on the severity of the data leak. Such a solution should not exclude any sensitive information leakage as false negatives and minimize the number of false alarms of erroneous identifications.

In the perspective of research work, complexities in the development of a monitoring platform for the detection of sensitive information leakages and evidence of hacking attacks can be interpreted as a natural language text classification problem of non-structured and semi-structured data. Therefore, the incorporation of machine-learning techniques and rule-based methods to improve the accuracy and reduce the false alarms are greatly encouraged within the security communities.

Despite the type of origin (text sharing sites, Facebook feeds, Twitter feeds, etc.), any content is ultimately a textual input, which can be subjected to a series of text processing tasks to identify the semantics. Based on the semantics, the system should be capable enough to predict the severity of a particular data feed along with granular level results of analysis to reduce the manual intervene.

1.3. Objectives

The primary objectives of this research are as follows:

- Develop a scalable platform, which monitors various text-based Internet channels for sensitive information leakages and evidence of hacking incidents. The proposed system is built such that it monitors PasteBin applications and social media feeds for indications of data leakages and notifications of hacking attacks.
- To analyze the content of a particular data leak and predict the sensitivity of the incident. The system will incorporate pattern-based and machine-learning based methodologies to analyze the content and rank the sensitivity on the severity.
- Once a security incident is identified, the system needs to notify the data owners about the incident with the immediate effect. The proposed system will maintain a database of the data owners and inform them when a data leakage is identified.

The system will follow an extensible architecture where further online data leakage sources and system functionalities can be integrated into the system without affecting the integrity. In the scenarios where the leaked content is not available, but an evidence of a data breach is available, the system should able to identify them as well. Ideally, the solution should not miss out any sensitive data leakage incident as false negatives and minimize the false positive rate to reduce the management overhead.

1.4. Research Contributions

Through the development of *LeakHawk* we make the following research contributions:

- Design for an automated and scalable framework for the early detection of sensitive information leakages and evidence of attacks about a defined domain (e.g., a country, organization or an individual).
- A text corpus for the use of research activities related to security information leakage.
- A methodology to define an information model about a particular entity, which contains the unique attributes to verify whether that entity is involved in a data leakage incident.
- A proof of concept solution is implemented targeting pasetbin.com for sensitive information leakages and evidence of hacking attacks related to Sri Lanka.

1.5. Outline

Chapter 2 presents the related work. It discusses the existing systems that monitor different online sources for information leakages. Furthermore, it provides a summary of text classification methodologies adapted by various applications and how they can be integrated to the development of *LeakHawk*. Chapter 3 presents the research methodology of the proposed early detection platform, *LeakHawk*. It describes how the each module is designed, proposed architecture and data flows. Furthermore, the chapter presents the evaluation criterion that is used to verify the achievement of functional and non-functional requirements. Chapter 4 describes the proof-of-concept implementation of the proposed architecture. The development process of the extensible early detection platform for detecting evidence of attacks and sensitive information leakages is also presented. Functional and performance analysis is presented in Chapter 5 together with a comparison of *LeakHawk* with the existing systems. Chapter 6 concludes the thesis by summarizing the results and introducing future work.

2. LITERATURE REVIEW

This chapter formulates the background information and existing literature related to the research problem. Section 2.1 introduces the data breaches and evidence of attacks. Furthermore, it discusses the criticality of exposing the hacking attacks and sensitive data leakages onto the Internet. Section 2.2 presents a brief history of security incident exposures related to Sri Lanka. Section 2.3 provides an analysis of the PasteBin applications in terms of architecture, features and limitations with respect to security incident monitoring. A discussion on the existing PasteBin surveillance systems and their capabilities are presented in Section 2.4. Section 2.5 analyzes the current text classification methodologies and how they can be incorporated in to the design of *LeakHawk*. Section 2.6 discusses the existing Data Loss Prevention (DLP) systems and how that knowledge can be incorporated to fine-tune the proposed data leakage monitoring platform.

2.1. Data Breaches and Evidence of Attacks

2.1.1. What is a data breach?

The most valuable asset owned by an organization is its own data. From military secrets to customer information, data owners put multi-level security controls to protect the data, to assure the confidentiality, integrity and availability aspects of it.

A *data breach* is defined as an incident where sensitive, protected or confidential data has potentially been exposed, stolen or used to/by an unauthorized individual [5]. Such data breaches could range from viewing a credit card number by a fellow employee via shoulder surfing to sophisticated database dump containing thousands of confidential customer records resulted in an Advanced Persistent Threat (APT).

A data breach can be harmful in many ways. Once the security controls are penetrated, and the valued information is put into a wrong hand, the consequences are unpredictable. For example, it could put an entire nation at risk of a terrorist attack or an organization may have to pay a huge penalty or lose its reputation, damaging its competitive advantages. An individual who is subjected to a credit card breach may lose a significant amount of money via unauthorized transactions [6].

Today, hackers tend to publish everything they find on the Internet. Followed by a data breach, leaked information is made available in various web-based channels such as PasteBin applications. As per our study, such data dumps may contain interesting information such as:

- Cardholder Information (e.g., Credit card numbers, track data, etc.)
- Login Credentials
- Database Dumps
- Configuration Files
- Personally Identifiable Information
- Confidential financial documents including corporate e-mail conversations

2.1.2. Evidence of attacks

Apart from sensitive information, hackers publicize evidence of attacks via social media feeds and text sharing sites. In most cases, results of politically motivated attacks and Hacktivist movements are posted online to embarrass the targeted entities. Notably, following types of attacks are exposed via online means:

- Web site defacements
- DDoS attacks
- SQL Injection attacks
- DNS related attacks (zone transfers and cache poisoning)

2.1.3. Making Data Breaches and Hacking Incidents Public

For organizations that own critical information assets such as customer data, intellectual property and proprietary corporate data, the risk of a data breach is now higher than ever before. When it comes to government and military-related entities, it becomes more and more critical. Even in the cases where certain organizations that do not have very sensitive information under their repositories, but maintains a good online presence, will be under external threats with respect to their reputation. For example, the primary website of a renounced non-profit organization can be defaced by a Hacktivist group which eventually poses a severe damage to their reputation.

Data leakages can be defined as a result of a combination of opportunity, motivation, and rationalization (fraud triangle). Out of these three, motivation driven attacks could be either target the data possessed by the target organization or simply the reputation of the target organization.

Hackers reveal the stolen content for various motives. Sometimes cyber criminals are paid by rivals to target the infrastructure and data owned by their opponents. Even though the value of the stolen data is expired, hackers tend to publish the data dumps online, just to harm the reputation of the data owners. As evident in the recent data breach of one of the major private banks in Sri Lanka, the published content did not affect a direct financial loss, but greatly impaired the reputation of the bank [7]. Alternatively, a successful penetration of security parameters of a renowned organization could significantly improve the status of a hacker who conducted the attack. Revealing the stolen content will attest the misconduct and the attackers are endorsed among the hacking communities.

The primary motivations for exposing data breaches and evidence of attacks via various online channels such as social media and text sharing sites can be summarized as follows:

- Damage the reputation of the target (once published, the damage will be proliferated as the other interested parties will utilize the exposed data for making further attacks)
- Cyber criminals are financially motivated to harm the reputation of the competitive organizations.
- Expose the concerns of external security postures (some attackers target the vulnerabilities of popular websites to let them know about the lack of controls of their external security posture. The motive behind such attacks are not malicious, but exposing the vulnerabilities into public channels will violate the white-hat security principles)

- To improve the reputation among hacking communities (hackers publish successful attacks conducted by them to get recognition)
- For the own pleasure of the attackers (some hacking incidents are just made public by the attackers just for their pleasure)

2.2. Data Leaks Related to Sri Lanka

For years, cyber warfare has been used to conduct sabotage and espionage against governments, officials, and public and private corporations. Cyber warfare has targeted missile guidance systems, power grids, nuclear reactors and more [8].

Although not being an iconic character in the cyber warfare, Sri Lanka has suffered numerous hacking incidents that have been exposed via online channels. In 2011, a series of attacks were carried out by a hacker group called AnonymousSriLanka targeting a set of government institutes, educational institutes, Internet Service Providers, etc. [2]. These attacks were politically motivated and identified as an outcome of anger towards Sri Lanka after the eradication of LTTE terrorists. In 2013, another set of attacks were conducted against a set of online targets belong to Sri Lankan organizations [3]. Apart from these major incidents, some ad-hoc sensitive information dumps and evidence of hacking incidents have been posted in online channels time to time. Most out of the remaining incidents are exposed via social media feeds. Refer APPENDIX A for a list of data breaches and hacking incidents exposed via PasteBin sites related to Sri Lanka. This list contains only the publicly disclosed breaches. Recent incident targeting one of the major commercial banks in Sri Lanka was exposed via a Twitter Feed [7].

2.3. Pastes and PasteBins

A *paste* is defined as a textual content posted onto a website where it receives its unique URL so that it can then be shared to access the paste. The contents of a paste could be anything – a programming code chunk, configuration file, a recipe, an algorithm or of particular interest here, a dump of leaked information.

Text sharing applications (also known as PasteBins, paste sites, and code sharing sites) available on the web, allows users to post snippets of text, usually source code or log files, for public viewing. These simple websites provide the users an easy interface for creating, managing and sharing textual content via multiple channels. These web applications were originated to assist Internet Relay Chat (IRC) to share a large amount of texts between users using the unique URL provided by the website. Usually, pasting of large quantities of text is considered bad etiquette in IRC channels.

Following are the most used PasteBin applications [9]:

- pastebin.com
- www.pastebay.net
- pastebin.mozilla.org
- anonymous.piratenpad.de
- piratepad.net
- codepad.org
- shorttext.com
- hastebin.com
- pastie.org
- codeupload.com
- stickypaste.com
- cryptobin.org
- privatepaste.com
- securepastebin.com
- pastebay.com
- pastebin.ca
- www.anonpaste.me
- slexy.org
- gist.github.com
- paste.ubuntu.com

Almost all the above websites provide more or less the same functionalities for the users with a similar structure. As a whole, every site considers a paste as a textual content with multiple properties such as author, subject, expiry period, and size.

www.pastebin.com was the first PasteBin application which was developed in 2002 [1]. It is the most popular PasteBin among the programmers as well as hacking communities. By 2015, a total number of active pastes was more than 65 million [10]. First security information breach on pastebin was reported in 2009 when roughly 20,000 compromised Hotmail accounts were disclosed in a public post [11]. Initiated from that, pastebin has become one of the major playgrounds of the renowned Hacktivist groups like Anonymous and Lulz.

2.3.1. PasteBin Applications and Security Incidents

Being simple, reliable and easy-to-use, text sharing websites such as pastebin allows their users to even anonymously publish documents online and keep them valid for a longer time span. Most of the pastes are not proactively moderated by the website administrators. These are the exact features required by hacker groups or whistleblowers to publish sensitive content on the Internet. As a result, while being used by the programmers to store and share pieces of source codes or configuration information, PasteBin applications are frequently and inevitably abused by the hacker communities for illegal activities such as leaking compromised content to the public, boast about targeted attacks carried out by hacking groups, etc.

The following are some of the key reasons that drive the popularity of PasteBin sites among hacking communities for sharing information and leak-out data breaches [12], [13]:

1. Ease of use – PasteBins consist of simple interfaces that allow users to post without much effort and create a unique URL for each paste
2. PasteBin sites allow anybody to use their services without any authentication, which preserves the anonymity of the users (if the users submit data via a proxy chain, Tor or any other tool which takes care of the privacy, complete anonymity will be preserved)
3. Allows users to share long text messages without violating the AUPs of third-party websites like Twitter and IRC chat servers

Entities who are concerned about their data security and security researchers in general monitor PasteBin applications for sensitive content and evidence of hacking incidents. Those websites have become a primary origin of uncovered data breaches.

2.3.2. Site Structure – Pastebin

Most of the well-known PasteBin sites follow a similar architecture. This section of the literature survey describes the attributes and functionalities of the www.pastebin.com regarding the importance of monitoring for sensitive information leakages and evidence of hacking attacks.

Figure 2-1 illustrated the main interface of the homepage of pastebin.

Table 2-1 describes the each attribute of the main interface in detail. Trending Pastes page allows the users to view the pastes with most hits [14]. It can be customized to display popular pastes at different times such as right now, last seven days, last 30 days; last 365 days and all time. Figure 2-2 shows the trending pastes in the last month. As seen in the figure, almost all the pastes are apparently related to a data leak or hacking incident. Public Archive or the Paste Archive page lists all the newly added pastes on a single page [15]. If anyone is interested in scrapping PasteBin for data leaks or hacking notifications, he/she will need to monitor this page. However, the application does not allow the users to make too many requests. Such IP are blacklisted for few hours. Most of the PasteBin applications follow the same behavior and that is one of the hurdles in building PasteBin monitoring tools.

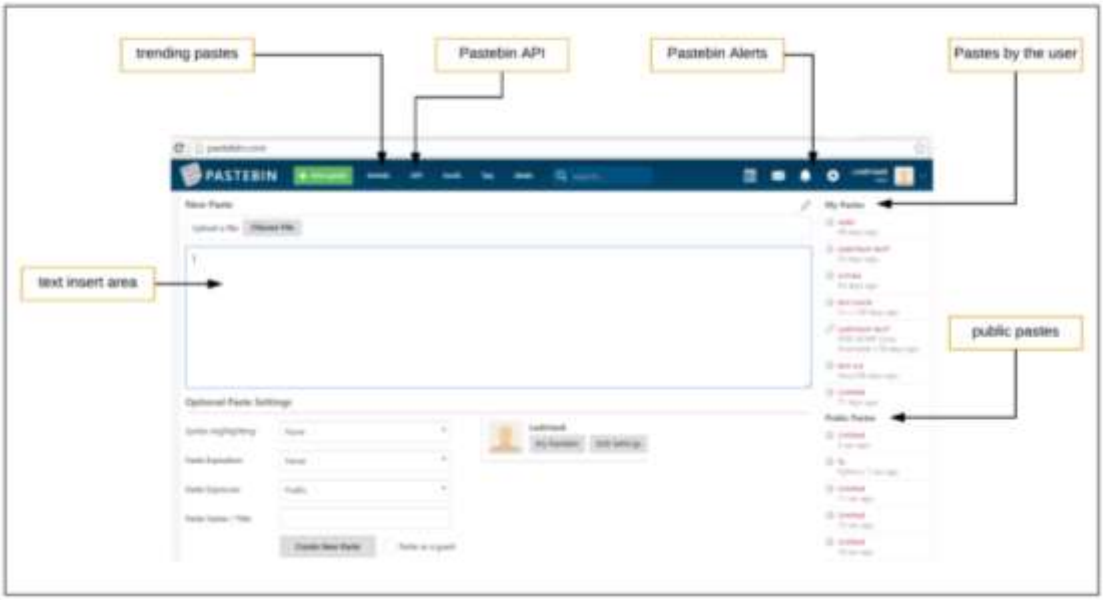
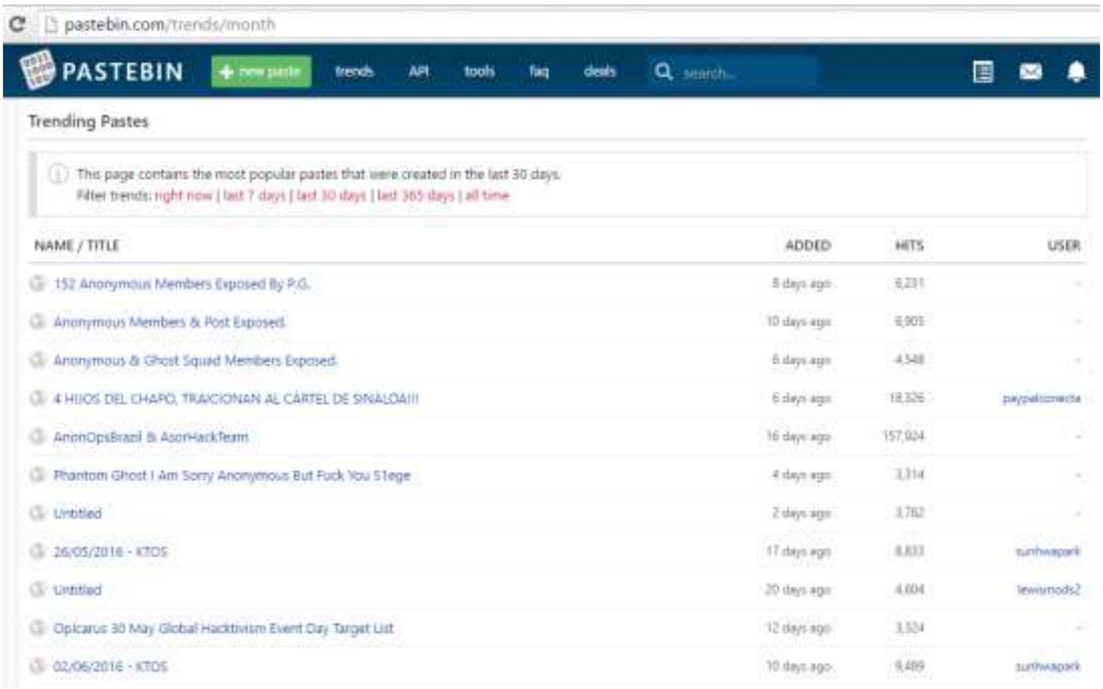


Figure 2-1: Main interface – pastebin.

Table 2-1: Attributes of the main interface – pastebin.

Section	Description / Importance
Trending Pastes	Trending pastes lists the most frequently accessed pastes by all the users. Mostly this section lists leaked data from popular targets as such data will attract a lot of attention. (see Figure 2-2)
PasteBin API	PasteBin provides an API for the users to publish their posts conveniently. It also provides a scrapping API (paid service) for searching and downloading pastes.
PasteBin Alerts	PasteBin allows the users to provide a set of keywords and be notified via e-mail when a post is made containing any of those keywords.
Text Insert Area	This area will contain the text dumps. Normal users can post data up to a maximum size of 512 kilobytes; PRO users can paste up to 10MB. A single paste can accommodate considerably a larger text dump which is one of the reasons paste sites are used by hacking communities to dump their data.
Pastes by the user	This section lists the pastes made by the logged-in user.
Public Pastes	This section is called the PasteBin Archive. It is frequently being updated with all the public pastes made by all users. If someone is interested in monitoring the PasteBin real time for leaked data, he will be required to focus on the content of this page. (see Figure 2-3)

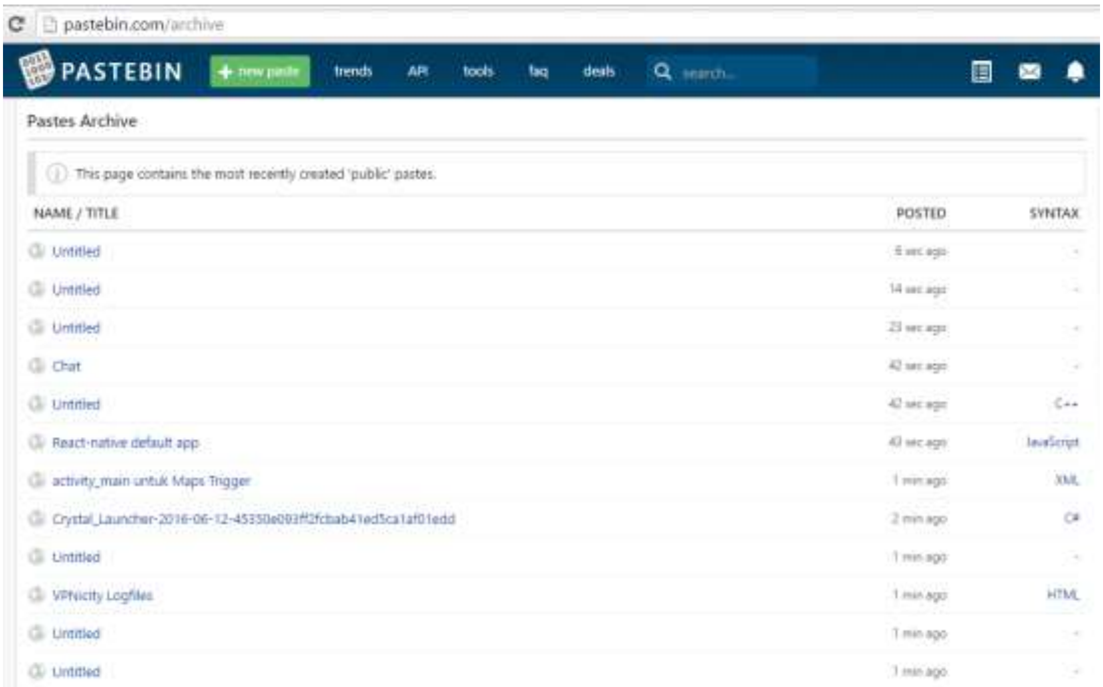


Trending Pastes

This page contains the most popular pastes that were created in the last 30 days.
Filter trends: right now | last 7 days | last 30 days | last 365 days | all time

NAME / TITLE	ADDED	HITS	USER
152 Anonymous Members Exposed By P.G.	8 days ago	6,231	-
Anonymous Members & Post Exposed.	10 days ago	6,905	-
Anonymous & Ghost Squad Members Exposed.	6 days ago	4,548	-
4 HIJOS DEL CHAPO, TRAICIONAN AL CÁRTEL DE SINALOA!!!	6 days ago	18,326	paypalconecta
AnonOpsBrazil & AsorHackTeam	16 days ago	157,924	-
Phantom Ghost I Am Sorry Anonymous But Fuck You S1ege	4 days ago	3,314	-
Untitled	2 days ago	3,782	-
26/05/2016 - KTOS	17 days ago	8,833	sunhwapark
Untitled	20 days ago	4,604	lewismods2
Opicarus 30 May Global Hacktivism Event Day Target List	12 days ago	3,324	-
02/06/2016 - KTOS	10 days ago	9,489	sunhwapark

Figure 2-2: Trending posts page – pastebin (snapshot was taken on June 12, 2016).



Pastes Archive

This page contains the most recently created 'public' pastes.

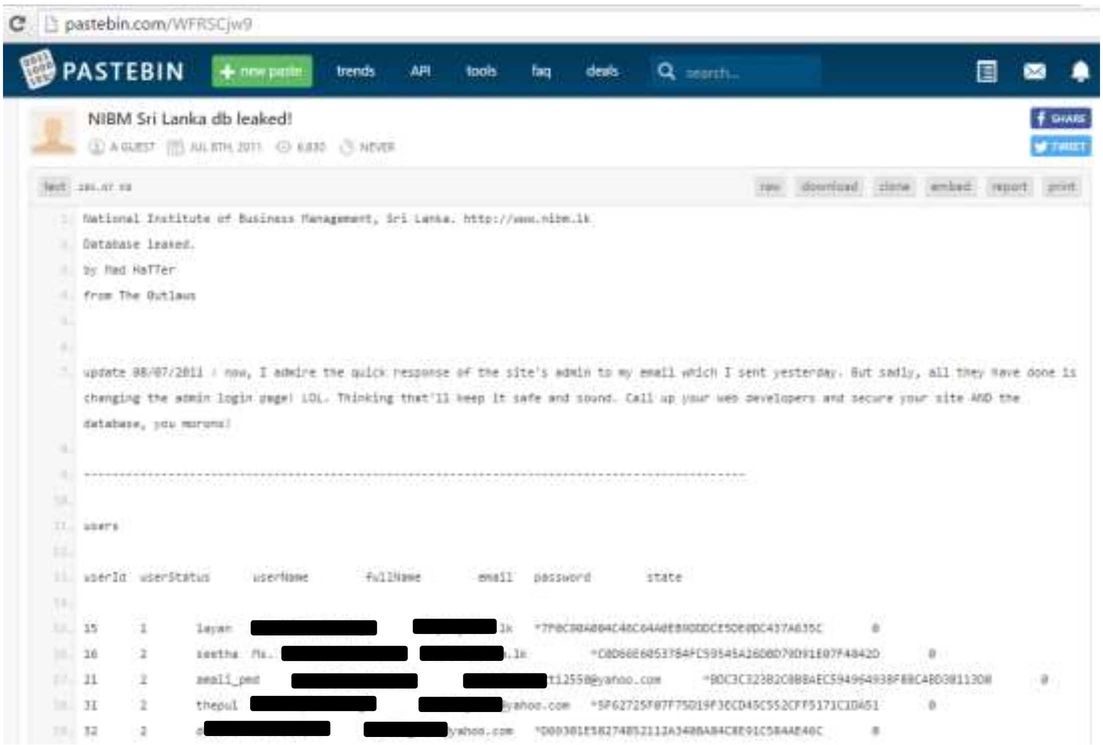
NAME / TITLE	POSTED	SYNTAX
Untitled	6 sec ago	-
Untitled	14 sec ago	-
Untitled	23 sec ago	-
Chat	42 sec ago	-
Untitled	42 sec ago	C++
React-native default app	43 sec ago	JavaScript
activity_main untuk Maps Trigger	1 min ago	XML
Crystal_Launcher-2016-06-12-45350e093ff2fcbab41ed5ca1af01edd	2 min ago	C#
Untitled	1 min ago	-
VPNicity Logfiles	1 min ago	HTML
Untitled	1 min ago	-
Untitled	1 min ago	-

Figure 2-3: Site Structure – Public Archive Page.

Figure 2-4 is an example of a paste/post that can be found in PasteBin. To illustrate the attributes, the author used a sample paste containing a critical database dump of a renowned educational institute in Sri Lanka. Each such paste has the following properties:

1. Unique URL: e.g. pastebin.com/WFRSCjw9
2. Subject: e.g. NIBM Sri Lanka db leaked!
3. User: e.g. GUEST (not authenticated)
4. Published Date: e.g. JUL 8TH, 2011
5. Unique Visits till the date: e.g. 6830
6. Expiry Date: e.g. NEVER
7. Raw URL: e.g., http://pastebin.com/raw/WFRSCjw9
8. Size: e.g. 285.47 KB
9. Syntax: e.g. TEXT
10. Key: e.g. WFRSCjw9

Although the PasteBin is frequently being misused for posting breached data, hacking notifications, login credentials, pornographic content, website does maintain an Acceptable Use Policy as seen in Figure 2-5. Pastebin makes it clear that posting personal data, email lists, login credentials are against the AUP and will result in its removal. However, with the amount of posts being made per day, the site administrators depend on the abuse reports submitted by the users for content removal, rather evaluating each paste. However, the other PasteBin applications may be less accommodating, which require commercial or legal motivation for content removal and to retrieve origin information to support forensic investigations.



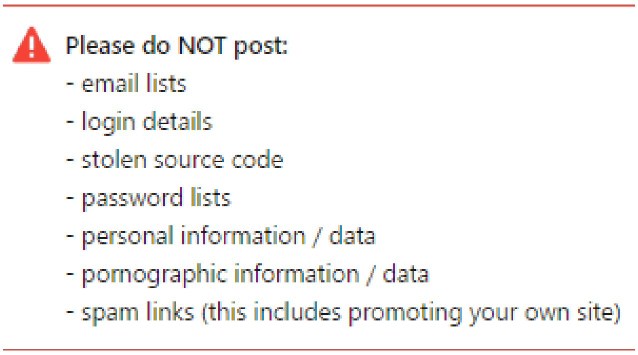
pastebin.com/WFRSCjw9

NIBM Sri Lanka db leaked!

A GUEST JUL 8TH, 2011 6,830 NEVER

```
National Institute of Business Management, Sri Lanka. http://www.nibm.lk
Database leaked.
by Mad HaTTer
from The Outlaws


update 08/07/2011 : now, I admire the quick response of the site's admin to my email which I sent yesterday. But sadly, all they have done is changing the admin login page! LOL. Thinking that'll keep it safe and sound. Call up your web developers and secure your site AND the database, you morons!

------------------------------------------------------------------------------------------

users

userId  userStatus     userName       fullName       email   password       state

15      1       layan [redacted]     [redacted]lk  *7F0C98A084C48C64A0EB9DDDCE5DE0DC437A635C       0
16      2       seetha  Ms. [redacted]  [redacted].lk        *C0D66E6053784FC59545A260BD79D91E07F4842D       0
21      2       amali_pmd   [redacted]     [redacted]t1255@yahoo.com      *BDC3C323B2C0BBAEC594964938F88C4BD381130B       0
31      2       thepul [redacted]     [redacted]yahoo.com    *5F62725F07F75D19F3ECD45C552CFF5171C1D451       0
32      2       d[redacted]     [redacted]yahoo.com   *D00301E5827485211 2A3480AB4C8E91C5B44E40C       0
```

Figure 2-4: Sample DB dump posted on PasteBin.



Please do NOT post:
- email lists
- login details
- stolen source code
- password lists
- personal information / data
- pornographic information / data
- spam links (this includes promoting your own site)

Figure 2-5: www.pastebin.com AUP [16].

Although the AUP warns the users not to post any sensitive data, site owners do not enforce any control to prevent the users from doing so. That is one of the primary reasons for the popularity of PasteBin among hacking communities.

2.4. Existing PasteBin Monitoring Systems

Early discovery of leaked data allows for immediate removal and damage limitation. With that intention, numerous systems are developed to monitor these paste sites in different scales. Most of these solutions follow a semi-automated approach and try to identify the sensitive information based on static keyword lists. One of the crucial parts of these platforms is the human intervened manual validation of the identified data leaks to conclude whether they are false positives or not. However, with the vast amount of potential data leaks detected, this approach will produce a much higher percentage of false-positives, which makes it harder for the administrators to act upon. PasteBin applications are frequently being monitored by intelligence agencies, social media giants and other organizations and individuals who take data leakages seriously. Here we discuss some of the existing mechanisms that are developed to address these requirements in detail. For each existing solution, the author presents a comparison and discusses the importance of a fully automated platform as proposed in this thesis.

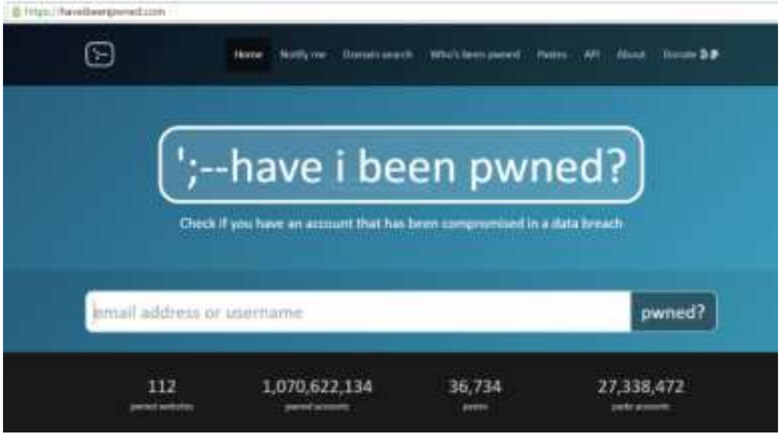
2.4.1. Facebook Monitors Pastebin for Leaked Credentials

Facebook has started to monitor PasteBin and related text sharing sites from 2014 with the purpose of identifying potential credential leaks of users [17]. This system does not target only the Facebook login credential leaks. As the users just reuse the passwords across multiple websites, they focus on all the credentials leaked online. Upon finding a collection of email addresses and passwords, Facebook utilizes an automated process to check them against the user database of the social network. This process was initiated by the company followed by the incident where 700,000 DropBox Credentials were leaked on PasteBin with email and password pairs [18]. Facebook focuses only on the pastes containing email addresses and respective credentials, and the users cannot customize it to generalize for other sensitive content. The underlying architecture of this mechanism is not available to analyze. In comparison, the proposed platform covers a broader scope of sensitive information types apart from the user credentials and analyzes the depth of each incident in detail.

2.4.2. Haveibeenpwned.com [HIBP]

haveibeenpwned.com is a popular security monitoring service that allows Internet users to check if their personal data has been compromised by data breaches [19]. The website maintains an extensive database of all the dumps from data breach incidents occur on the Internet. It just allows users to search for their information by entering their username or email address (see Figure 2-6). Service will match the entered data against its database and verify whether they are connected with any previous data breach. It also allows users to sign up to be notified if their email address appears in future dumps. In 2014, Troy Hunt, the creator of this web service, enriched its services by utilizing some of the existing PasteBin monitoring tools to add potentially leaked data into its database automatically. Thus, the service allows users to check whether their personal information has been leaked to PasteBin sites [13]. The website later added a new functionality to allow users or institutes to provide multiple domain names managed by them, to check whether they are involved with any data breach.

Although this website provides an extensive security monitoring service, it does not facilitate the users to track any sensitive content other than credentials. For instance, private keys, Credit Card dumps, Configuration dumps, are very commonly leaked in to pastebin. haveibeenpwned.com heavily depends on the DumpMonitor (DumpMon) Twitter bot which monitors PasteBin-like sites for leaked content [20].

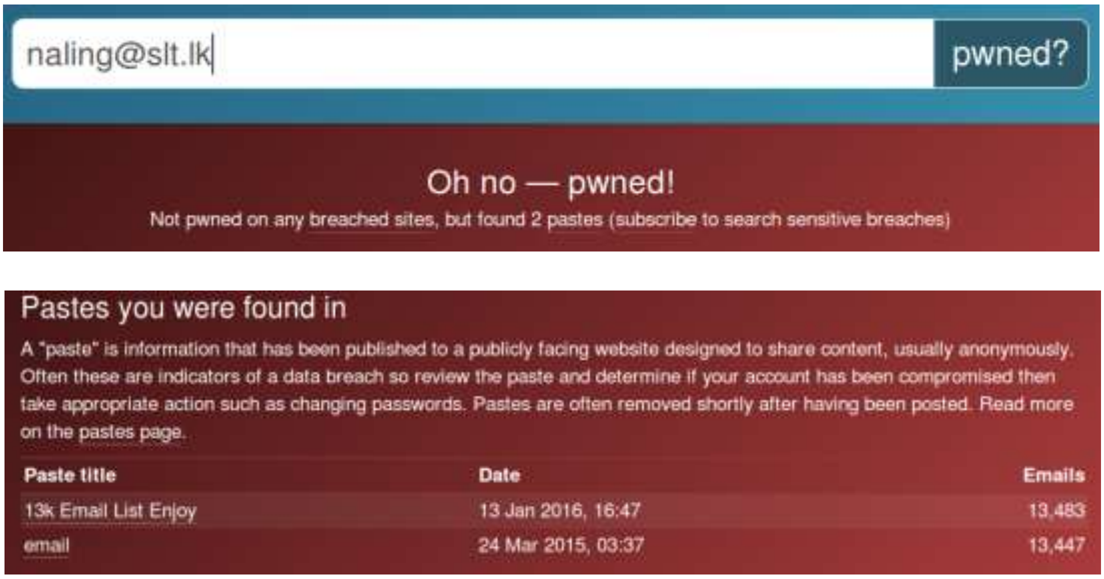


';--have i been pwned?

Check if you have an account that has been compromised in a data breach

Figure 2-6: HIBP primary interface.

During our evaluation on the accuracy of the results provided by the HIBP, it was identified that the PasteBin monitoring script used by the DumpMon twitter bot generates considerable number of false-positive results. However, the author identified some critical false-negatives regarding certain user accounts. For example, query made for “naling@slt.lk” will state that there are two different pastes in pastebin which contains this email address (see Figure 2-7 for a sample false-negative result of HIBP). However, *LeakHawk* training set generation process identified another critical PasteBin paste, which contains the evaluated email address along with a password (see Figure 2-8). Therefore, it is evident that the system could produce critical false-negatives. Table 2-2 lists the features of HIBP.



naling@slt.lk

Oh no — pwned!

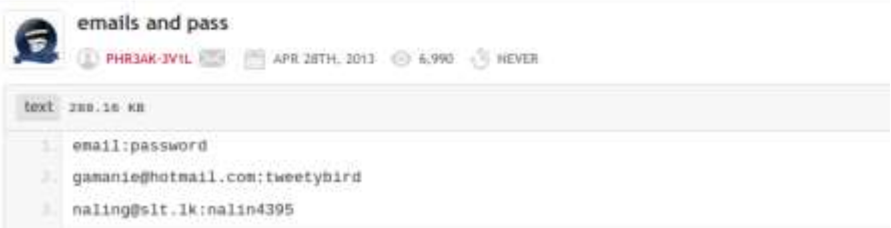
Not pwned on any breached sites, but found 2 pastes (subscribe to search sensitive breaches)

Pastes you were found in

A "paste" is information that has been published to a publicly facing website designed to share content, usually anonymously. Often these are indicators of a data breach so review the paste and determine if your account has been compromised then take appropriate action such as changing passwords. Pastes are often removed shortly after having been posted. Read more on the pastes page.

Paste title	Date	Emails
13k Email List Enjoy	13 Jan 2016, 16:47	13,483
email	24 Mar 2015, 03:37	13,447

Figure 2-7: Instance of false-negative findings of HIBP.



```
email:password
gamanie@hotmail.com:tweetybird
naling@slt.lk:nalin4395
```

Figure 2-8: Screenshot of a pastebin post containing credentials.

Table 2-2: Feature comparison of HIBP.

Feature	Description
Target online channels	PasteBins, file hosting sites
Availability of Architecture / Source code	No
The mechanisms used	Keyword based rules, manual integration of new dumps
Can be customized by the user	No
Precision / Recall	Good Recall, good precision
Available methods to define the search domain	Email addresses (one at a time), domain names (one at a time), usernames (one at a time)

In contrast, *LeakHawk* covers a broader range of attack vectors and provide the freedom to customize the platform for different uses. HIBP developers have access to even publicly unpublished information via private channels and underground forums. However, it lacks a scalable architecture that can be extended.

2.4.3. Pastefind

PasteFind [21] is a python script developed by Matt Fuller. It can be used to monitor new pastebin pastes for a provided search term. It also provides the facility to set a parameter for the time between two consecutive requests made to pastebin, as the website blocks the IP addresses that make more frequent requests. The pasteFind source code is available at [21]. Due to recent changes in the pastebin site, pastebin-find.py is not functioning as intended without some tweaks in the source code. Currently, the author is not managing the source code repository. Refer Table 2-3 for a feature list of PasteFind.

Table 2-3: Feature comparison of PasteFind.

Feature	Description
Target online channels	www.pastebin.com
Availability of Architecture / Source code	Yes
The mechanisms used	Keyword based rules
Can be customized by the user	Yes
Precision / Recall	Poor recall, poor precision
Available methods to define the search domain	Keywords, regular expressions

2.4.4. Google Alerts and Google Custom Search

Google Alerts can be used to monitor PasteBin sites [22]. But the process is not efficient as it depends on the indexing delay of Google search engine. Apart from that, the Google custom search can also be utilized to monitor PasteBin sites. For example, the following search pattern can be used to search several PasteBin sites for the availability of both the terms “Sri Lanka” “hack”.

Query: *site:pastebin.com OR site:paste2.org +"leak" +"Sri Lanka"*

2.4.5. PasteMon

PasteMon was initially developed by Xavier Garcia at shellguardians.com as a Python script (pastebin.py) to monitor pastebin using regular expressions. Later Xavier Mertens at rootshell.be rewrote the script in Perl and enhanced the functionalities of the script [23]. Pastemon.pl runs in the background as a daemon and monitors pastebin for interesting content (based on regular expressions). Detected instances are sent to Syslog so that even a Security Incident and Event Management (SIEM) solution can be configured to monitor the logs. PasteMon utilizes keyword-based rules and regular expressions to identify the sensitive content posted on PasteBin applications with decent Recall. However, it is evident that it has introduced many false-positives to the output which makes the system is very unusable without proper filters in place. Refer Table 2-4 for the feature list of PasteMon.

Table 2-4: Feature comparison of PasteMon.

Feature	Description
Target online channels	Multiple PasteBin sites including www.pastebin.com
Availability of Architecture / Source code	Yes
Mechanisms used	Keyword based rules, regular expressions
Can be customized by the user	Yes
Precision / Recall	Good recall, poor precision
Available methods to define the search domain	Keywords, regular expressions

Compared to PasteMon, *LeakHawk* incorporates machine learning-based text classification methodologies, which provide a better accuracy, precision and recall.

2.4.6. LeakedIn

LeakedIn is developed with the objective of making visitors aware of the risks of losing data [24]. It monitors the PasteBin applications based on PasteMon script. LeakedIn provides an RSS feed for all the identified sensitive content in the target sites. Table 2-5 lists the features of LeakedIn.

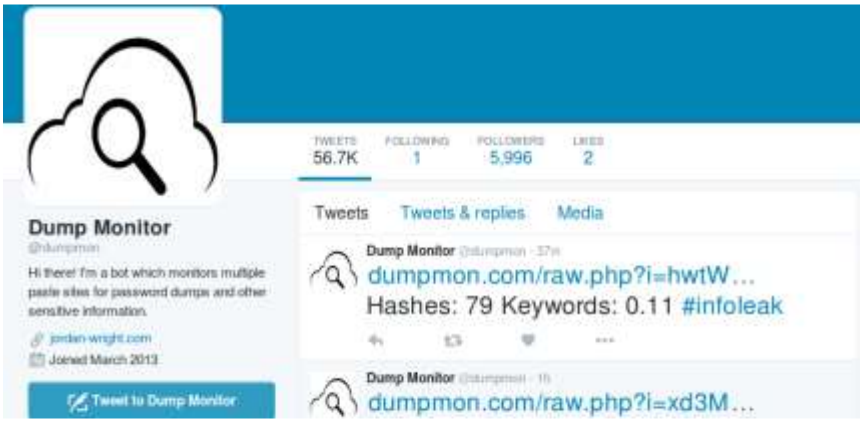
Compared to the initial implementation of *LeakHawk*, LeakedIn covers a better breadth of attack vectors and leakage types. However, the underlying mechanisms based only on regular expressions provide a considerable amount of false-positives.

Table 2-5: Feature comparison of LeakedIn.

Feature	**Description**
Target online channels	Multiple PasteBin sites including www.pastebin.com
Availability of Architecture / Source code	Yes (only the rules are visible)
Mechanisms used	regular expressions
Can be customized by the user	No
Recall / Reliability	Good recall, poor precision
Available methods to define the search domain	Only a feed is available. Cannot customize

2.4.7. DumpMon

DumpMon is a Twitter bot that tracks and reports password dumps and other sensitive information shared on paste sites such as Pastebin [20]. Troy Hunt [19] uses DumpMon as the core for monitoring PasteBin sites under his project haveibeenpwned.com. See Figure 2-9 for the Twitter feed of DumpMon.



Dump Monitor
@dumpmon

Hi there! I'm a bot which monitors multiple paste sites for password dumps and other sensitive information.

jordan-wright.com
Joined March 2013

Dump Monitor @dumpmon · 37m
dumpmon.com/raw.php?i=hwtW...
Hashes: 79 Keywords: 0.11 #infoleak

Dump Monitor @dumpmon · 1h
dumpmon.com/raw.php?i=xd3M...

Figure 2-9: Twitter feed of DumpMon.

DumpMon utilizes regular expressions to detect sensitive information leakages on PasteBin applications. Although Machine Learning (ML) techniques are not incorporated, DumpMon can be employed to identify the following data types at an acceptable accuracy:

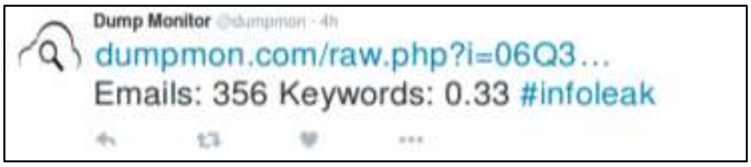
- Account/Database dumps
- Google API Keys

- SSH private keys
- Cisco Configuration Files
- Honeypot Log Dumps

DumpMon covers a more breath of PasteBins such as Pastie.org, Pastebin.com, Slexy.org. A separate thread is running for each target PasteBin site, which monitors for new pastes, downloads each one and matches it against a series of regular expressions. If a possible match is found, DumpMon will post a tweet via its Twitter page. Figure 2-10 shows a sample tweet of an identified sensitive data dump. Figure 2-11 illustrated the architecture of the DumpMon. A multi-threaded core of the platform monitors each PasteBin site. It incorporates the built-in settings and regular expression library to identify sensitive information of the retrieved feed and publish via a tweet [25].

DumpMon produces a comparatively large number of false-alarms. It significantly increases the management overhead in responding to a large set of false detections. It is important to have a multi-layered architecture for filtering the unnecessary false-detections and improve the precision, accuracy and recall. Refer

Table 2-6 for the feature list of DumpMon.



Dump Monitor @dumpmon · 4h
dumpmon.com/raw.php?i=06Q3...
Emails: 356 Keywords: 0.33 #infoleak

Figure 2-10: Sample DumpMon tweet.

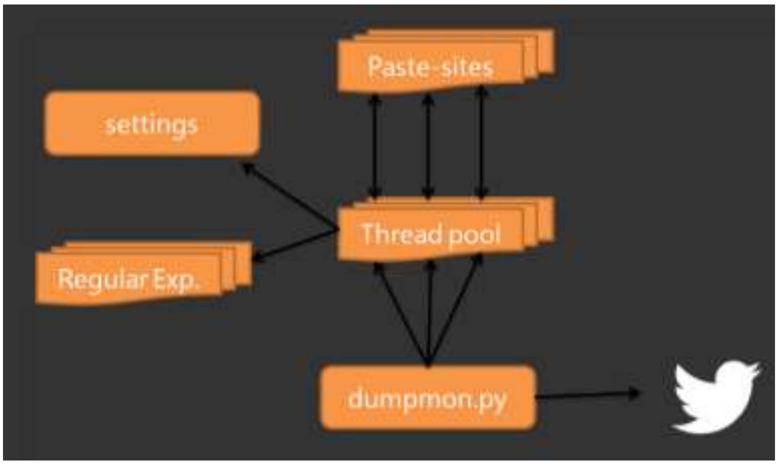


Figure 2-11: DumpMon Architecture [25].

Table 2-6: Feature comparison of DumpMon.

Feature	Description
Target online channels	Pastie.org, Pastebin.com, Slexy.org
Availability of Architecture / Source code	Architecture is provided, but source codes or regular expressions are not exposed
Mechanisms used	Regular expressions
Can be customized by the user	No
Recall / Reliability	Average recall, poor precision
Available methods to define the search domain	Only a feed is available. Cannot customize

2.5. Text Analytics for Sensitive Document Classification

The core functionality of the *LeakHawk* is its text classification process, which analyzes the textual data based on keyword-based rules and machine learning techniques. These techniques are frequently being used in the discipline of text analytics.

Text analytics is the process of analyzing unstructured text, extracting relevant information, and transforming it into useful business intelligence [26]. It can be performed by manual means, which result in high precision and recall but at the cost of a large amount of time and effort. Today, the analysis and extraction process takes advantage of techniques that originate in computational Linguistics/Natural Language Processing (NLP), statistics, and Machine Learning (ML). Under the umbrella of text analytic techniques, text mining logic of the proposed system is more concentrated on the problem of text classification of unstructured data.

2.5.1. Structured and Unstructured Data

Data is classified as structured, semi-structured or unstructured. Usually, the structured data resides in fixed fields of spreadsheets, databases, etc., while unstructured data refers to free-form texts as in text documents. In between that, certain types of data are classified as semi-structured, in which the data neither reside in a relational database nor just plain textual content, but with some process the data can be transformed and stored in a structured manner [27] (see Figure 2-2 for an illustration of documents in terms of structurization).

All the data origins that *LeakHawk* targets such as PasteBin applications and social media feeds can be viewed as mostly unstructured, yet some feeds are semi-unstructured in nature (for example, the JSON feed of new pastes in pastebin). Therefore, the problem of sensitive data identification and classification can be represented as a text classification problem of unstructured data.

2.5.2. Document Classification

Document classification is a subset of ML tasks in the form of NLP. The primary objective of document classification is to assign a document to one or more classes or categories. Document classification is known under a number of synonyms such as text classification, document/text categorization or routing and topic identification [28].

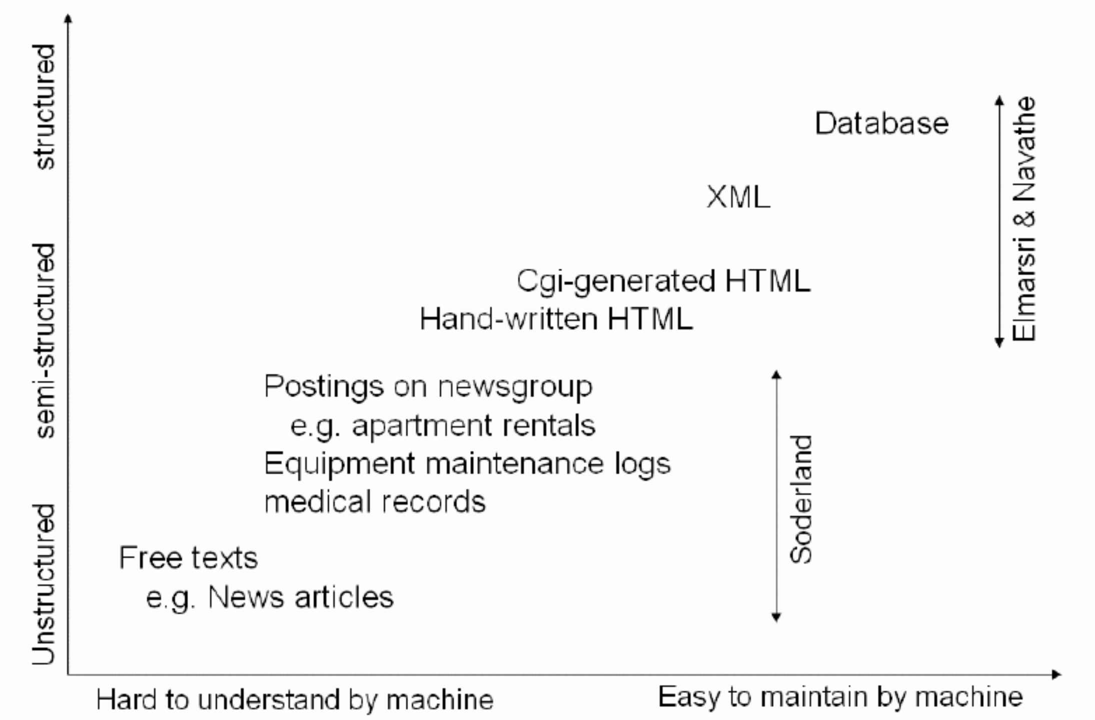


Figure 2-12: Structurization of various documents [29].

Figure 2-13 illustrates a classification of records based on the sensitivity of the content. Classification scheme followed by a particular organization may vary depend on the types and associated risks of the data they possess. As per classification methodology developed by Massachusetts Institute of Technology (MIT), the documents are classified as seen in Figure 2-13 [30].

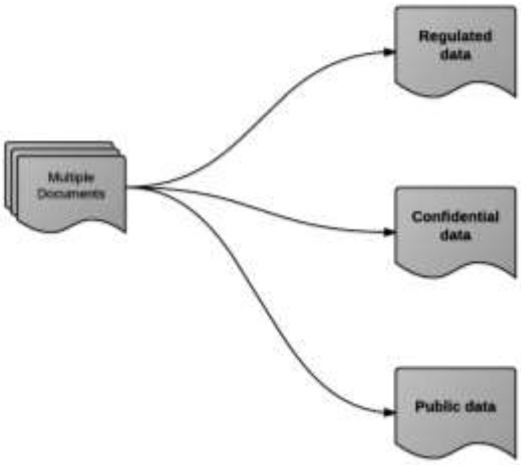


Figure 2-13: Document Classification.

Content Classifier module (refer Section 3.2.6) of the *LeakHawk* utilizes a set of multiple text classifiers for the purpose of identifying the type of data dump for further analysis. Based on various checkpoints and associated features, each simple classifier will analyze the content and classify each new paste to a pre-defined class such as Credit Card Dump, Credential Dump, DNS related exploit and so on.

Document Classification has been a standard text analytic methodology being used in popular analytical problems found in the existing literature, for example:

- Deciding whether an email is spam or not [31], [32]
- Categorization of user reviews on movies, mobiles apps [33]
- Gender Prediction [34]
- Deciding what the topic of a news article is, from a fixed list of subject areas such as sports, technology, and politics.

Each example literature mentioned above follows any of the following document classification methods.

- Multi-class classification
- Binary classification
- Multi-label classification

Binary classification is classifying the input into only two classes, such as spam or non-spam, male or female, sensitive or non-sensitive. *Multi-class* (also referred as *multinomial classification*) is the problem of classifying input instances into one of the more than two classes such as organizing the sensitive information found in

PasteBin into one of the pre-defined classes, e.g., credit card dump, user credentials, private keys, configuration files. Multiclass classification makes the assumption that each sample is assigned to one and only one label. Another variation is multi-label classification where more than one labels or classes are assigned for a particular input [35].

Proposed early detection platform will perform numerous text classification tasks in different modules. For instance, the primary objective of the *evidence classifier* is to predict whether a given input textual document contains evidence of either a hacking attack or not. Hence, it is required to classify each input into either as positive or negative. Therefore, it should follow a binary text classification methodology.

The *Content Classifier* module used in *LeakHawk* (refer section 3.2.6) should classify the input instances to one of the pre-defined classes. Therefore, the available methods would be either a complex multi-class classifier or a set of simple binary classifiers in sequence mode.

2.5.3. Text Classification Process

Supervised text classification process will follow the generic methodology seen in Figure 2-14: Supervised Text Classification Process [36]. In the training phase (a), feature extractor maps each input feed to a feature set. These feature sets, contain the parameters, which define the characteristics of each input. Pairs of feature sets and labels are fed into the ML algorithm to generate a model. In the prediction phase (b), the same feature extractor is used to convert the unclassified inputs to feature sets. These feature sets are then fed into the model, which generates predicted labels [36].

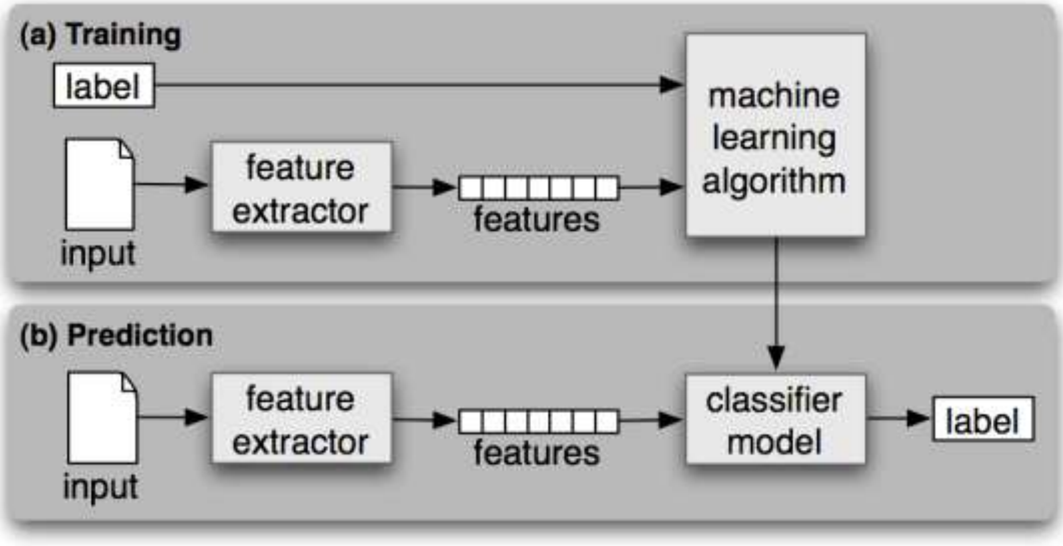


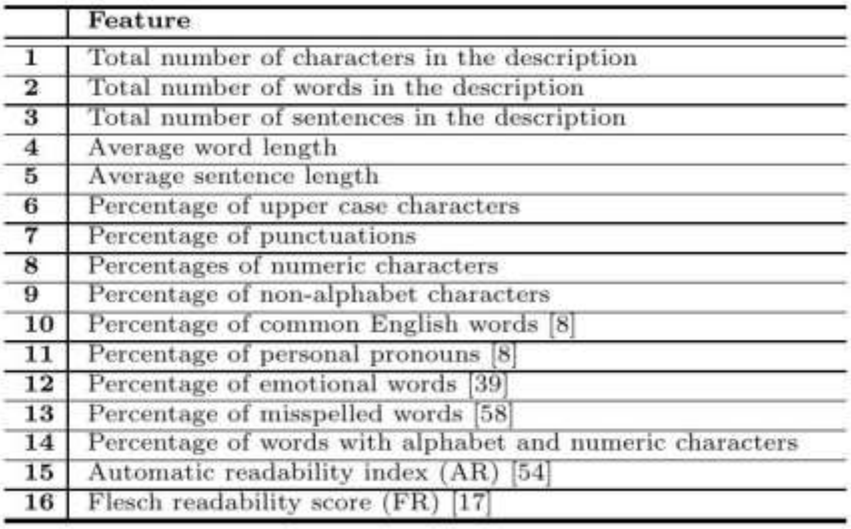
Figure 2-14: Supervised Text Classification Process [36].

2.5.3.1. Preparation of Data Collection

Senevirathne et al. [31] manually prepared the Input data collections as the training set. Most of the data sets are collected by crawling particular targets such as App Stores [31] and web crawlers [32]. In [31], apart from the content, corresponding metadata such as app name, app description, and app category are also fetched. Other than the features identified from the content analysis, such metadata about a particular input will generate some important features which could improve the accuracy of the classifier.

2.5.3.2. Generation of Feature Set

Features are created to highlight characteristics, which map the inputs to the correct class. For example, Senevirathne et al. [31] selected the feature set in Figure 2-15 to define the checkpoint “Does the app description describe the app function clearly and concisely?”



	Feature
1	Total number of characters in the description
2	Total number of words in the description
3	Total number of sentences in the description
4	Average word length
5	Average sentence length
6	Percentage of upper case characters
7	Percentage of punctuations
8	Percentages of numeric characters
9	Percentage of non-alphabet characters
10	Percentage of common English words [8]
11	Percentage of personal pronouns [8]
12	Percentage of emotional words [39]
13	Percentage of misspelled words [58]
14	Percentage of words with alphabet and numeric characters
15	Automatic readability index (AR) [54]
16	Flesch readability score (FR) [17]

Figure 2-15: Example feature-set for a given checkpoint [31].

Feature extraction should look at distinguishable characteristics of the dataset. Following list provides some of the factors that can be taken into consideration when feature set is selecting:

- Content-based features [31], [32]
  - n-gram likelihoods
  - Fraction of page drawn from globally popular words
  - TF-IDF values of each word in the dictionary
  - Number of words in the page title
  - Average length of words
- Metadata based features [32]

Some of the metadata-based features used are app category, the size of the app, develop information, etc.

2.5.3.3. Selection of the Classifier

Once the feature set is identified, feature selection and extraction methods are carried out on the extracted features to sustain the curse of dimensionality. In the practice, the *curse of dimensionality* is defined as the degradation of performance after a maximum number of features [37]. After these steps, the appropriate classifier is selected with a suitable number of features to fulfill the precision and recall requirements of the system. Once the classifier is completed, a model is developed on that classifier and new inputs are classified into the pre-defined classes based on the feature values.

In the training data set preparation phase, *LeakHawk* uses a majority of real sensitive pastes of the PasteBin sites along with a small percentage of artificially generated data to cater the dimensionality requirements. *LeakHawk* utilizes some of the aforementioned feature extraction methodologies in the development of *Evidence Classifier* and *Content Classifier*. Term Frequency – Inverse Document Frequency (TF-IDF) values are generated for most popular word unigrams, bigrams, and trigrams. TF-IDF weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus [38]. It uses some of the features produced out of the available metadata such as the syntax of the paste, availability of the subject, authors' name, and history of the author.

2.6. Classification Methodologies of DLP Solutions

Data Leakage Detection and Prevention (DLD/DLP) are well-established security mechanisms used by enterprises for the purpose of classification and monitoring of information at rest, in use and in motion. DLP solutions are optimized to locate and catalog sensitive information stored throughout the enterprise and monitor and control the movement of those identified sensitive information across the businesses and end user systems [39]. They are fundamentally different from sensitive data leakage detection mechanisms like LeakedIn, DumpMon or the proposed solution, *LeakHawk*. Still, some of the mechanisms used in text analysis in DLP solutions can be adapted to the development of the *LeakHawk*. Furthermore, both the families of solutions suffer from a common set of limitations such as analysis of graphical content and multilingual support. Next, we discuss some of the mechanisms used by existing DLP solutions, which can be utilized for the development of *LeakHawk*.

Hart, Manadhata, and Johnson [40] discussed some important aspects of text analytics used in DLP solutions:

- Sensitive data can be classified as proprietary organizational data and general data irrespective of the organization. Personally, Identifiable Information, e.g., names, credit cards, social security numbers, are confidential data regardless of the organization. When the domain is generalized to cover trade secrets type data, it will introduce more false-positives and eventually reduce the precision of the classifier.

- A DLP system faces two operational challenges: performance and accuracy, where *LeakHawk* is more focused on the accuracy aspects of it as the performance issues are not that critical compared to DLP solutions.
- Current DLP products identify confidential data in three ways: regular expressions, keywords, and hashing. First, two methods are frequently used in PasteBin monitoring applications while hashing methods are not effective since the textual data are reproduced before being published into online sources.
- A successful DLP classifier must meet two primary evaluation criteria. It must have a low false negative rate (i.e., misclassifying sensitive documents) as well as the low false positive rate for any non-sensitive document. Any system becomes unusable if it generates false alarms despite how good the system in detecting positive instances.

2.6.1. Multi-Level Analysis to Improve Precision and Recall

Multi-level filtering mechanisms are recommended to improve the identification of all the sensitive documents (maximum recall) and reduce the number of non-sensitive documents in the predicted data set of the system (maximum precision), [41].

The final decision on the sensitivity label for particular textual content is reached by a combination of results of several algorithms and methodologies. The multi-level analysis makes the classifier more effective by concentrating each level of classification or filtering for a particular data type. As shown in Figure 2-16, [41] uses multiple algorithms and rule bases for each type of data, based on the hierarchy of content classification.

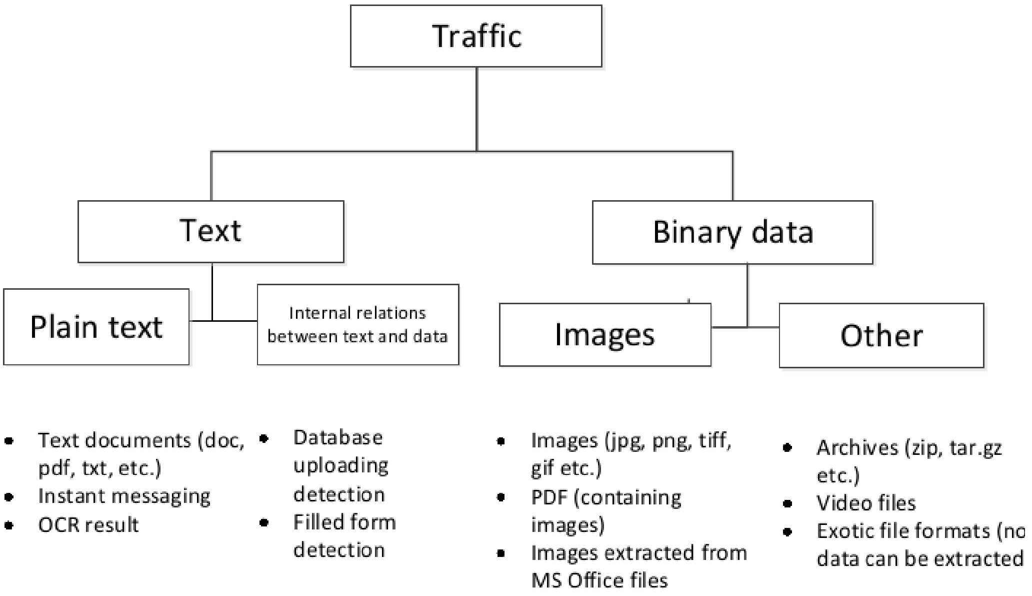

Traffic
Text
Binary data
Plain text
Internal relations between text and data
Images
Other
Text documents (doc, pdf, txt, etc.)
Instant messaging
OCR result
Database uploading detection
Filled form detection
Images (jpg, png, tiff, gif etc.)
PDF (containing images)
Images extracted from MS Office files
Archives (zip, tar.gz etc.)
Video files
Exotic file formats (no data can be extracted

Figure 2-16: Multi-level analysis for classification [41].

The proposed early detection platform does not focus on non-textual content, yet identify different types of sensitive data and evidence of hacking attacks, which require a granular level of analysis to improve the precision and recall. Successful identification of sensitive information while improving the accuracy would require multiple layers of filtering mechanisms, where each entity is specialized for a particular purpose. The same concept can be incorporated into the design of *LeakHawk*.

2.7. Classification of Textual Content Based on Sensitivity

Information classification is the classification of data based on its level of sensitivity. Clearly labeled data to inform the stakeholders to know the inherent level of sensitivity against a predefined scale is primarily designed to help ensure that the required level of safeguard is provided. Basically, the information possessed by an entity can be categorized into three types [42] as follows:

- Corporate data – Information about the organization, such as financial information, intellectual property, strategic plans, policy, practices, procedure
- Employee data – Employee data are information about individuals who work for the organization
- Customer data are information about companies or individuals who provide

revenue to the organization, such as account, transactional and contact information

The baseline for the information classification is not a standardized process, unless the data owner is not military based entity [43]. Different organizations can define restrictive information classification policies based on their data types and risk assessment results. Information that is leaked into public sources can consist of various sensitivity levels. In most cases, they contain PII such as passwords, PINs, private keys, email lists and financial data such as credit card related account data and sensitive authentication data.

For this study, the author use two information classification methodologies defined and utilized by three major educational institutes [44], [45], [46]. As per the study, primarily the sensitive information can be categorized into three major groups:

- Confidential Data
- Internal Use Only
- General Use

Confidential information is defined such that whose unauthorized disclosure, compromise or destruction would result in severe damage. PII, financial information, and health information are categorized as confidential.

Internal Use Only (IUO) is information that must be protected due to proprietary or privacy considerations. This includes employee information, unique identification numbers, etc.

Information tagged as *General Use*, is the data, which is regarded as publicly available email addresses, names, and telephone numbers.

2.8. Summary

Chapter 2 discussed the context of data breaches and highlighted the importance of an early detection platform to identify data leakages related to web-based channels such as PasteBin sites and social media sites. Furthermore, the chapter discussed the history of security incidents related to pastebin in the context of Sri Lanka. It also discussed about the existing data breach monitoring platforms and their key features and limitations. Chapter 2 also highlighted the primary attributes of pastebin and how they can be integrated into to the proposed platform in terms of data leakage identification and classification. Furthermore, the chapter analyzed the existing text classification approaches that can be incorporated into the proposed monitoring platform to improve the efficiency and effectiveness.

3. METHODOLOGY

This chapter introduces the research methodology, the design considerations used for the study, and how it has guided the data collection, analysis, and development of the proposed platform. Section 3.1 describes the design aspects of the proposed platform with respect to the extensibility and modularity. Section 3.2 describes each component of *LeakHawk* in terms of development and usage. Section 3.3 describes the data flows and activities within the proposed platform. Section 3.4 introduces the text classification process, which utilizes machine learning principles. Section 3.5 describes the sensitivity labeling process followed by an introduction to the evaluation criteria of the proposed functionalities in Section 3.6.

3.1. System Design

3.1.1. Layered Architecture

Proposed data leakage monitoring platform follows an extensible architecture, which allows future expansions and enhancements without significant changes in the implementation. *LeakHawk* is designed such that the components can be customized to cater new functional and scalability requirements. As per the component diagram (see Figure 3-1), the proposed solution follows a layered architecture. Table 3-1 describes the each layered component of *LeakHawk* in detail.

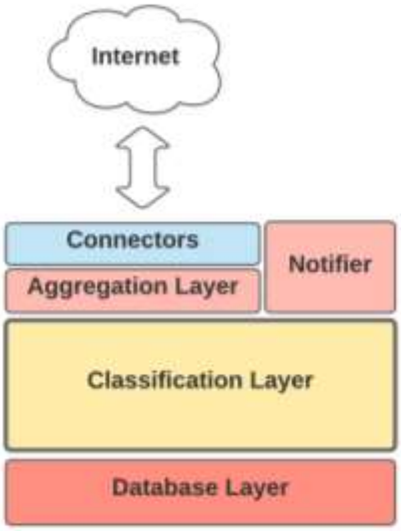


Figure 3-1: LeakHawk layered architecture.

Table 3-1: Components of LeakHawk architecture.

Component	Description
Connectors	Connectors are the entities, which keep track of new pastes, posts, and/or tweets made at the online channels that are being monitored. When a new data entry is added, the respective sensor will download it and feed into the aggregator.
Aggregators	The primary objective of the aggregators is to pre-process and align the data feed to the classifier.
Classification Layer	All the text analytic processes are done at the classification layer. It is the primary component of the LeakHawk.
Database Layer	Database layer stores the retrieved data along with the metadata, which is fed into the classification layer for processing. Furthermore, it stores the information schema (keyword list) which defines the information domain, which LeakHawk is configured to monitor. It also maintains the administrative contacts of the data owners to notify about identified data leakages and evidence of hacking attacks.
Notifier	When a security incident is predicted, Notifier will alert the respective data owners via the configured methods (e.g., email and SMS)

3.1.2. High-level Architecture

Figure 3-2 illustrates the high-level architecture design of the *LeakHawk*. It can incorporate multiple feeds from different data origins (e.g., PasteBin applications, Facebook feeds, Twitter feeds, etc.) and aggregate them into the primary classification engine, the *LeakHawk Core*. Despite the origin of text feed, ultimately the input data will be just plain text. Therefore, the classification engine functions irrespective of the data source. Each textual input feed is processed by the *LeakHawk Core* to measure the sensitivity of the content. The sub-modules within the *LeakHawk Core* will classify each input into one or more pre-defined classes and process them under a rule-base designed for each class. Furthermore, the system evaluates the textual feed for any evidence of a hacking attack against the target in which the system is configured to focus. Ultimately, the system will classify each input feed based on the severity of the incident and utilize the notification module to warn the data owners regarding the data leakage via different means.

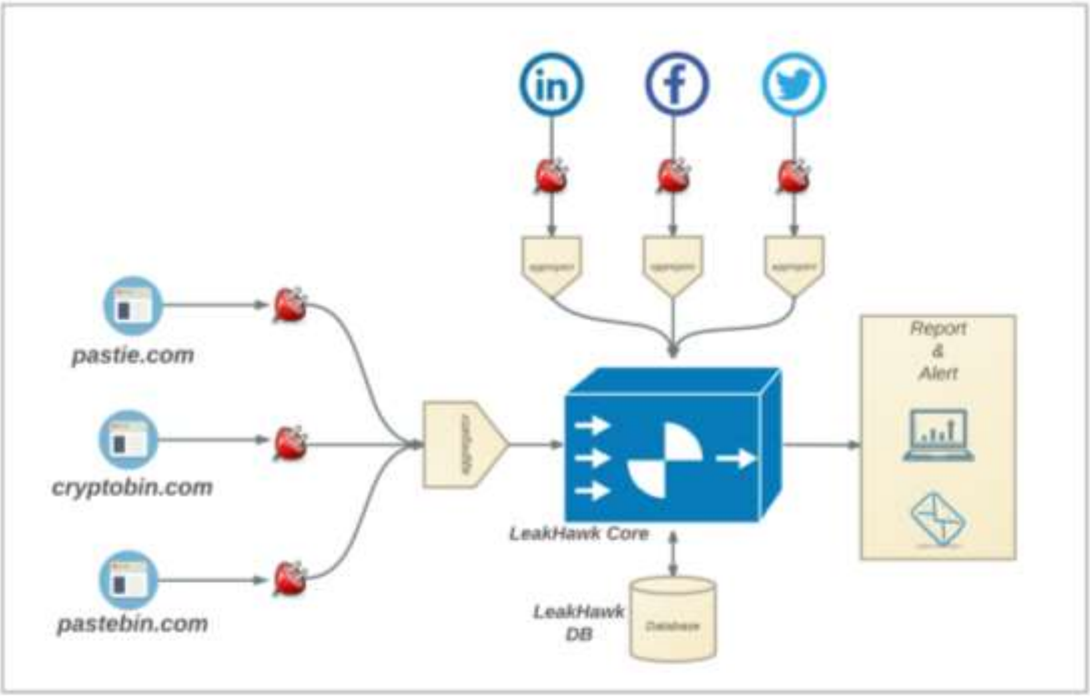


Figure 3-2: High-level architecture of LeakHawk.

3.2. Component Architecture

Figure 3-3 illustrates the component architecture of the proposed monitoring platform. The proposed solution is comprised of multiple components that perform different functionalities. For a single information source, *LeakHawk* performs multi-level filtering and classification procedures to identify and classify data leakages and evidence of hacking attacks.

Each element aims to fulfill a set of architectural, functional and performance requirements. Sensors keep track of the new posts and fetch them while assuring the timeliness and comprehensiveness. When a new post is retrieved, *PRE filter* excludes the non-textual and non-relevant feeds. A particular instance of *LeakHawk* is configured to monitor data leakages related to an entity. Unique identifiers related to the entity are defined as an information template in the *Context Filter*. It executes the configured rule-set for the input feed and extracts only the relevant posts to the entity. Any potential data leakage is identified and categorized by the *Evidence Classifier* and *Content Classifier*. *Synthesis Process* consolidates the results obtained by the classifiers along with the past data retrieved from the database and predicts the severity of the incident.

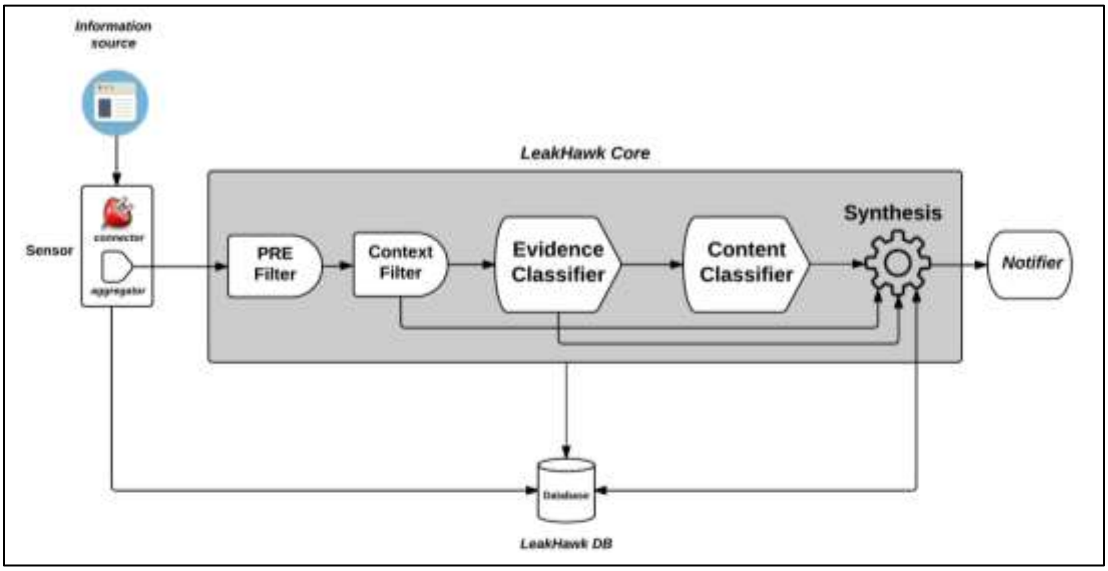


Figure 3-3: The component architecture of LeakHawk.

The proposed platform consists of the following modules:

1. LeakHawk Core
    a. PRE filter
    b. Context filter
    c. Content Classifier
    d. Evidence Classifier
2. Aggregators & Connectors
3. Notification module

3.2.1. LeakHawk Core

LeakHawk Core is the primary module of the proposed monitoring platform. It performs the following set of text analysis tasks:

1. Analyze the textual input for any evidence of data leakage
2. Analyze the textual input for any proof of hacking incident
3. Classify each textual input into pre-defined set of classes based on multiple checkpoints and metadata
4. Evaluate the sensitivity score of a particular input with respect to the parameters defined for each class
5. Predict the sensitivity label for each input

LeakHawk Core consists of four major modules, namely:

- PRE filter
- Context Filter
- Content Classifier
- Evidence Classifier

Figure 3-4 illustrates the arrangement of filters and classifiers within *LeakHawk*. Each component is described in the proceeding sections in detail. Out of the four text-processing entities, *PRE filter* and the *Context filter* utilizes keywords and regular expressions to identify the relevant and irrelevant documents. *Content Classifier* and *Evidence Classifier* use both pattern-based classification procedures and Machine Learning (ML) based classification procedures to identify and classify the sensitive data extracted from the textual documents.

Arrangement of the filters and classifiers can be modified depending on the application of LeakHawk. For example, *Context filter* can be applied in advance to the *PRE filter*, if the percentages of false negatives that are introduced by the *PRE filter* adversely affects the detection rate of *Context filter*.

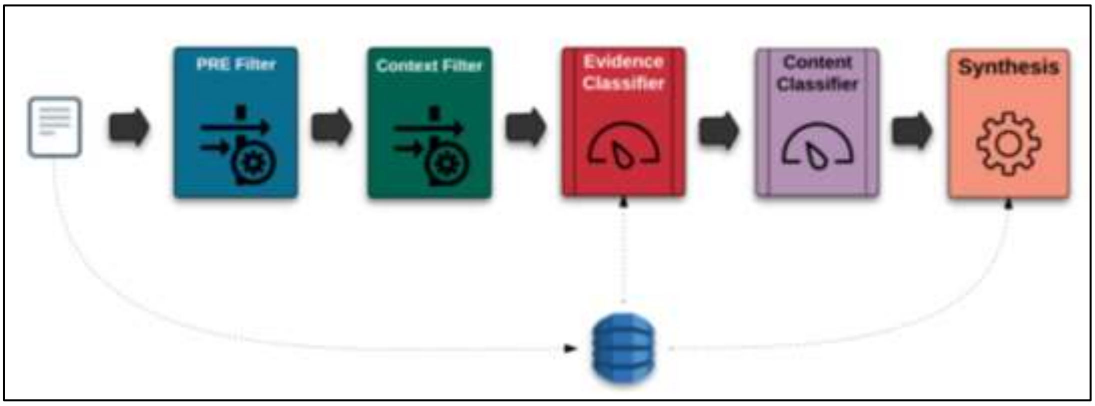


Figure 3-4: Arrangement of Filters and Classifiers.

3.2.2. PRE Filter

Each of the input textual input is passed through the *PRE filter* before processed by the classifiers. The primary objective of the *PRE filter* is to screen-out the inputs, which are non-sensitive in nature, such as video game chat sessions, pornographic content, and torrent information (see Section 0 for further information). Table 3-2 lists the types of data inputs that are identified and filtered out by the PRE filter.

Table 3-2: Irrelevant input data types.

Input type	Description
Trial and empty pastes	Trial pastes are the posts made by users to check the functionality of the pastebin while empty pastes are the posts with empty content.
Non-Textual Pastes	Non-textual documents such as binary files will be excluded from feeding in.
Pornographic content	Pastebin is frequently used to publish links to pornographic content and premium accounts of pornographic websites
Code snippets	Pastebin is mostly used for sharing code snippets. If a comprehensive filter can be built to exclude them, it will significantly reduce the overhead. This exclusion can utilize the “syntax” attribute in a pastebin post.
Game Chats	Pastebin is used by the computer gaming communities for exchanging game cheats, chat sessions, etc.

The usage of *PRE filter* is optional. If used, it reduces the processing weight on the subsequent filters and classifiers as they involve running certain resource intensive logic functions on the data entries.

3.2.3. Context Filter

The filtered output from the PRE filter is passed through the *Context filter*. The *Context filter* is designed to let the administrator or the users of the monitoring platform, to configure the information domain which is used by the *LeakHawk Core* as the context. The context defines the information regarding a particular organization, nation or an individual that is unique for each entity. Thus, the *Context filter* screens out the context-irrelevant information and extracts only the input documents related to the context the system is focused on (refer section 4.4 for details).

The *Context filter* utilizes a set of information templates, which can be used to define the information that the *LeakHawk* look for, in the target data sources. For example, if the *LeakHawk* is utilized by an individual, he/she can configure a template containing his/her unique information domain.

3.2.4. Defining the Information Domain

Characterization of the information domain of a particular organization, pertaining to the sensitivity, must be performed by considering multiple perspectives. It requires the domain knowledge from a business domain expert as well as from an information security expert.

Formulation of the unique identifiers will improve the precision of the monitoring platform. However, to improve the comprehensiveness of the detection rate, it is required to expand the scope, to cover a domain which encompasses the target entity. This will introduce further false positives (reduce the precision) but maximize the recall. Expanding the scope also provides the space for attack forecast and identify trending movements related to a particular target.

Figure 3-5 is an example domain diagram of banks. It is required to concentrate on the related entities within the larger domain such as finance companies and insurance companies. Coarse-grained identifiers are not essential for the relevant organizations.

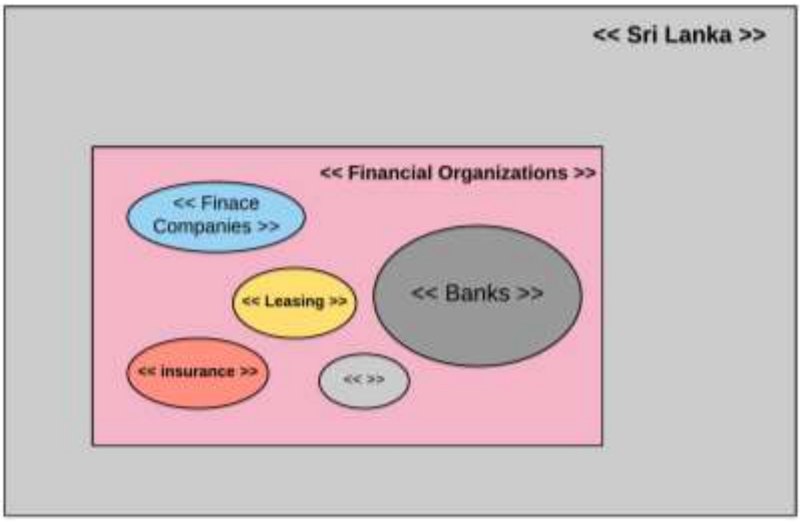


Figure 3-5 : Sample domain diagram.

Named Entity Recognition (NER) technologies use keywords to identify the entities, so if the document does not contain the specific keywords defining the target object, monitoring platform will not consider that post as relevant. For instance, a post with an evidence of an attack may contain the phrase “series of defacement attacks against the government websites of southeast Asia”. Analysis of such a post will not accurately identify the target entities if the system is not configured to capture a larger scope than the unique identifiers.

3.2.5. Information Templates

Defining the information templates pertaining to a particular entity is tedious and mostly a manual task. In order to simplify the effort, *LeakHawk* provide information templates to let the data owners to define the scope. *LeakHawk* incorporates keywords and regular expressions to determine the context of a particular instance. Once the unique identifiers are provided by the user, *LeakHawk* expands the scope by generating possible combinations of the keyword base. Refer Section 4.4 for an example information template.

3.2.6. Content Classifier

Content Classifier classifies each textual input into a set of classes. Each class is defined by a template comprised of multiple checkpoints that evaluate the content. The set of classes is pre-defined, and the list is not exhaustive as the categorization of sensitive content is not comprehensive. Ideally, it should assign each input to one or more of the existing classes. After the class assignment, *LeakHawk Core* executes a set of rule-based checks to identify the sensitive content with respect to the each class. For example, the *Content Classifier* labels a particular input document as a Credit Card Information Dump based on the content and metadata of the document.

3.2.7. Evidence Classifier

The objective of the *Evidence Classifier* is to identify whether the input document indicates a sense of an attack or a sensitive information leakage. For example, if the document contains sufficient evidence related to a Hacktivist involvement such as hacker group announcements, keywords related to a hacking attack, then the *Evidence Classifier* will predict a possible security attack or data leakage incident.

3.2.8. Connectors and Aggregators

For each origin of the information source, a pair of the connector and an aggregator is required. Connectors maintain an uninterrupted connection to each data source. The primary objective of a connector is to fetch all the new posts published on the targeted data origin without any false-negative. Aggregators convert the data input retrieved from the connectors and feed them into the *LeakHawk Core*. A pair of connector and aggregator is defined as a sensor. So a particular sensor will detect a new post being made at a particular target and respond to that by downloading it together with the metadata.

3.2.9. Notification Module

Once a particular input document is analyzed, and the sensitivity label is predicted, notification module will alert the respective data owner or the administrative contact regarding the data leakage incident. *LeakHawk* classifies each relevant document as CRITICAL, HIGH, and LOW.

E-mail and SMS based notifications can be incorporated into the platform. Containment process can be streamlined if both HIGH and CRITICAL grade incidents are notified via both e-mail and SMS while LOW-grade incidents are dispatched via e-mail only.

3.3. Input, Output and Processing Steps

Figure 3-6 illustrates an abstract view of the inputs and outputs the proposed system. The intention of *LeakHawk* is to predict the sensitivity label for a given textual input after determining the relevancy of the input to the pre-defined information domain. Refer Table 3-3 for a sample output for a given input under a given context.

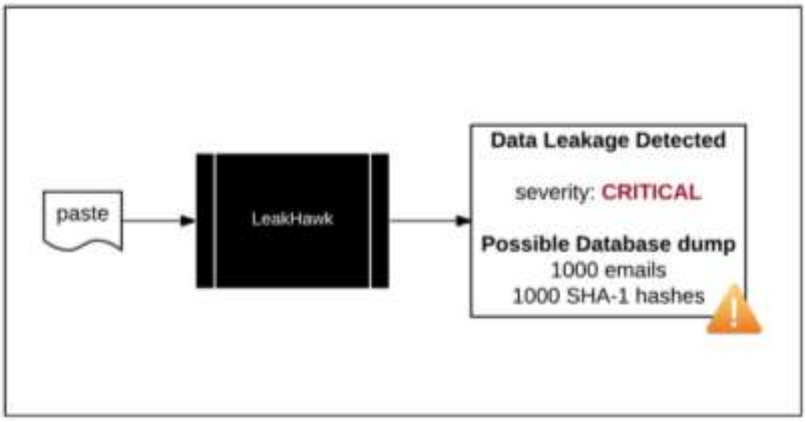


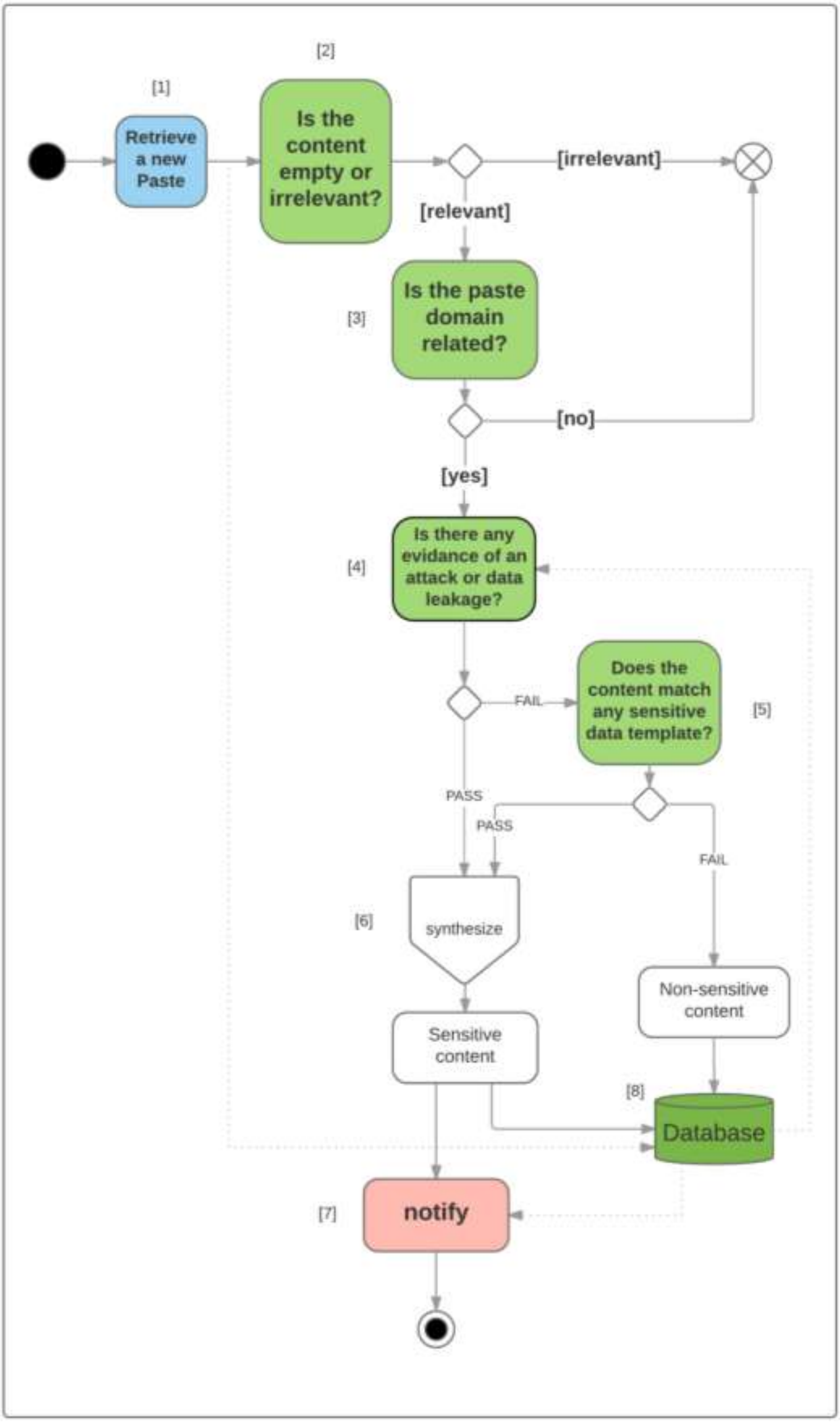
Figure 3-6: Sample output of LeakHawk.

Table 3-3 : Sample Scenario of a Positive Detection

CONTEXT	Sri Lanka
INPUT	A paste posted at www.pastebin.com (paste001.txt)
OUTPUT	paste001.txt contains sensitive information related to the given information domain, with a classification label of HIGH

When a new paste is downloaded, it will be passed through a set of filters and classifiers in order to determine its sensitivity label. As seen in Figure 3-7, a textual document is passed through multiple filters and checkpoints in the classification process as follows:

1. When a new paste is published on a paste site or a post is made on a social media site, the respective connector will download the textual content. The retrieved document is aggregated into the *LeakHawk Core*. Furthermore, the system will extract the metadata and save it into the database along with the text content.
2. Within the *LeakHawk Core,* initially the PRE filter checks whether the content is empty or contain any of the pre-defined patterns of unrelated content. If the document is irrelevant, activity will be terminated. Else, the document is passed via the *Context filter.*



[2]
[1]
Retrieve a new Paste
Is the content empty or irrelevant?
[irrelevant]
[relevant]
Is the paste domain related?
[3]
[no]
[yes]
Is there any evidance of an attack or data leakage?
[4]
FAIL
Does the content match any sensitive data template?
[5]
PASS
PASS
FAIL
[6]
synthesize
Non-sensitive content
Sensitive content
[8]
Database
[7]
notify

Figure 3-7: LeakHawk activity diagram.

3. *Context filter* evaluates the input for context relevancy, extracts the relevant documents, and skip if irrelevant. Else, it is delivered to the both *Context Filter* and the *Evidence Classifier* for further analysis.
4. *Evidence Classifier* recognizes the indications of a hacking attack or the existence of sensitive content. It incorporates the metadata and previous incidents into consideration when making the analysis.
    a. If any evidence is found, it will be directed to the synthesis process irrespective of the result of *Content Classifier*.
    b. If no proof is found, *LeakHawk Core* will depend only on the results from the *Content Classifier*.
5. *Content Classifier* matches the document content with the pre-defined set of classes (e.g., Credit Card Information, Database Dump, etc.). A particular input could match one or more classes depending on the content.
    a. If at least one matching class found, analysis results will be fed into the synthesis process.
    b. If no match is found, the document will be classified as non-sensitive, and respective data is saved in the database for statistical purposes.
6. The synthesis process will analyze the following input combinations:
    a. If both the *Content Classifier* and *Evidence Classifier* are passed (each has a positive result regarding a potential security incident), both the results are synthesized to predict a data leakage incident or a probable attack.
    b. If only the *Content Classifier* is passed, only that analysis result will be incorporated into the final estimation.

    Furthermore, it defines a set of quantitative and qualitative attributes to measure the sensitivity for each class identified in the classification phase.
7. Based on the sensitivity label defined by the synthesis process, *Notification module* will warn the corresponding administrator about the incident.
8. The database stores the following information
    a. Metadata of all the fetched documents
    b. Textual content of each document
    c. Administrative contact of each respective data owner

3.4. Text Classification Process

ML-based classifiers employed in *Content Classifier* and *Evidence Classifier* are based on statistical language processing techniques. They follow a supervised learning approach, which infers a function from a labeled training corpus. As per the standard statistical text classification techniques and the practical applications of such [31], [32], [34], development of the classifiers in *LeakHawk* will follow the methodology illustrated in Figure 3-8 .

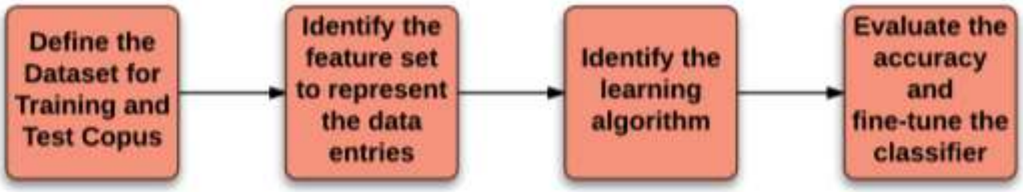

Define the Dataset for Training and Test Copus
Identify the feature set to represent the data entries
Identify the learning algorithm
Evaluate the accuracy and fine-tune the classifier

Figure 3-8: Procedure in building the classifier.

The first step is to define the corpus or a dataset for the classifier. It should contain data for both training and test purposes. A dataset to train the classifiers can be prepared by multiple means. As observed in [34] and [32] a comprehensive crawler can generate the required number of samples for the dataset. If the crawler is correctly used to extract all the possible combinations of data entries, the resulting dataset will achieve the desired state of the real-world data entries.

Once the dataset and desired classes are defined, it is required to identify the feature vectors to map the input documents. A single feature vector entry will represent a single input document as illustrated in Figure 3-9.

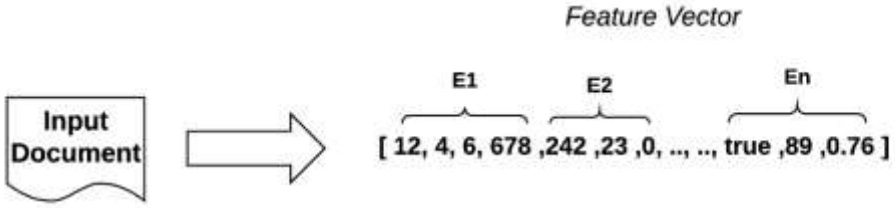


Figure 3-9: Document representation as a feature vector.

Next step would be to find a learning algorithm to build the model for each classifier. Once the classifier is selected, the test set is used to verify the accuracy, precision and recall. As per the results of the test set evaluation, the classification process is enhanced by feature selection and feature extraction criteria.

3.5. Sensitivity Labeling Process

Information extracted from the *Evidence Classifier* and *Content Classifier* along with the sensitivity identification mechanisms will define the sensitivity label of a particular input document. This process is subjective and depends on the risk factors associated with the data possessed by the data owners.

Sensitivity labeling process takes two major points into consideration in determining the sensitivity of a particular data entry:

1. Semantics of the data
2. Magnitude of the data

Semantics define the sensitivity based on the meaning of the content itself. Magnitude fine-tunes the semantic based sensitivity label based on the amount.

For instance, consider a credit card dump containing only a couple of primary account numbers (PAN). PAN itself cannot be utilized for a complete transaction without combining it with an at-least expiry date. Hence, that data entry is labeled as HIGH as it contains confidential information. However, the severity can be escalated to a CRITICAL level, in any of the following scenarios:

1. Number of PANs exposed within a single document is more than a pre-configured value
2. Corresponding expiry dates or any other sensitive authentication data is disclosed together with the PANs.

The sensitivity of the data depends on the classification standards followed by the data owners. Therefore, the definition of classification labels is a manual process. It can incorporate the existing classification criteria in the current practice (refer [44], [45],[46])

Table 3-4 lists the common data types found in pastebin, categorized according to the classification schemes discussed in [44], [45],[46].

Table 3-4: Potential sensitive information available in pastebin.

Confidential Data	**Internal Use**	**General Use**
A username or email address, in combination with a password or security question and answer	User IDs	Public keys
Shared secrets	Electronic or digitized signatures	Email addresses
PINs	Vulnerability information	Names
Cryptographic private keys		Physical addresses
Passwords or credentials		Telephone numbers
Payment Card Information Cardholder name, Service code Expiration date, CVC2, CVV2 or CID value, PIN or PIN block, Contents of a credit card's magnetic stripe		

3.6. Evaluation

An input document could be either sensitive or non-sensitive. Alternatively, *LeakHawk* could classify a particular document as sensitive or non-sensitive. As illustrated in Figure 3-10, there are four possible types of document classes:

1. Originally a sensitive document, correctly classified as sensitive by LeakHawk (True Positive - TP)
2. Originally a non-sensitive document, correctly classified as non-sensitive LeakHawk (True Negative - TN)
3. Originally a sensitive document, incorrectly classified as non-sensitive by LeakHawk (False Negative - FN)
4. Originally a non-sensitive document, incorrectly classified as sensitive LeakHawk (False Positive - FP)

Based on the above possibilities, three key performance measures are defined such as Precision and Recall.

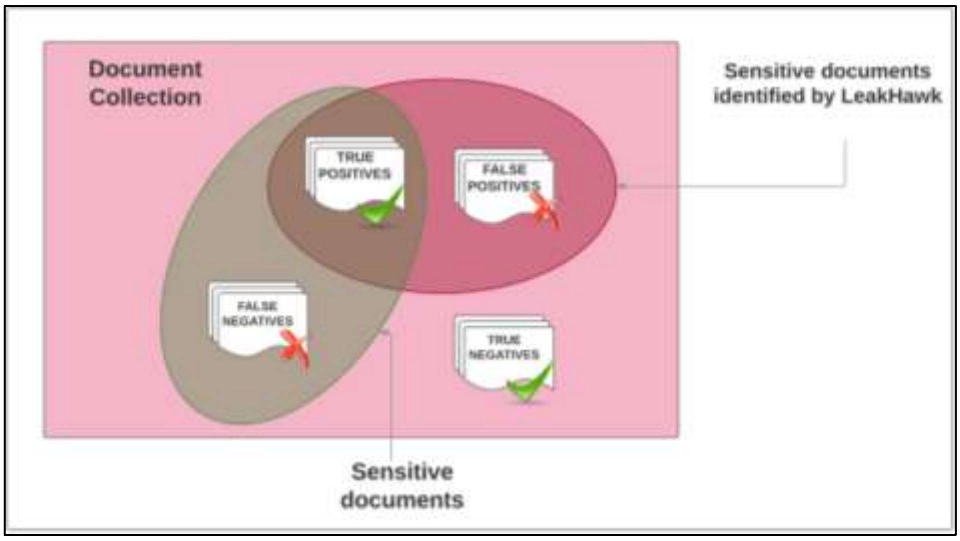


Figure 3-10: Accuracy, Precision, and Recall of LeakHawk.

Precision

Precision is the fraction of correct extractions out of all the extractions made (see equation 3.1). In other terms, precision is the number of true positives out of all the positive extractions made by the system.

$$Precision = \frac{TP}{TP+FP} \tag{3.1}$$

Recall

Recall defined as the fraction of correct extractions made out of the possible relevant data (see equation 3.2). In other terms, recall is the number of true positives out of all the sensitive documents in the data set.

$$Recall = \frac{TP}{TP+FN} \tag{3.2}$$

3.7. Summary

Chapter 3 discussed the design considerations in the development process of *LeakHawk*. It describes the proposed architecture along with the components in detail. The author highlights the methodologies to be adopted for analyzing the performance of the proposed platform in terms of functional and non-functional requirements.

4. PROOF OF CONCEPT IMPLEMENTATION

This chapter describes the implementation aspects of *LeakHawk*, configured as an early detection platform for monitoring sensitive information leakages and evidence of hacking attacks. *LeakHawk* is a proof of concept (POC) of the methodology presented in Chapter 3. Section 4.1 presents the scope of the POC implementation. Section 4.2 to section 4.4 describes the implementation of the filter modules of *LeakHawk* in detail. Section 4.5 and section 4.6 presents the text classification process utilized by the Classifiers and the synthesis process, including the utilized Machine Leaning principles. It describes the granular-level implementation details of the sensitivity classification and ranking process by explaining the process followed by the classifier designed for Credit Card related frauds.

4.1. Scope of POC

In general, this instance of *LeakHawk* will monitor the pastebin for sensitive data leakages and evidence or attacks, targeting the entities based in Sri Lanka. The POC of *LeakHawk* is designed and developed with the following features:

- POC will target only www.pastebin.com. A fully-fledged sensor (combination of a connector and an aggregator) is built for monitoring pastebin for new posts (refer Section 4.2).
- A *PRE filter* was developed to screen out a set of document types, which are non-textual and non-relevant (see Section 4.3).
- A *Context filter* was designed to encompass the possible types of attributes that are related to Sri Lanka (refer Section 4.4).
- An E*vidence classifier* is built to detect all the apparent indications of a security-related incident (refer Section 4.5).
- A *Content Classifier* is built to categorize the input documents into a set of pre-defined classes (refer Section 4.5).
- Qualitative and quantitative attributes are defined to identify the sensitivity label for each identified document during the synthesis process (see Section 4.6).

4.2. Pastebin Sensor

The pastebin sensor is a Java-based application, which queries the pastebin for all the new posts as soon as it is published on the page. New pastes are downloaded to the *LeakHawk* database along with the metadata. Implementation of the pastebin sensor fulfills the following functional and non-functional requirements:

- Timeliness
- Comprehensiveness
- Should not violate the AUP of pastebin

Pastebin provides a scraping Application Programing Interface (API) allowing the users to query for new posts being made at the site. However, the administration does not permit access unless the user is a LIFETIME PRO member. If a regular user queries the page with higher intensity, the server will blacklist the source IP from further querying the archives page. However, without particular intensity in requests, it is not possible to extract all the posts published on the website.

Pastebin scraping API allows access to the newly published posts by querying http://pastebin.com/api_scraping.php. This link is accessible for the whitelisted IPs only [47]. When a query is made, the website responds with the latest posts as a standard JavaScript Object Notation (JSON) object. See Figure 4-1 for a sample output received from pastebin.

```
[   {
        "scrape_url": "http://pastebin.com/api_scrape_item.php?i=vCj9gzTw",
        "full_url": "http://pastebin.com/vCj9gzTw",
        "date": "1466767678",
        "key": "vCj9gzTw",
        "size": "124",
        "expire": "0",
        "title": "Password Dump of SL RSR",
        "syntax": "text",
        "user": "HackTeam"
    },
    {
        "scrape_url": "http://pastebin.com/api_scrape_item.php?i=GE8pXqTZ",
        "full_url": "http://pastebin.com/GE8pXqTZ",
        "date": "14667678564",
        "key": " GE8pXqTZ ",
        "size": "454",
        "expire": "0",
        "title": "Credit Card list for sale",
        "syntax": "text",
        "user": "The$eller"
    },]
```

Figure 4-1: Sample JSON output from pastebin.

As per the analysis conducted over a period of one month (from 27th of April to 26th of May 2016), following statistics were gathered about the intensity of the posts published on the website.

- Average number of posts per minute: 24
- Maximum number of posts made per minute: 89
- Minimum number of posts made per minute: 1

Based on the analysis, it was concluded that it is safe to configure the sensor to download a maximum of 100 posts per minute to cover all the pastes made at the website. Figure 4-2 illustrates an output of the pastebin sensor. As a demo, this instance is configured with a limited set of keywords from the *Context Filter*.

```
************* Reading the configuration file ***************
Key Word List : [Sri Lanka, SriLanka, sinhala, LK,  Colombo,  Ceylon]
Allowed Syntax List : [text, java]
Do you want to apply the context filtering ? [y/n] : n
Current Time: 2016.06.29 at 10:38:13
Scanning page : http://pastebin.com/api_scraping.php?limit=100
FeedEntry [Scrapper Url=http://pastebin.com/api scrape item.php?i=663UZEMt,
Key=663UZEMt, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=GaNfzvCD,
Key=GaNfzvCD, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=fbdhGi8t,
Key=fbdhGi8t, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=C29xLNvt,
Key=C29xLNvt, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=WEc1wBZ4,
Key=WEc1wBZ4, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=HgjEi6ct,
Key=HgjEi6ct, Title=Admin Power Menu v1.0 by Hyuna, Matching Keyword=null,
Syntax=pawn]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=4B5fmY4e,
Key=4B5fmY4e, Title=Ray Donovan Stagione 4, Matching Keyword=null,
Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=FdKjz7Wc,
Key=FdKjz7Wc, Title=ForEylone, Matching Keyword=null, Syntax=java]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=jAH7EPgD,
Key=jAH7EPgD, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=hUiJwacY,
Key=hUiJwacY, Title=ecapeboy4455, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api scrape item.php?i=ah11BvJt,
Key=ah11BvJt, Title=VoidLauncherCrash - CrazyCraft3 v3.0 Minecraftv1.7.10 -
Wed Jun 29 18:07:15 BST 2016, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=iMm8zJPn,
Key=iMm8zJPn, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=wyRKveAs,
Key=wyRKveAs, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=LCz5viGA,
Key=LCz5viGA, Title=, Matching Keyword=null, Syntax=text]
FeedEntry [Scrapper Url=http://pastebin.com/api_scrape_item.php?i=Fxe9QJA2,
Key=Fxe9QJA2, Title=, Matching Keyword=null, Syntax=text]
```

Figure 4-2: Partial output of pastebin sensor.

Initially, the sensor downloads 100 entries from the pastebin and in the next run, it will match the unique identifier of the top most post from the earlier list and download only the posts pasted after that entry. That logic will preserve the comprehensiveness of the sensor and make sure that all the posted entries are extracted.

LeakHawk downloads all the pastes to a local repository along with the metadata listed in Table 4-1 .At each cycle, downloaded data entries are passed through the series of filters, and only the relevant entries will be extracted for the text classification process.

Table 4-1: Extracted metadata from pastebin posts.

Metadata ID	**Description**	**Example**
entry_key	Unique identifier of a particular post	vCj9gzTw
entry_url	Full URL to access the post in the website	http://pastebin.com/api_scrape_item.php?i=vCj9gzTw
entry_title	Subject of the post	Password Dump of SL RSR
entry_file	Content of the file	“Test data.”
entry_matchingKeyword	Matched keyword in the *domain filter*	HACK
entry_user	User who posted the content	HackTeam

4.3. Implementation of PRE Filter

The *PRE filter* screens out the irrelevant inputs that are certainly non-sensitive. The list of non-sensitive content types is identified by analyzing the training corpus downloaded from the pastebin. Pre-processing procedures carried out at the *PRE filter* reduce a set of processing steps performed in the proceeding steps of the *LeakHawk*.

Table 4-2 lists the types of data inputs, which are identified and filtered out by the PRE filter and the implementation aspects of each. To improve the accuracy of the filter and to prevent it from screening out any relevant and sensitive content, it is required to focus only on the keywords, which clearly identify the non-sensitive and unrelated content.

Table 4-2: Sample content of the PRE filter

Data Type	Implementation details
Trial and empty pastes	Search the text body for the keywords: “test”, “demo.”
Non-Textual Pastes	Output of the “file” command in Linux determines whether a file is textual or non-textual (e.g. binary file)
Pornographic content	A list of keywords is formulated to identify the documents containing pornographic content.

4.4. Implementation of Context Filter

In general, *Context filter* defines the unique identifiers pertaining to a particular organization or an individual who uses the monitoring platform to monitor for data leakages. Use of the *Context filter* for the monitoring process is optional and depends on the interest of the user. If the filter is not used, *LeakHawk* will function as an early detection platform for detecting information leakages and evidence of attacks irrespective of a target domain. In such an application, the users can incorporate the NER techniques to identify the entities who have been subjected to the security incident. The goal of NER system is to determine all textual mentions of the named entities [46]. Previous work by Varish et al. [48] in the development of an information extraction system for extracting information about security vulnerabilities from web text, encompass a standard NER tool, OpenCalais [49] with greater precision.

LeakHawk POC is configured to monitor sensitive information leakages and evidence of hacking attacks targeting Sri Lanka. An *Information Template* defined for Sri Lanka, with respective examples is illustrated in Table 4-3.

Table 4-3: Information template for Sri Lanka.

Identifier	Description	Examples
Identification names	A particular country can be identified using different terms. Furthermore, it could consider those identifiers in other related languages (pastebin does have the Unicode supports) Names of the major cities can be mentioned instead on the country name In some cases, the country is referred with indirect terms	Sri Lanka, Lanka, Ceylon, ලංකා, LK (could introduce lot of false-positives) Colombo South Asia, south Asian country

Identifier	Description	Examples
Unique communities	Sometimes without mentioning the country name, distinct communities are targeted. This should not include the domains, which could add a lot of false positives.	Sinhala Sinhalese Buddhists
Language Detection	Language detection APIs are available such as detectlanguage [50].	English, Sinhala
Domain names	Use of regular expressions to identify the domains names related to Sri Lanka. e.g. government websites (domain name ending with gov.lk) LK domains in general (domain names ending with .lk) Domain names containing Identification names related to Sri Lanka	www.president.gov.lk example.lk example.lk.com srilanka.com lanka.org
Unique identifier formats of the citizens	When a large community is targeted, unique identifiers could be exposed.	National Identity Card number Driving License Number Passport Number
IP addresses related to Sri Lanka	In certain cases, the IP addresses within the Sri Lanka could be involved in a particular attack. WHOIS database [51] can be utilized to identify the location of a particular IP address.	112.134.100.10 222.165.128.4
Credit / Debit Card ranges	Bank Identification Number (BIN) ranges are defined uniquely to identify each issuing bank in the world. But these databases are not freely available. A survey was conducted by the author by querying all the major banks in Sri Lanka regarding their BIN ranges. This list should also cover any BIN ranges of the local payment brands (e.g. LankaPay)	
Popular characters	This list may contain some popular characters who could be subjected to an online attack.	<<president>> <<prime minister>> <<ex-presidents>> <<army commanders>>
Major organizations and conglomerates	Certain posts may directly mention the organization names and conglomerates without mentioning the country name. So it is safe to search for those names separately.	Mobile and Internet service provider names Sri Lankan organizations (Banks, Telecommunication companies, Insurance, Finance, Textile, etc.) Conglomerates (Cargills Ceylon, Keels, Aitken Spence, Hemas, etc.) Famous TV channels
Other		LTTE, genocide, civil war

In general, converting the keywords to lowercase is done to reduce the number of false-negatives. In some cases, preserving the case of the keywords is required, for instance, LK. The defined attributes for a particular instance of *Context Filter* are implemented using keyword lists and regular expressions. At each download cycle, the corresponding thread will execute those logics, and only the positive matches are extracted and saved into a new table in the database. The models developed by the *Evidence Classifier* and the *Content Classifier* will only execute on the entries in that table.

4.5. Classification and Synthesis Process

The *Evidence Classifier* and the *Content Classifier* are the primary components of the *LeakHawk Core*. Both machine learning and keyword/regular expression based classification tasks are incorporated in these classifiers. Based on a sample data set, each filter utilizes its own feature set and classifier(s) and produce models to predict the classes of new data entries.

4.5.1. Defining Class Labels

Prior to the selection of the corpus and the checkpoints, it is required to identify the desired outcomes of each classifier.

The Evidence Classifier and each Content Classifier labels any input document ***d*** *as either positive or negative such that the resulting training set will have* ***m*** *number outputs:*

```
Input:
A document d
A fixed set of classes C={positive, negative}
A training set of m hand-labeled documents (d1,c1),....,(dm,cm)
Output:
A learned classifier γevidence :d → c
```

For the purpose of POC, the author has categorized the information into nine different classes based on the analysis conducted on the security incidents related to pastebin. The list of categories is not exhaustive and requires more comprehensive classification to cover all the types of sensitive data. However, the platform does not skip any data entry related to the context. When a data entry is not classified under any of the defined classes, that document will be tagged as unclassified, altogether making ten classes. Table 4-4 lists the desired class labels for the *Evidence Classifier* and *Content Classifier*.

Table 4-4: Class labels for classification.

Classifier	Method	Class Labels
Evidence Classifier	Binary classification	P: Positive N: Negative
Content Classifier	Binary classification (for each sensitive data type)	P: Positive N: Negative
	Multi-class classification (one-vs-all approach)	CC: Credit Card information UC: User Credentials DB: Database dump DA: DNS Attack EO: Email only PK: Private keys EC: Email conversation CF: Configuration files WD: Website Defacement

4.5.2. Defining Dataset

Both *Evidence Classifier* and the *Content Classifier* were trained with the same dataset (corpus). However, the number of entries for each classifier instance varies based on the type of outcome. It contained the data entries collected from the following methods and sources:

1. The connector module of *LeakHawk* along with the corresponding aggregator was used to download the new data entries published in the pastebin archive page [15]. This method only applies for the new pasted being made, not to gather old pastes.
2. A crawler specifically built to download pastebin posts for a given keyword. This method involves manual pre-processing steps to bypass some constraints enforced by the website.
3. Archived pastes downloaded from some of the Internet archive sites.
    a. archive.org [52] contains older posts between October 2013 to July 2014.

   b. Pasbdmp.com [53] archives the daily posts but downloading the data is a tedious task. However, it is possible to access some deleted posts as the site archives all the posts in real-time.

4. To reduce the adverse effects of dimensionality, certain types of inputs need to be artificially produced by truncation and rebuild.

By utilizing the above methodologies, a corpus of 1193 positive samples was formulated to be as the training set for each classifier. The corpus of the *Evidence Classifier* contained 940 positive samples. The corpus for the *Content Classifier* included 1193 positive sample as a whole. However, for each binary classifier, positive samples vary. Refer Table 4-5 for the number of positive samples per each class. Negative samples per each training data set are selected from the above dataset. At each level of training, different numbers of data entries are selected until the best possible combination is met.

Table 4-5: Positive samples of each binary classifier.

Binary Classifier Name	No of positive samples
CC	109
UC	265
DB	195
DA	52
EO	73
PK	25
EC	18
CF	164
DA	292

4.5.3. Manual Class Labeling Process

Once the dataset is prepared, each data entry is manually labeled to match the respective class. Section 4.5.3.2 introduces the heuristic checkpoints, which are used to label each data point manually. The process of manual labeling was conducted by a set of domain experts in the field of information security.

4.5.3.1. Heuristic Checkpoints

When a domain expert is classifying a particular data entry into a pre-defined class, he/she will utilize the values of certain checks conducted on the data points. These checkpoints are heuristic based and converted into respective feature spaces at the machine-learning phase.

4.5.3.2. Heuristic Checkpoints of the Evidence Classifier

Table 4-6 presents the heuristic checkpoints used by the *Evidence Classifier*. The defined checkpoints are formulated targeting the attributes of the pastebin (refer Section 2.3.2 for the attributes of a pastebin post). However, most of the checkpoints are generalized to match other related PasteBin sites. The majority of the features associated with each checkpoint is based on the textual content and applicable to any textual input.

Table 4-6: Heuristic checkpoints of the evidence classifier.

Attribute	Heuristic ID	Description
User	E1	Does the user, seems suspicious? • GUEST or registered • History of the user • Percentage of related incidents
Subject	E2	Is there any evidence of a hacking attack on the subject?
	E3	Are there any signs of usage of a security tool?
	E4	Are there any signs of security vulnerability?
	E5	Evidence of a Hacktivist movement? • Hacking group names • Hacking group signatures • #op
Content	E6	Is there any evidence of a hacking attack in the body text?
	E7	Are there any signs of usage of a security tool in the body text?
	E8	Are there any signs of security vulnerability in the body text?
	E9	Proof of a Hacktivist movement in the body text? • Hacking group names • Hacking group signatures • #op

4.5.3.3. Heuristic Checkpoints of Content Classifier

The purpose of the *Content classifier* is to classify the input textual documents to a set of the pre-defined classes. For the POC implementation, nine different classes were defined as follows:

- CC: Credit Card information
- UC: User Credentials
- DB: Database dump
- DA: DNS Attack
- EO: Email only
- PK: Private keys
- EC: Email conversation
- CF: Configuration files
- WD: Website Defacement

Table 4-7 lists the heuristic checkpoints defined for the identification of Credit Card related frauds. Table 4-8 lists the heuristic checkpoints defined for the identification of credential compromises. Likewise, checkpoint heuristics are identified for all the classes. During the manual classification process, human agent verifies the class by checking these checkpoints manually. The next section describes how the same process is implemented to use under the machine-learning phase using feature sets.

Table 4-7: Heuristic checkpoints for Credit card related frauds

Attribute	**Checkpoint ID**	**Description**
Subject	CC1	Does the subject contain any of the top-10 credit card related n-grams? • Percentage of unigrams • Percentage of bigrams • Percentage of trigrams
	CC2	Does the subject contain card fraud related terms?
Content	CC3	Does the body text include Credit Card numbers?
	CC4	Does the body text contain any of the top-10 credit card related n-grams? • Percentage of unigrams • Percentage of bigrams • Percentage of trigrams
	CC5	Does the body text contain credit card related terms?

Table 4-8: Heuristic checkpoints for credential compromises.

Attribute	Checkpoint ID	Description
Subject	UC1	Does the subject contain any of the top-10 credential dump related n-grams? • Percentage of unigrams • Percentage of bigrams • Percentage of trigrams
Content	UC2	Does the body text follow a pattern? • Number of emails • Number of hashes • Email per line • Hashes per line
	UC3	Does the body text contain any of the top-10 credential dump related n-grams? (first 10 lines) • Percentage of unigrams • Percentage of bigrams • Percentage of trigrams e.g. Password, pwd, pw dump, username, user, credential, etc.
	UC4	Does the body text contain any evidence of a 'key' to the following dump?

4.5.4. Feature Set for Evidence Classifier

For each checkpoint, a set of features is required to be identified. Each input document is represented as a feature-value vector for the classifier.

Checkpoint E1: Does the user seem suspicious?

Analysis conducted on the corpus suggests that most of the security incidents are exposed via registered user accounts. So that metadata based heuristic can be denoted by a feature. At the aggregation layer, all the relevant metadata is stored in the database. So the respective feature value can be extracted. If the user of a particular post has involved in an earlier incident, there is a high possibility that the current post is a relevant one. Based on this analysis, three features are extracted as follows:

1. Is the user registered or GUEST user?
2. Has he been involved in earlier incidents?
3. The frequency of the earlier posts related to Sri Lanka?

Checkpoint E2 & E6: Is there any evidence of a hacking attack?

When data leakages or hacking attacks are exposed as a public note, it is very common that the file names (attributed by *subject* in a pastebin post) and the text body contain keywords and patterns, which suggest possible sensitive information. For instance, a set of names of files that contain confidential information is shown in Figure 4-3.

```
CC_091_credit_card_leak_2016_#bentthimble.txt
CC_120_visa_hacked_by_JOKER..txt
DB_030_hu.edu.pk_database_leaked_by_team_IHC
DB_093_DB_dump
DB_185_preferate.net_Database_leaked_by_installer.txt
UC_002_1000_Email___Password_users_Database
UC_005_#100k_Hacked_Facebook_Accounts
UC_030_50+_email_logins_-_Leaked_by_Vaxx
UC_069_Dropbox_Email_Password_Leak
UC_109_fresh_credentials_nov2014.txt
UC_185_#OP_PayBack_To_China_-_Free_The_Animals_leak
UC_246_User_Pass_Email_Dump
```

Figure 4-3: Sample file names of sensitive files.

After extracting the common unigrams that are used when exposing such content, two features were defined such as the *presence of hacking related unigrams in the subject* and the *count of hacking related unigrams in the subject*. Figure 4-4 illustrates a sample extraction process carried out for identifying the documents with evidence of data leakages or hacking attacks by analyzing the subject.

```
$ for i in *; do;  ls "$i" | grep -oi
"hacked\\|hackd\\|leak\\|leaked\\|pwned\\|pwnd\\|dump\\| {MORE}" | wc -l ; done
```

Figure 4-4: Extraction of entries with related unigrams in subject labels.

In terms of the text body, the analysis conducted on the content of 1193 positive samples of text corpus suggested that generally, only the first set of lines contain related keywords in the text body. For instance, see Figure 4-5 where the actual sensitive data entries start after the line number 19. Therefore, the presence of hacking related bigrams in the first ***n*** lines of the text body and the count of hacking related bigrams in the first *n* lines of the text body are selected as features. Based on the corpus analysis, the average value of ***n*** was identified as 20.

```
nalinda@Asteroid ~/temp/names $ head -n30 UC_069_Dropbox_Email_Password_Leak
***** DROPBOX HACKED *****

6,937,081 DROPBOX ACCOUNTS HACKED
PHOTOS - VIDEOS - OTHER FILES

MORE BITCOIN = MORE ACCOUNTS PUBLISHED ON PASTEBIN
As more BTC is donated , More pastebin pastes will appear
To find them, simply search for "DROPBOX HACKED" and you
will see any additional pastes as they are published.

FIRST TEASER - 400 DROPBOX ACCOUNTS Just to get things going...

SEND BTC DONATIONS TO 1CqjUQocCJiqvNwRvgt8pknxwa76typFPj

BACK AND CHECK PASTEBIN FOR NEW DROPBOX DROPS
THE MORE BTC DONATED WILL REFLECT HOW MANY MORE LOGIN AND PASSWORDS
ARE RELEASED PUBLIC.

dunnglendaj@yahoo.com:
joyce1717 vanbartley@aol.com:
aaron1 dtherealtor@gmail.com:
serena1023 bobc2799@yahoo.com:
Heather9931 bencoxhomes@gmail.com:
bencox88 ingrid.soluaga@gmail.com:
rionoe mmitchell@interorealestate.com:
Cooper11 dagilismichael@yahoo.com:
lr!stom8 Lmeyer@EnvisionInvestors.com:
Brokerap1 theronparker1@yahoo.com:
```

Figure 4-5: Sample text body of an email dump.

Figure 4-6 illustrates a sample extraction process carried out for identifying the documents with evidence of data leakages or hacking attacks by analyzing the subject.

```
$ for i in DATASET/*/*; do;  grep –oiE
"hacked\|hackd\|leak\|leaked\|pwned\|pwnd\|dump\| {MORE}" | wc –l; done
```

Figure 4-6: Extraction of entries with related unigrams in text body.

Checkpoint E3 & E7: Is there any evidence of usage of a security tool?

When a name of a security tool or pattern of a dump of a security tool is evident in the text body, there is a high possibility that the data entry is related to a data leakage or hacking incident. For example, a post may mention about hacking tools such as *Kali Linux* or *SQLmap,* which infer a possible exploitation of vulnerability.

Checkpoint E4 & E8: Is there any evidence of security vulnerability?

When a security vulnerability is mentioned, there is a high possibility that the post is published in the context of data leakage or hacking attack. For example, a post may mention about system vulnerabilities such as *Cross-Site Scripting* or *SQL injection,* which infer a possible exploitation of vulnerability.

Checkpoint E5 & E9: Evidence of a Hacktivist movement or targeted attack?

A considerable amount of the sensitive information leakages and notifications of evidence of attacks are related to Hacktivist movements. Such posts contain one or more of the following attributes:

1. Major Hacktivist group names (e.g., ANONYMOUS, LULZSEC, etc.)
2. Hacktivist group slogans (e.g., "We do not forgive")
3. Evidence of a targeted movement (e.g., #OPSriLanka and #OPUSA)

The occurrence of such phrases (binary feature) and count of the occurrences (multi-value) was selected as features. Table 4-9 summarizes the feature set used for the *Evidence Classifier.*

Table 4-9: Feature set of evidence filter.

Is the user registered or GUEST user?
Has he been involved in earlier incidents?
The frequency of the earlier posts related to Sri Lanka?
Presence of hacking related bigrams in the subject
Term frequency of hacking related bigrams in the subject
Presence of hacking related bigrams in the first 20 lines of the text body
Term frequency of hacking related bigrams in the first 20 lines of the text body
Presence of a name of a major hacking tool
Is there a mention of a security vulnerability
Presence of a hacker group names
Presence of hacker group signatures
Presence of terms related to a Hacktivist movement

4.5.5. Feature Set for Content Classifier

This section describes the development process incorporated in defining the characteristics associated with each checkpoint for the *Content Classifier*. To limit the space, we explain only the granular level implementation details of the classifier designed for identifying Credit Card related data leakages.

CC1 & CC5: Does the file contain any of the top-10 credit card related n-grams?

Every class has its unique set of word-unigrams, word-bigrams or word-trigrams. In the POC, the author used top *k*-word n-grams of the positive class as features. By utilizing a Python script which was developed using the NLTK toolkit [54], was used to identify the n-grams of the positive training dataset. Figure 4-7 illustrates word cloud of the top 50 unigrams of the CC positive class. Figure 4-8 illustrates the word cloud of top 35 bi-grams of the CC positive class. Figure 4-9 shows the top 20 trigrams of the CC positive class.

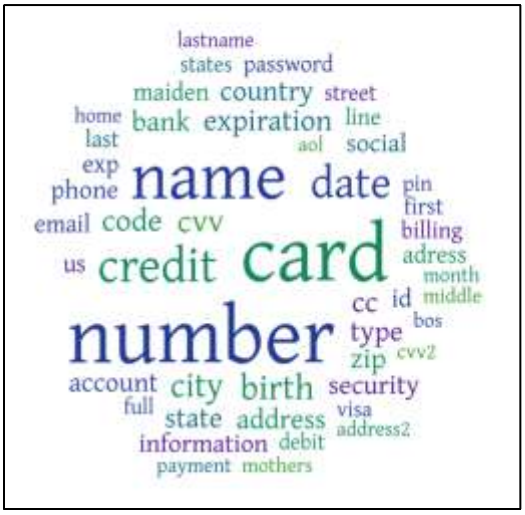


Figure 4-7: Top 50 unigrams of the CC class.

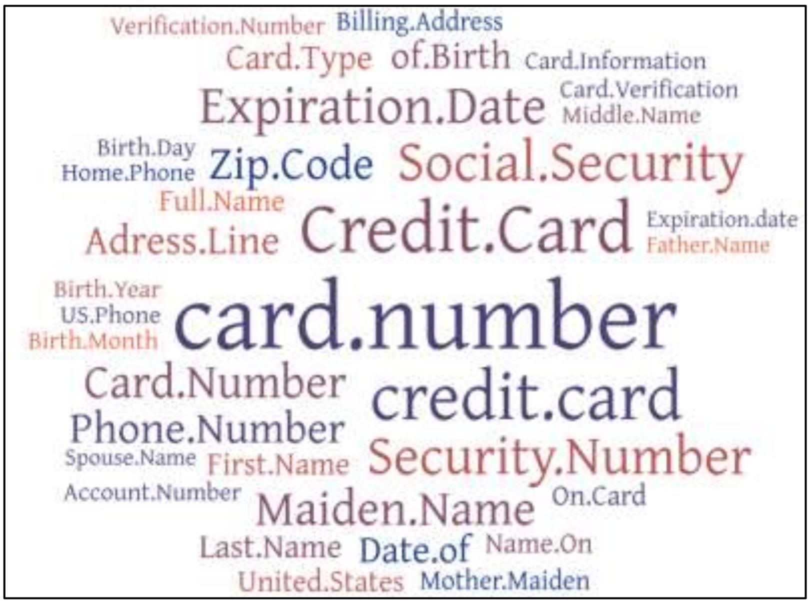


Figure 4-8: Top 35 bi-grams of the CC class.

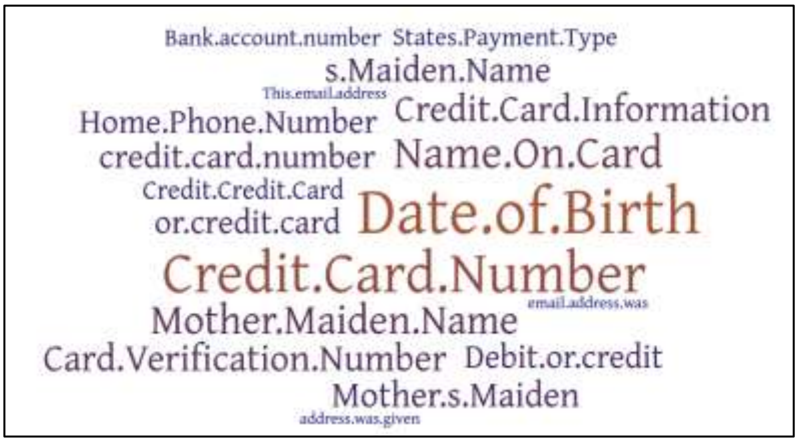


Figure 4-9: Top 20 trigrams of the CC class.

In general, each n-gram can be used as a feature for the classifier. However, to reduce the dimensionality of the feature space, only the n-grams with better TF-IDF rating were selected as features. For certain n-grams, the case is preserved, for instance, CVV and CVC.

CC2: Does the subject contain card fraud related terms?

Although the top n-grams list covers most of the features, it may exclude certain specific terms related to Credit Card frauds. Both the subject and the body text may contain those terms. We used the occurrence of those words as a feature. Following code snippet illustrates the extraction of the number of terms related to card frauds from the subject name as well as text body (see Figure 4-10).

```
$for i in DATASET/*/*; do;  ls "$i" | grep -oi
"card_dump|working_card|cc_dump|skimmed\| {MORE}"  | wc -l; done

$ for i in DATASET/*/*; do;  grep -oiE
"card_dump|working_card|cc_dump|skimmed\| {MORE}" "$i" | wc -l; done
```

Figure 4-10: Extraction of entries with card fraud related n-grams.

CC3: Does the body text contain Credit Card numbers?

We used a generic credit card number detection regular expression and customized to include all the possible credit card number types in the training set (see Figure 4-11).

```
[2-6][0-9]{3}([ -]?)[0-9]{4}([ -]?)[0-9]{4}([ -]?)[0-9]{3,4}([ -]?)[0-
9]{0,3}[?^a-zA-Z]?
```

Figure 4-11: Custom regular expression for Credit Card numbers.

The number of matches of the above regular expression is taken as a separate feature.

CC5 & CC6: Does the body text contain credit card related terms?

Mostly used Credit Card related terms can be categorized as follows:

- Names of the payment brands (e.g., VISA, Mastercard, JCB, AMEX, American express, Discover, and Diners Club)
- Names of the attributes of cardholder data other than primary account number (e.g., Cardholder Name ,Expiration Date, and Service Code) [55]
- Names of the sensitive authentication data (e.g., Full track data, CVC2, CVV2, and CID) [55]

Occurrences of these terms are unique to the Credit Card related posts. So the existence and frequency of these terms are considered as features. Table 4-10 lists the features associated with the Content Classifier for Credit Card related frauds.

Table 4-10: Feature set of CC content filter.

TF-IDF ("card")
TF-IDF ("name on")
TF-IDF ("credit card")
TF-IDF ("card number")
TF-IDF ("maiden.name")
TF-IDF ("expiration date")
TF-IDF ("zip code")
TF-IDF ("Date of Birth")
TF-IDF ("Credit Card Information")
TF-IDF ("Credit Card Number")
number of matches for the regular expression of Credit Card number
Term frequency of related words of "expiration date"
Term frequency of related words of "CVV"
Term frequency of all the card data fraud related terms in the subject
Term frequency of all the card brands related terms
number of digits in the document
number of characters in the document
number of alphanumeric characters in the document
percentage of digits in the document
percentage of characters in the document

As described in Section 4.8, a set of features is identified per class, such that the data entries belong to that class can be uniquely identified using that feature set. Feature identification for the other set of classes follows the same procedure to determine the essential features. Furthermore, each data class has its own unique features in the data entries, which are needed to be included in the feature vector.

4.5.6. Classification

Once the feature sets are prepared, the next step is to select a classifier to combine the heuristics. The most suitable classifier is selected such that the precision and the recall values are maximized [32]. Experiments were carried out with a variety of classifiers such as support vector machines, decision tree based classifiers, and Naive Bayes classifiers.

Weka [56] is one of the major machine learning software written in Java. WEKA is a product of the University of Waikato (New Zealand) developed by combining implementations of existing machine learning techniques as a suite. Weka provides a GUI for producing visual results and has a general API to combine its functionalities with other applications. During this research project, Weka GUI was used (Weka developer 3.9) for building and testing the classifier models and weka API was utilized in the final product to classify the new instances.

4.5.7. Identifying Credit Card Dumps with CC Classifier

This section describes the text classification process followed in this project by demonstrating the development of the CC classifier. The purpose of the CC classifier used in *LeakHawk* is to identify the credit card related sensitive information leakages. In the training phase, It combines the features related to the credit card fraud related heuristics and classifies any input document into one of the two classes: positive or negative. The positive class defines the credit card related sensitive documents and where the negative class includes both credit card related non-sensitive documents and sensitive documents not related to credit card frauds.

Based on the feature set described in Section 4.5.5, values for 21 features were calculated as illustrated in Figure 4-12.

Once the feature matrix is prepared, it was fed into the classifier as an Attribute-Relation File Format (ARFF) file (see Figure 4-13). An ARFF file is an ASCII text file that describes a list of instances sharing a set of attributes which is the default input file format used by Weka [57]. Based on this ARFF file, different classifiers were evaluated for maximum recall and precision using Weka.

Figure 4-14 shows the evaluation results under the Naïve Bayes Classifier, while Figure 4-15 illustrates the results under the Random Forest classifier.

Under the Naïve Bayes Classification method, the *CC Classifier* has achieved an accuracy score of 86% while Random Forest resulted in 100% for the training sample. Maximum *Precision* and *Recall* achieved under Naïve Bayes was respectively 0.981 and 0.486 for positive identification of CC dumps. For the training corpus, Random Forest resulted in maximum attainable values for *Precision* and *Recall.*

```
152,154,148,0,2,149,10874,50996,61870,17.00,82.00,149,226,63,149,142,93,144,81,81,291
8,8,8,0,1,8,566,2726,3292,17.00,82.00,8,16,8,0,0,0,8,0,0,16
400,0,2,1,0,1,9242,411,9653,95.00,4.00,0,0,0,0,0,0,0,0,0,3
2,2,2,1,2,2,63,304,367,17.00,82.00,2,3,2,0,0,0,0,0,0,3
2,2,2,1,2,2,63,304,367,17.00,82.00,2,3,2,0,0,0,0,0,0,3
8,8,8,0,1,8,563,2705,3268,17.00,82.00,8,16,8,0,0,0,8,0,0,16
38,2,5,0,27,2,1382,2542,3924,35.00,64.00,2,21,2,0,0,0,0,0,0,22
10,22,11,0,1,11,963,4818,5781,16.00,83.00,11,22,0,11,11,0,11,11,11,33
9,9,9,0,9,9,439,1502,1941,22.00,77.00,9,1,9,0,0,0,0,0,0,1
1,1,1,1,0,1,46,395,441,10.00,89.00,0,2,0,0,0,0,0,0,0,2
107,100,100,0,100,94,8328,29886,38214,21.00,78.00,94,94,94,0,6,85,88,0,0,94
11,28,0,1,14,14,437,628,1065,41.00,58.00,0,1,0,0,0,0,0,0,0,1
6,6,6,0,6,6,306,1099,1405,21.00,78.00,6,1,6,0,0,0,0,0,0,1
20,0,20,0,0,20,500,60,560,89.00,10.00,0,0,0,0,0,0,0,0,0,0
3,3,3,0,3,3,133,385,518,25.00,74.00,3,0,3,0,0,0,0,0,0,0
7,7,7,0,9,7,293,827,1120,26.00,73.00,7,0,7,0,7,7,0,0,0,0
9,9,9,0,2,9,625,2940,3565,17.00,82.00,5,4,5,4,5,5,4,0,4,4
24,35,26,0,1,26,2472,11465,13937,17.00,82.00,18,14,1,18,9,55,8,6,8,34
7,7,7,1,2,11,243,1344,1587,15.00,84.00,0,3,0,0,0,0,0,0,0,4
7,7,7,1,2,11,243,1344,1587,15.00,84.00,0,3,0,0,0,0,0,0,0,4
3,3,3,1,3,6,161,517,678,23.00,76.00,3,7,0,0,0,0,3,0,0,7
2,2,2,1,0,0,92,495,587,15.00,84.00,0,2,0,0,0,0,0,0,0,3
6,4504,37,62,1,0,1,0,0,0,0,0,0,4,CC
143,85,83,1,69,33,6459,22925,29384,21,78,0,0,0,0,0,0,0,0,0,1,CC
36,11,12,1,3,10,1230,2259,3489,35,64,0,0,0,0,0,0,0,0,0,4,CC
27,5,5,1,3,3,1104,1882,2986,36,63,0,1,0,0,1,0,0,0,1,3,CC
```

Figure 4-12: Feature vector for CC classifier.

```
nalinda@Asteroid /media/nalinda/Win/Dropbox/Training_Set/original/DATASET/combi/CC $
cat CC.arff
@relation CC-valonly

@attribute CC1 numeric
@attribute CC2 numeric
@attribute CC3 numeric
@attribute CC4 numeric
@attribute CC5 numeric
@attribute CC6 numeric
@attribute '{#N}' numeric
@attribute '{#L}' numeric
@attribute '{#A}' numeric
@attribute NP numeric
@attribute CP numeric
@attribute CC7 numeric
@attribute CC8 numeric
@attribute CC9 numeric
@attribute CC10 numeric
@attribute CC11 numeric
@attribute CC12 numeric
@attribute CC13 numeric
@attribute CC14 numeric
@attribute CC15 numeric
@attribute CC16 numeric
@attribute @@class@@ {CC,Non-CC}

@data
152,154,148,0,2,149,10874,50996,61870,17.00,82.00,149,226,63,149,142,93,144,81,81,291
8,8,8,0,1,8,566,2726,3292,17.00,82.00,8,16,8,0,0,0,8,0,0,16
400,0,2,1,0,1,9242,411,9653,95.00,4.00,0,0,0,0,0,0,0,0,0,3
2,2,2,1,2,2,63,304,367,17.00,82.00,2,3,2,0,0,0,0,0,0,3
2,2,2,1,2,2,63,304,367,17.00,82.00,2,3,2,0,0,0,0,0,0,3
8,8,8,0,1,8,563,2705,3268,17.00,82.00,8,16,8,0,0,0,8,0,0,16
38,2,5,0,27,2,1382,2542,3924,35.00,64.00,2,21,2,0,0,0,0,0,0,22
10,22,11,0,1,11,963,4818,5781,16.00,83.00,11,22,0,11,11,0,11,11,11,33
9,9,9,0,9,9,439,1502,1941,22.00,77.00,9,1,9,0,0,0,0,0,0,1
```

Figure 4-13: A sample content of an ARFF file.

Based on the results it is evident that the precision and recall have improved with the usage of Random Forest Classifier. Further improvements were made by cross-validation to minimize the effects of over-fitting of the classifier to the training set. It will generalize the classifier to an independent data set. Figure 4-16 illustrates the final confusion matrix after cross-validation.

```
=== Evaluation on training set ===

Time taken to test model on training data: 0.31 seconds

=== Summary ===

Correctly Classified Instances         352               86.0636 %
Incorrectly Classified Instances        57               13.9364 %
Total Number of Instances              409

=== Detailed Accuracy By Class ===

                 Precision  Recall   ROC Area  Class
                 0.981      0.486    0.976     CC
                 0.842      0.997    0.977     Non-CC
Weighted Avg.    0.879      0.861    0.977

=== Confusion Matrix ===

   a   b   <-- classified as
  53  56 |   a = CC
   1 299 |   b = Non-CC
```

Figure 4-14: Results of Naive Bayes Classifier (Weka output).

```
=== Evaluation on training set ===

Time taken to test model on training data: 0.22 seconds

=== Summary ===

Correctly Classified Instances         409              100      %
Incorrectly Classified Instances         0                0      %
Total Number of Instances              409

=== Detailed Accuracy By Class ===

                 Precision  Recall   ROC Area  Class
                 1.000      1.000    1.000     CC
                 1.000      1.000    1.000     Non-CC
Weighted Avg.    1.000      1.000    1.000

=== Confusion Matrix ===

   a   b   <-- classified as
 109   0 |   a = CC
   0 300 |   b = Non-CC
```

Figure 4-15: Results of Random Forest Classifier (Weka output).

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         405               99.022  %
Incorrectly Classified Instances         4                0.978  %
Total Number of Instances              409

=== Detailed Accuracy By Class ===

                 Precision  Recall   ROC Area  Class
                 0.991      0.972    0.999     CC
                 0.990      0.997    0.999     Non-CC
Weighted Avg.    0.990      0.990    0.999

=== Confusion Matrix ===

   a   b   <-- classified as
 106   3 |   a = CC
   1 299 |   b = Non-CC
```

Figure 4-16: Analysis results after 10 fold cross-validation (Weka output).

The overall performance varied when the classifiers are used upon different test sets. A similar methodology was utilized with the other pre-defined classes, and final versions of the classifiers were extracted to be employed in the platform for predictions.

4.6. Sensitivity Classification Process

The result of *Evidence Classifier* and *Content Classifier* identifies the followings:

1. Documents containing possible sensitive information related to Sri Lanka
2. Documents containing evidence of attacks
3. Documents containing indications of sensitive content or evidence of attacks, but not the actual content

4. Categorization of each document containing sensitive data with respect to the pre-defined classes

With this set of information, the synthesis process classifies each input document in to three sensitivity levels namely CRITICAL, HIGH, and LOW. The POC defines the sensitivity label based on two attributes:

1. Semantics
2. Magnitude

The semantic analysis identifies the severity of the data leak based on the types of data while magnitude or the frequency analysis calculates the severity of data based on the amount of data records available. Table 4-11 shows the sensitivity matrix used in the POC implementation of *LeakHawk*. (*m, n,* and *p* are configurable parameters).

Table 4-11: Sensitivity Matrix used in the POC.

	CRITICAL	**HIGH**	**LOW**
Credit Card related Frauds	Credit card numbers > n	Credit card numbers < n	Evidence found only for the presence of card data, no matching content
	PIN / CVV2 /CVC2 /Track Data available with Credit Card numbers		
	Expiration dates / cardholder names available with Credit Card numbers		
Credential dumps	Email / hash combinations > m	Email / hash combinations < m	
DB dump	Recoverable hashes	Non-recoverable hashes	
Email only list		Email addresses > p	Email addresses < p
Private keys	Presence of private keys		
Email conversation		Contains CONFIDENTIAL tags	Potential email conversation
Configuration file	Configuration files with passwords		
DNS related attack	Targeted attack	Just the domain names are mentioned within a list of diverse targets	
Defacement attack	Targeted attack	Just the domain names are mentioned within a list of diverse targets	

5. DISCUSSION AND ANALYSIS

This chapter discusses the effectiveness of the methodologies that were used in LeakHawk, in terms of achieving the desired functional and non-functional requirements. Section 5.1 discusses the unique methodology followed by *LeakHawk* together with a comparison with two major existing systems. Section 5.2 analyzes the each component of *LeakHawk* by validating the achieved results compared to the primary objectives. It also discusses the limitations of each module in terms of practical usage. Section 5.3 presents the evaluation results of the text classifiers utilized in *LeakHawk*.

5.1. Unique Methodology of LeakHawk

The methodology adopted by *LeakHawk* is unique compared to the publicly known existing applications developed to cater the same requirement. Table 5-1 compares LeakHawk with two existing systems, LeakedIn, and HIBP.

Table 5-1: Comparison between HIBP, LeakedIn, and LeakHawk.

	HIBP	**LeakedIn**	**LeakHawk**
Scope	PasteBin applications, File hosting sites and manual methodologies to analyze data leakages in general	Multiple PasteBin sites including pastebin	Initial implementation targets only pastebin and the platform supports all the Pastebin applications.
Text classification mechanisms used	Keyword based rules, manual integration of new dumps	regular expressions and keyword-based rules	Keyword based rules, regular expressions and machine learning techniques
Can be customized by the user	No	No	Yes
Scalability	Scalable in terms of integrating new data leakages which expand the breadth	Only new regular expressions can be included to expand the breadth	The platform itself is extensible in terms of depth and breadth.
Available methods to define the search domain	Email addresses (one at a time), domain names (one at a time), usernames (one at a time)	Only a feed is available. Cannot customize	Provide information templates to define the domain extensively
Supported sensitive data types	User credentials only	Email lists, credentials, configuration files, database dumps, hacking notifications	Credit card dumps, email lists, credentials, evidence of attacks, database dumps,

Sensitivity classification	none	none	Classify the data leaks based on the semantics and magnitude
Notifications	Individual emails	none	Notify the data owners via email with sensitivity label with analysis results

5.2. Analysis of Components of LeakHawk

LeakHawk follows a multi-layer approach in which multiple components are designed to minimize the number of false-negatives to an insignificant amount while reducing the number of false alarms. Apart from that, the batch processes executed on the input documents do not introduce a significant amount of delays in processing. However, certain documents indicated delays above average due to certain factors such as length and content types. In the practical sense, that delay will not affect the overall effectiveness of the containment process itself comprises of series of manual tasks, which are inherently time-consuming.

Text classifiers used in the system achieved better performance in terms of *precision* and *recall* with a certain exception under complex test sets.

5.2.1. Analysis of Pastebin Sensor

The implementation of connector and aggregator as the pastebin sensor module in the POC, assured the timeliness and comprehensiveness requirement as expected.

Test Case: *Submit 40 posts to pastebin within a period of 1 minute and verify whether LeakHawk can fetch all the posts.*

Result: *LeakHawk downloaded all the 40 posts altogether 58 (18 usual posts by others) posts pasted within a one-minute cycle.*

The assessment was repeated for10 times on 8 different days within a timespan of two weeks and resulted in zero false-negatives. Thus, we can conclude that the pastebin sensor functions as desired.

Timeliness preserves the evidence for a future forensic investigation since in certain cases; posts are deleted after publishing due to various reasons. The platform will have a copy of each post with a rare exception of the deleted posts within few seconds.

Limitations

Users will require subscribing for the PRO membership of pastebin to get access to the scrapping API (less than USD 50 as of June 2016). Pastebin identifies the PRO users based on the IP address. In a dynamic IP environment, users will have to whitelist their IPs manually via the pastebin portal at each IP change. Otherwise, the platform will not be able to access the service.

5.2.2. Analysis of PRE Filter

The *PRE filter* screens out the posts, which are non-sensitive in nature, such as video game chat sessions, pornographic content, and torrent information. It also eliminates non-textual posts such as binary files. As the average number of posts made in pastebin is less than 50, this filter was not useful in that scenario except for the exclusion of binary inputs. However, when the model is extended to support social media feeds, *PRE filter* will effectively improve the performance of the subsequent filters and classifiers by removing unrelated posts beforehand.

Limitations

Usage of this filter is set as optional as it may introduce false negatives into the platform as the input documents are inherently unstructured.

5.2.3. Analysis of Context Filter

The *Context filter* is a unique feature in *LeakHawk,* which allows the user to define the information domain, which is used by the *LeakHawk Core* as the context for monitoring pre-defined targets. Users or administrators can specify the information domain using information template. The information template is represented as a bag of words and regular expressions.

Table 5-2 illustrates the results of seeding 2,300 samples of textual documents across the *Context Filter*. The seed contains 220 manually labeled documents that are pre-validated as related to Sri Lanka. Ideally, the filter should identify 220 positive samples and 2,080 negative samples. The table lists the Document Frequency (DF) of each term (or positive matches) selected by the *Context Filter* (Column four). True Document Frequency (TDF), or True positives (column two) denotes the correct matches related to Sri Lanka, while False Document Frequency (FDF), or False Positives (column three) denotes the number of documents selected by the filter which is not relevant to Sri Lanka.

Figure 5-1 illustrates the document frequency distribution of each term and pattern irrespective of the accuracy). Figure 5-2 shows the distribution of false positives and true positives within the total selections of the *Context Filter*. As per the table and the diagrams, following key observations are made:

- Keywords such as "Lanka", "Sri Lanka" and "LK" are accountable for most of the results (irrespective of the accuracy)
    - "Lanka" = 188/220 = 85%
    - "Sri Lanka" = 152/220 = 68%
    - "LK" = 104/220 = 47%
- However, usage of "LK" introduces a considerable amount of false positives (see Figure 5-3). It is accountable for 24 false-positives, which are 38% from the total false-positives introduced by all the attributes.
- In a scenario where the case of the word "LK" is not-preserved, the number of false-positives increased to 36; which is a 50% increase.
- Pattern matching methods identify certain results, which are not captured by the above keywords but result in many false positives.
- Unique keywords such as names of conglomerates, favorite characters extract accurate results.

So it is evident that the use of multiple identifiers is necessary for the successful identification of positive instances with minimal false-negatives. Case sensitivity will significantly affect the accuracy and sensitivity of the results. Thus the performance of the *Context Filter* is extensively depending on the comprehensiveness of the values of the information template being used.

Table 5-2: Distribution of positive results from Context Filter.

Term / Pattern	TDF = True Positives	FDF = False Positives	DF = Total
lanka	180	8	188
sri lanka	149	3	152
srilanka	34	2	36
ceylon	3	2	5
LK (case-preserved)	80	24	104
colombo	3	1	4
Sinhala	1	1	2
sinhalese	1	0	1
buddhists	2	1	3
ID number format	1	1	2
IP address range	12	4	16
*.lk domains	38	4	42
lanka.* domains	4	0	4
Credit Card BIN ranges	22	12	34
popular characters	5	0	5
major conglomerates	3	0	3
other	2	0	2

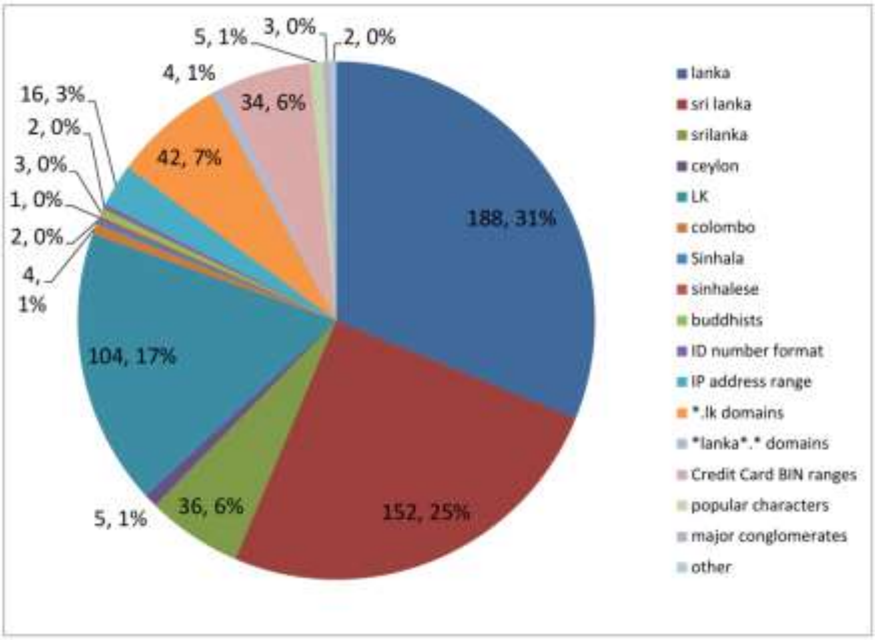


Figure 5-1: Distribution of total number of positive results.

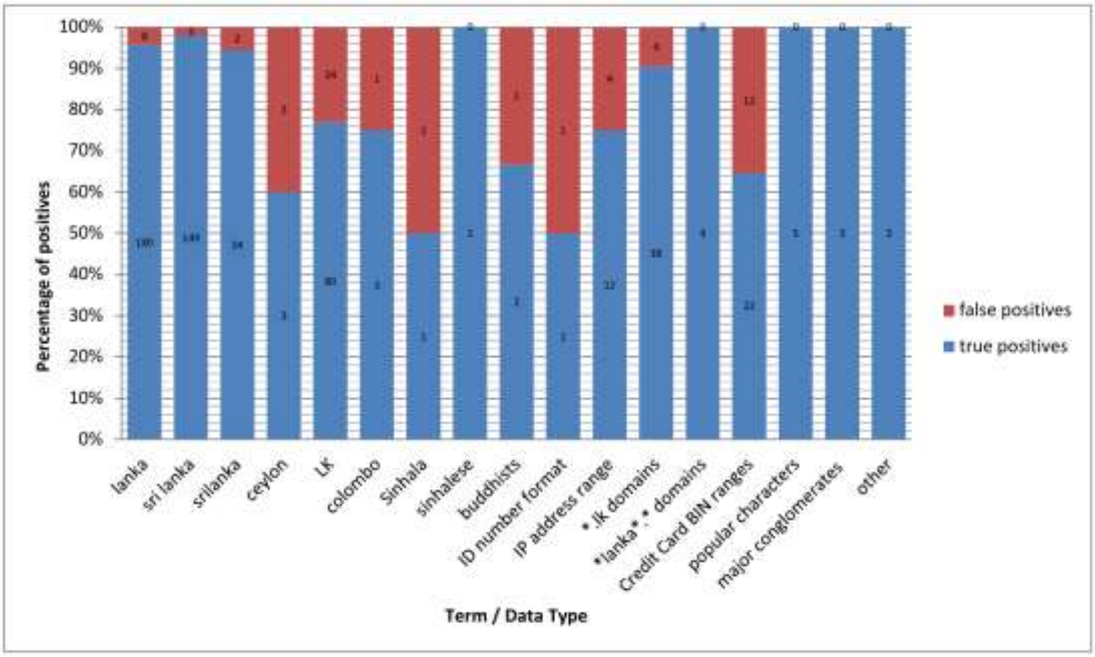


Figure 5-2: Distribution of true positives and false positives.

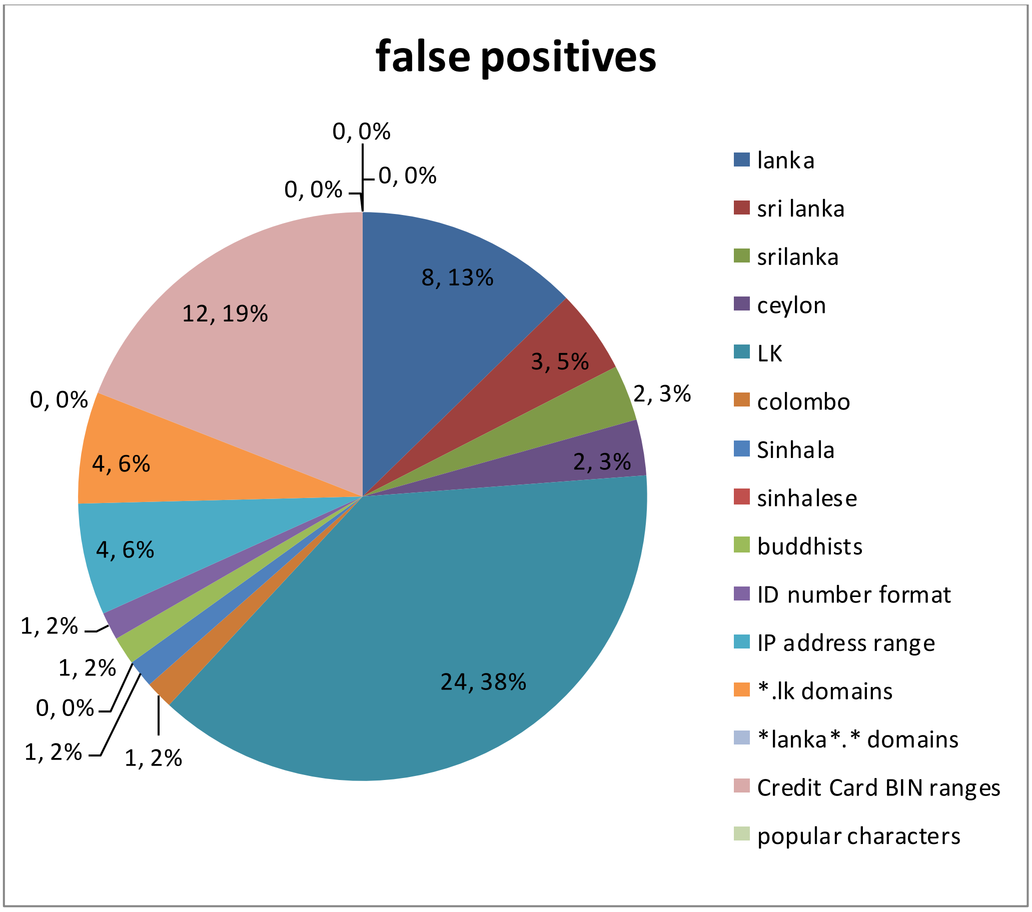


Figure 5-3: Distribution of false positives.

Limitations

Identifying all the keywords and regular expressions is a tedious task, which involves a considerable amount of manual effort. Formulation of information template for Sri Lanka, as explained in Section 4.4 provides general instructions to follow in defining the context.

5.3. Analysis of Document Classification Process

Once the relevant dataset is extracted, *Evidence Classifier* and *Content Classifier* extract and categorize the documents containing sensitive information or evidence of hacking attacks.

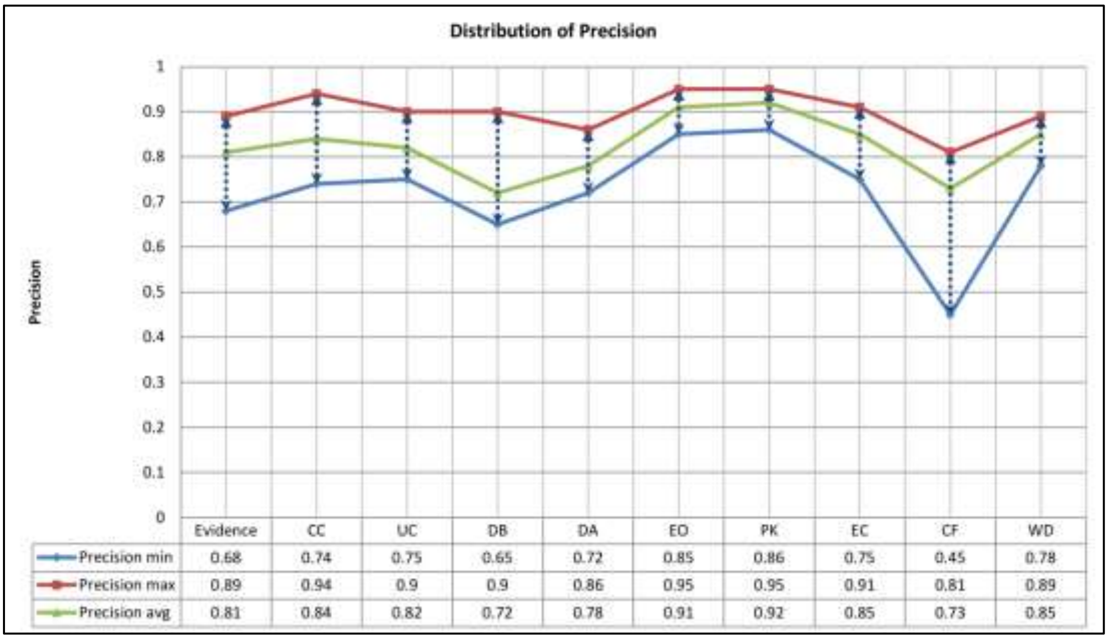
In the LeakHawk POC, altogether 10 classifiers were used in two different modules. As discussed in Section 4.5.2, the corpus was formulated by different means. Summary of the data points in the corpus as follows:

- No. of positive training samples for all the *Content Classifiers*– 1193 (see Table 4-5 for the distribution of input samples for each binary classifier under the Content Classification.
- No of positive training samples for the *Evidence Classifiers* – 940
- *Evidence Classifier* is fed with 10 different samples of test data with the number of entries per seed ranging from 100 to 1,000.
- Each *Content Classifier* is fed with 20 different sample test sets with the number of entries per seed ranging from 30 to 850.

Each classifier performed with diverse results upon different data sets. Figure 5-4 illustrates the distribution of maximum, minimum and average precision values taken by each Classifier upon multiple cross-validation datasets.

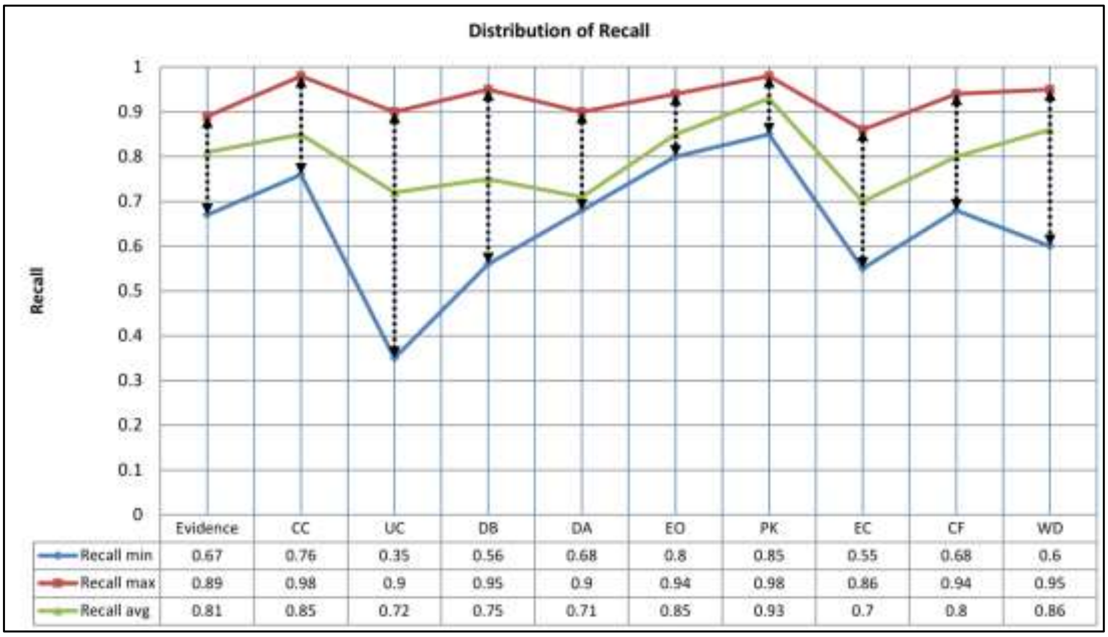
As per the graph, the classifiers for identifying Email Only (EO) data leaks, Private Key breaches (PK) and evidence of Website Defacement (WD) attacks indicate better performance in terms of the range (the difference between maximum and minimum values of precision. It infers that when those classifiers categorize a set of inputs as such, the number of false-positives in the selection is minimal.

Figure 5-5 illustrates the distribution of recall values taken by each classifier upon multiple cross-validation datasets. As per the graph, Credit Card (CC), EO, Private Key (PK) and WD classifiers perform better than 85% in terms of recall. PK classifier indicates the best average and range in terms of recall. It suggests that when the classifier predicts a set of inputs as Private Keys, that positive dataset will contain the majority of the Private Keys in the dataset with significant sensitivity. Alternatively, the classifiers for the User Credentials (UC) and E-mail conversations have a wider range and lower average recall. Further analysis suggests that the majority of false negatives associated with the UC are the dumps with passwords (not containing attributes that can be extracted with patterns such as e-mails and hashes). Heuristics defined for the UC are not dominant enough to identify particular password dumps.



	Evidence	CC	UC	DB	DA	EO	PK	EC	CF	WD
Precision min	0.68	0.74	0.75	0.65	0.72	0.85	0.86	0.75	0.45	0.78
Precision max	0.89	0.94	0.9	0.9	0.86	0.95	0.95	0.91	0.81	0.89
Precision avg	0.81	0.84	0.82	0.72	0.78	0.91	0.92	0.85	0.73	0.85

Figure 5-4: Distribution of Precision.



	Evidence	CC	UC	DB	DA	EO	PK	EC	CF	WD
Recall min	0.67	0.76	0.35	0.56	0.68	0.8	0.85	0.55	0.68	0.6
Recall max	0.89	0.98	0.9	0.95	0.9	0.94	0.98	0.86	0.94	0.95
Recall avg	0.81	0.85	0.72	0.75	0.71	0.85	0.93	0.7	0.8	0.86

Figure 5-5: Distribution of Recall.

6. CONCLUSION AND FUTURE WORK

6.1. Summary

LeakHawk is developed to fill the void of the lack of an effective and scalable early detection platform to monitor sensitive data leakages and evidence of hacking attacks. Primarily it targets online channels such as PasteBin sites and social media feeds where most of the data breaches are originated.

The key contribution of this research is the design and development of a unique methodology for the identification and classification of data leakages by utilizing rule-based evaluations and machine-learning techniques. The proposed methodology significantly reduces the number of false-negatives while minimizing the false positives, which improves the usability of the platform in terms of reducing the management overhead. Furthermore, the proposed methodology reduces the number of manual verification procedures by providing granular level analysis results along with a sensitivity label, which can be used as a benchmark to define the associated risk of each incident and invoke the correct containment procedures. Moreover, the implementation of the methodology is scalable where it can be extended to monitor multiple content sources, a large volume of content, variety of contents and entities.

A working model of the proposed platform was implemented as a proof of concept (LeakHawk 1.0). The POC monitors www.pastebin.com, the mostly used Pastebin application, for sensitive information leakages and evidence of hacking attacks related to Sri Lanka. *LeakHawk 1.0* incorporated a set of ten machine-learning based text classifiers for the severity classification with *precision* varying between 45%-95% with an average of 82% and *recall* ranging between 35%-98% with an average of 80%.

The development process of *LeakHawk* focused on five major aspects namely, breadth, depth, timeliness, consistency, and accuracy. *LeakHawk* is more focused on the depth of the problem rather than the breath. It considers all the feeds that come from a given source as textual content and utilize text-engineering methodologies to drill further down the meaning of the content. It will assist the administrators to understand the severity of each security incident, which means the depth of the problem of identifying security information leakages on the Internet. To cover the breath, it is required to add all the paste sites; social media feeds via connectors and aggregators. The sensor module and the classifiers assure the timeliness and consistency aspects where all the posts published at a particular target are fetched and classified without a significant delay.

In terms of practical usage, *LeakHawk* involves a set of manual procedures in defining the information model containing the unique attributes of a particular entity. The comprehensiveness of the information model will significantly affect the coverage of the platform in verifying whether that entity is involved in a data leakage incident. In support of that, the POC implementation provided certain guidelines that can be used for different domains.

6.2. Future Work

Functional and performance aspects of *LeakHawk* when dealing with social media feeds are not evaluated in this research. In order to cater a large amount of input feeds retrieved from such sources, *PRE filter* module should be enhanced to screen out unrelated feeds in an efficient manner. In a possible future expansion, features of Complex Event Processing (CEP) [58] can be integrated into the platform by enhancing the *PRE filter* to cater large volume of input feeds. In most cases, data leakage incidents exposed via social media feeds do not contain the actual dumps of the leak, rather external links along with evidence of the incident. Therefore, the *Evidence Classifier* will be the primary text analytic component to be utilized for such applications.

The POC implementation caters only a limited set of sensitive data classes, which are common to pastebin. In general, the sensitive data types are exhaustive and require further class definitions to enhance the classification process. Furthermore, the sensitivity classification methodology can be generalized to cater all the sensitive data types by incorporating ontology engineering principles [59].

To generalize *LeakHawk* for other PasteBin applications, other than www.pastebin.com, several enhancements are necessary for the respective connectors and aggregators. These changes are mostly needed due to the fact that

Paste sites differ in the availability of an API, search functions, access limitations, etc.

A comprehensive performance analysis needs to be conducted in terms of overall performance and extensibility. Once the *Content Filter* is enriched to cover a more breadth of sensitive data types, it will be required to conduct a thorough analysis, in terms of the precision and recall of classifiers. Furthermore, a dashboard can be integrated into the platform to enhance the management and usability along with multiple alerting mechanisms.

Overall performance of the Content Classifier has been reduced due to the poor performance of some of the sub-classifiers. For example, the minimum overall Recall value of the Content Classifier is 35% as the maximum Recall value achieved by the UC Classifier is 35%. Therefore, it is required to enhance the performance of the under-performing sub-classifier to improve the overall performance. Overall performance of the Context Classifier can be improved by integrating WordNet [60]. WordNet concepts for text analytics can significantly reduce the manual overhead in defining the information templates.

The performance of the *LeakHawk* can be significantly improved by integrating canary traps [61]. A set of unique keywords or identifiers can be embedded to the sensitive information possessed by an organization, and those keywords can be used as seeds by the *LeakHawk* in the monitoring phase.

REFERENCE

[1] "Pastebin.com - #1 paste tool since 2002!" [Online]. Available: http://pastebin.com/. [Accessed: 08-Jun-2016].

[2] "AnonymousSriLanka's Pastebin - Pastebin.com." [Online]. Available: http://pastebin.com/u/AnonymousSriLanka. [Accessed: 16-Jun-2016].

[3] "Davyjones's Pastebin - Pastebin.com." [Online]. Available: http://pastebin.com/u/davyjones. [Accessed: 16-Jun-2016].

[4] "National/governmental CERTs - Baseline Capabilities — ENISA." [Online]. Available: https://www.enisa.europa.eu/topics/national-csirt-network/csirt-capabilities/baseline-capabilities. [Accessed: 15-Jun-2016].

[5] "Data Breaches." [Online]. Available: http://www.privacy.wv.gov/tips/Pages/DataBreaches.aspx. [Accessed: 18-Jun-2016].

[6] "How Harmful Can a Data Breach Be? - InfoSec Resources." [Online]. Available: http://resources.infosecinstitute.com/the-cost-of-a-data-breach-how-harmful-can-a-data-breach-be/. [Accessed: 12-Jun-2016].

[7] "Commercial Bank of Ceylon Hacked? - BankInfoSecurity." [Online]. Available: http://www.bankinfosecurity.com/commercial-bank-ceylon-apparently-hacked-a-9103. [Accessed: 15-Jun-2016].

[8] "Hacking, Data Breaches & Cyber Warfare | IT Consulting & Technology Services Distributor." [Online]. Available: http://gcgcom.com/hacking-data-breaches-cyber-warefare/. [Accessed: 27-Jun-2016].

[9] "Pastebag - Text Sharing Applications." [Online]. Available: http://app.urlbag.com/1fxr8Bs0. [Accessed: 12-Jun-2016].

[10] "Pastebin - Wikipedia, the free encyclopedia." [Online]. Available: https://en.wikipedia.org/wiki/Pastebin#cite_note-6. [Accessed: 18-Jun-2016].

[11] "Guess what: Code sharing sites being used to share emails and passwords ALL year round." [Online]. Available: http://thenextweb.com/2009/10/05/guess-code-sharing-web-services-share-emails-passwords-year/#gref. [Accessed: 18-Jun-2016].

[12] S. Matic, A. Fattori, D. Bruschi, and L. Cavallaro, "Peering into the Muddy Waters of Pastebin," in Proc. *A Global treaty on cybersecurity and cybercrime: a contribution for peace, justice and security in cyberspace,* Oslo, 2011, pp. 16–17.

[13] "Troy Hunt: Introducing paste searches and monitoring for 'Have I been pwned?'" [Online]. Available: https://www.troyhunt.com/introducing-paste-searches-and/. [Accessed: 12-Jun-2016].

[14] "Trending Pastes at Pastebin.com." [Online]. Available: http://pastebin.com/trends. [Accessed: 12-Jun-2016].

[15] "Pastes Archive - Pastebin.com." [Online]. Available: http://pastebin.com/archive. [Accessed: 12-Jun-2016].

[16] "Pastebin.com - FAQ [Frequently Asked Questions]." [Online]. Available: http://pastebin.com/faq. [Accessed: 29-Jun-2016].

[17] M. R. O. 17 and 2014 Internet, "Facebook is taking a proactive approach to fighting password leaks," *TechRadar*. [Online]. Available: http://www.techradar.com/news/internet/facebook-is-taking-a-proactive-approach-to-fighting-password-leaks-1269726. [Accessed: 12-Jun-2016].

[18] "700,000 Dropbox credentials hacked, hacker leaks 'Dropbox Hacks Teasers' on Pastebin." [Online]. Available: http://www.techworm.net/2014/10/700000-dropbox-credentials-hacked-hacker-leaks-dropbox-hacks-teasers-pastebin.html. [Accessed: 12-Jun-2016].

[19] "Have I been pwned? Check if your email has been compromised in a data breach." [Online]. Available: https://haveibeenpwned.com/. [Accessed: 12-Jun-2016].

[20] "Dump Monitor (@dumpmon) | Twitter." [Online]. Available: https://twitter.com/dumpmon?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor. [Accessed: 12-Jun-2016].

[21] "matthewdfuller/pastebin-find: Python script to monitor new Pastebin pastes for a provided search term." [Online]. Available: https://github.com/matthewdfuller/pastebin-find. [Accessed: 27-Jun-2016].

[22] "Google Alerts - Monitor the Web for interesting new content." [Online]. Available: https://www.google.com/alerts. [Accessed: 17-Jun-2016].

[23] "xme/pastemon: pastebin.com Content Monitoring Tool." [Online]. Available: https://github.com/xme/pastemon. [Accessed: 17-Jun-2016].

[24] "LeakedIn." [Online]. Available: http://www.leakedin.com/. [Accessed: 17-Jun-2016].

[25] "RaiderSec: Introducing dumpmon: A Twitter-bot that Monitors Paste-Sites for Account/Database Dumps and Other Interesting Content." [Online]. Available: http://raidersec.blogspot.com/2013/03/introducing-dumpmon-twitter-bot-that.html. [Accessed: 17-Jun-2016].

[26] "Text Analytics Software | Text Analytics & Text Mining Tools, Software." [Online]. Available: http://www.clarabridge.com/text-analytics/. [Accessed: 18-Jun-2016].

[27] "Structured, semi structured and unstructured data | Jeremy Ronk." [Online]. Available: https://jeremyronk.wordpress.com/2014/09/01/structured-semi-structured-and-unstructured-data/. [Accessed: 18-Jun-2016].

[28] C. Goller, J. Löning, T. Will, and W. Wolff, "Automatic Document Classification-A thorough Evaluation of various Methods.," *ISI*, vol. 2000, pp. 145–162, 2000.

[29] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaala, "A survey of web information extraction systems," *Knowl. Data Eng. IEEE Trans. On*, vol. 18, no. 10, pp. 1411–1428, 2006.

[30] "Sensitivity of Data | Information Systems & Technology." [Online]. Available: https://ist.mit.edu/security/data_sensitivity. [Accessed: 18-Jun-2016].

[31] S. Seneviratne, A. Seneviratne, M. A. Kaafar, A. Mahanti, and P. Mohapatra, "Early Detection of Spam Mobile Apps," in Proc. *24th International Conference on World Wide Web*, 2015, pp. 949–959.

[32] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," in *Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 83–92.

[33] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, "Why people hate your app: Making sense of user feedback in a mobile app store," in Proc. *19th ACM SIGKDD International Conference on Knowledge discovery and data mining*, 2013, pp. 1276–1284.

[34] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti, "Your installed apps reveal your gender and more!," 2014, pp. 1–6.

[35] "1.12. Multiclass and multilabel algorithms — scikit-learn 0.17.1 documentation." [Online]. Available: http://scikit-learn.org/stable/modules/multiclass.html. [Accessed: 18-Jun-2016].

[36] "6. Learning to Classify Text." [Online]. Available: http://www.nltk.org/book/ch06.html. [Accessed: 18-Jun-2016].

[37] gandalf.psych.umn.edu, "Feature Selection/Extraction."

[38] "Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining." [Online]. Available: http://www.tfidf.com/. [Accessed: 29-Jun-2016].

[39] ISACA, "Data Leak Prevention - ISACA." 2010.

[40] M. Hart, P. Manadhata, and R. Johnson, "Text classification for data loss prevention," in Proc. *Privacy Enhancing Technologies*, 2011, pp. 18–37.

[41] E. Pshehotskaya, S. Ryabov, and T. Sokolova, "New Approaches to Data Classification in DLP Systems," in Proc. International Conference on Computing Technology and Information Management, S. l., 2014.

[42] "Data Classification." [Online]. Available: http://www.infosectoday.com/Articles/Data_Classification/Data_Classification.htm. [Accessed: 25-Jun-2016].

[43] SANS institute, "information-classification-who-846.pdf." Feb-2003.

[44] "Guidelines for Data Classification-Computing Services ISO - Carnegie Mellon University." [Online]. Available: http://www.cmu.edu/iso/governance/guidelines/data-classification.html. [Accessed: 25-Jun-2016].

[45] "Data Classification Guide | Privacy and Information Security." [Online]. Available: https://security.illinois.edu/content/data-classification-guide. [Accessed: 25-Jun-2016].

[46] "Information Security - Information Classification Standard." [Online]. Available: https://daf.csulb.edu/offices/vp/information_security/information_classification_standard.html. [Accessed: 25-Jun-2016].

[47] "Pastebin.com - Scraping API." [Online]. Available: http://pastebin.com/api_scraping_faq. [Accessed: 24-Jun-2016].

[48] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan, “Extracting information about security vulnerabilities from web text,” in Proc. *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, 2011, vol. 3, pp. 257–260.

[49] “Thomson Reuters | Open Calais API.” [Online]. Available: http://www.opencalais.com/opencalais-api/. [Accessed: 21-Jun-2016].

[50] “Language Detection API.” [Online]. Available: https://detectlanguage.com/. [Accessed: 23-Jun-2016].

[51] “WHOIS Search, Domain Name, Website, and IP Tools - Who.is.” [Online]. Available: https://who.is/. [Accessed: 23-Jun-2016].

[52] “Pastebin Pastes Collection : Free Web : Download & Streaming : Internet Archive.” [Online]. Available: https://archive.org/details/pastebinpastes. [Accessed: 21-Jun-2016].

[53] “Pastebin dump collection.” [Online]. Available: http://psbdmp.com/. [Accessed: 22-Jun-2016].

[54] “Natural Language Toolkit — NLTK 3.0 documentation.” [Online]. Available: http://www.nltk.org/. [Accessed: 25-Jun-2016].

[55] PCI Security Standards Council, “PCI_DSS_v3-2.pdf.” Apr-2016.

[56] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[57] “weka - ARFF (stable version).” [Online]. Available: http://weka.wikispaces.com/ARFF+%28stable+version%29. [Accessed: 29-Jun-2016].

[58] “Apache Flink: Introducing Complex Event Processing (CEP) with Apache Flink.” [Online]. Available: https://flink.apache.org/news/2016/04/06/cep-monitoring.html. [Accessed: 30-Jun-2016].

[59] Q. He, L. Qiu, G. Zhao, and S. Wang, “Text categorization based on domain ontology,” in Proc. *International Conference on Web Information Systems Engineering*, 2004, pp. 319–324.

[60] “About WordNet - WordNet - About WordNet.” [Online]. Available: https://wordnet.princeton.edu/. [Accessed: 21-Dec-2016].

[61] “Canary Trap Explained - Simplicable.” [Online]. Available: http://arch.simplicable.com/arch/new/what-is-a-canary-trap. [Accessed: 28-Jun-2016].

7. APPENDIX A: DATA LEAKS RELATED TO SRI LANKA

Date	Title	Targeted Entity	Type of Attack	Posted By	URL
2011-Jul	NIBM Sri Lanka db leaked!	NIBM	DB Dump	A GUEST	http://pastebin.com/WFRSCjw9
2011-Aug	Sri Lanka's Military - Airforce.LK DNS Fuck3D Leaked	Sri Lanka Airforce	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/Tne79Zf3
2011-Aug	Sri Lanka's Government CERT DNS Fuck3D Leaked	SLCERT	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/7qERjgYM
2011-Aug	Sri Lanka's Military - Police.LK DNS Fuck3D Leaked	Sri Lanka Police	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/ziPhuaYn
2011-Aug	Sri Lanka Bell DNS Fuck3D Leaked (LANKABELL.NET)	LankaBell	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/qAgSbXLD
2011-Aug	HUTCHISON TELECOM (GSM) Sri Lanka's DNS Fuck3D Leaked	HUTCHISON TELECOM	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/rGLfXBuT
2011-Aug	Sri Lanka's Largest Community Forum's (ELAKIRI.COM) DNS Fuck	ELAKIRI.COM	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/nN7kQLS6
2011-Aug	LANKACOM Sri Lanka's Another ISP (Internet Exchange) DNSi	LANKACOM	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/gt7XWe95
2011-Aug	Untitled	SRI LANKAN ARMY	DB Dump	W3BD3F4C3R	http://pastebin.com/wMjTzwvh
2011 Aug	SUNTEL/WOW Sri Lanka's 2nd Largest Telco/ISP Provider's DNSi	SUNTEL/WOW	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/mTsasMzH
2011-Aug	Sri Lanka's Largest and National Internet Data Center DNSi	Sri Lanka Telecom	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/52khFWrM
2011-Aug	Sri Lanka's Military - Defence.LK DNS Fuck3D Leaked	Defence.LK	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/eHVYSeJX
2011-Aug	Sri Lanka's National Domain Registry's (NIC.LK) DNS Fuck3D	NIC.LK	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/RQaS6wVh

2011-Aug	SLT.LK SEA-ME-WE Border Gateway Router Rooted	Sri Lanka Telecom	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/kXSxcUQh
2011-Aug	ETISALAT SRI LANKA Telecom Provider's and ISP DNS Fuck3D	ETISALAT	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/wpxQijc7
2011-Aug	Sri Lanka's Military - Navy.LK DNS Fuck3D Leaked	Sri Lanka Navy	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/r3PaXjT1
2011-Aug	University of Colombo Sri Lanka - DNS Fucked and Leaked	University of Colombo	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/zxrka602
2011 Aug	University Colombo - Sri Lanka SSH/RSA Login Key Dump	University of Colombo	Private Key Compromise	ANONYMOUSSRILANKA	http://pastebin.com/8U0vqmLs
2011-Aug	Sri Lanka's Largest Private IT/Management University DNSi	SLIIT	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/pDd2NcBh
2011-Aug	DIALOGSL.COM - Sri Lanka's Largest Mobile GSM Provider's DNSi	Dialog Telecom	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/76SK3NgT
2011-Aug	Sri Lanka's National Mobile/GSM Provider and ISP's DNS Fuck	MOBITEL	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/pWKarbwT
2011-Aug	Sri Lanka's Largest Mobile Provider's DNS Fuck3D Leaked	Dialog Telecom	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/SmJaERVn
2011-Aug	Sri Lanka's National University Network DNS Fuck3D Leaked	LEARN	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/8rQXjrja
2011-Aug	University of Moratuwa Sri Lanka - DNS Fucked and Leaked	University of Moratuwa	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/UdbMKQuh
2011-Aug	PARLIAMENT of Sri Lanka's (Parliament.LK) DNS Fuck3D Leaked	SL PARLIAMENT	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/MA48zvw4
2011-Aug	Sri Lanka's Largest and National Telecom Provider's DNS Fuck	Sri Lanka Telecom	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/qLE76Kpn
2011 Aug	Sri Lanka's Military - NAVY.LK EMAIL/WEB Server Exposed/Fuck	Sri Lanka Navy	Enumeration Attempt	ANONYMOUSSRILANKA	http://pastebin.com/ddGZdkM7
2011 Aug	Sri Lanka's Military - Navy.LK OS/WEB Server Exposed/Fuck3D	Sri Lanka Navy	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/qbNT1pQf

2011 Aug	LONDON STOCK EXCHANGE DEVELOPER - MILLENNIUMIT.COM - DNSi	MIT Sri Lanka	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/vVEuak0a
2011 Sept	UGC.AC.LK - University Grants Commission of Sri Lanka	UGC	DNS Related Attack	ANONYMOUSSRILANKA	http://pastebin.com/h9KBE2xn
2011-Sep	Warning to Anonymous SriLanka	-	Hacking notification	SLCYBERARMY	http://pastebin.com/Qq5HFNR2
2011-Sep	Untitled	SUNDAY TIMES OF SRI LANKA	DB Dump	W3BD3F4C3R	http://pastebin.com/aTFWNgWj
2011-Sep	Untitled	LANGUAGE COMMISSION	DB Dump	W3BD3F4C3R	http://pastebin.com/CHLs5cUj
2011-Sep	Untitled	digitalhouse.lk	DB Dump	W3BD3F4C3R	http://pastebin.com/RUsrCBWB
2012-Apr	Untitled	srilancars.com	Credential Dump	A GUEST	http://pastebin.com/5FJi2nuB
2013-Jan	Sri Lanka Lbo.lk hacked	Lbo.lk	Credential Dump	DAVYJONES	http://pastebin.com/quBfpAYu
2013-Jan	Sri Lanka Foreign employment ministry website hacked and here	Foreign employment ministry	DB Dump	DAVYJONES	http://pastebin.com/V9ddGkrD
2013 Jan	board of investment Sri Lanka hacked	board of investment	DB Dump	DAVYJONES	http://pastebin.com/p0RfJJPH
2013 Feb	Sri Lanka high commission Maldives hacked	Sri Lanka high commission of Maldives	DB Dump	DAVYJONES	http://pastebin.com/QMMG9Bu2
2013 Feb	Sri Lanka sports minister website dumped	sports minister's personal website	DB Dump	DAVYJONES	http://pastebin.com/RiFfxv8U
2013 Feb	UNHCR.LK	UNHCR.LK	DB Dump	DAVYJONES	http://pastebin.com/5UaZ6XgB
2013 Feb	Sri Lankan President Mahinda Rajapaksha theatre hacked	Mahinda rajapaksha theatre	DB Dump	DAVYJONES	http://pastebin.com/zUZ29Lnd
2013 Feb	spiceisland.lk hacked by BlacKhatAnon- MySQL	spiceisland.lk	DB Dump	BLACKHATANON	http://pastebin.com/RHvhNdQ7

2013-Feb	www.nanasala.lk hacked by BLacKhatAnon (SL BL@cKhat$)	nanasala.lk	DB Dump	BLACKHATANON	http://pastebin.com/nLEUs9Aj
2013-Aug	Sri Lanka School And U.S Sites Hacked By KamiSecTeam	mvmv.sch.lk	Website Defacement	ANUARLINUX	http://pastebin.com/1drJpvd8
2014-Apr	#OpSrilanka - Hacked by ACA	Multiple targets	Website Defacement	A GUEST	http://pastebin.com/KNPP4G4z
2014-Jun	Anonymous Sri Lanka Reborn @ 2014 Against Muslim Extremists	-	Hacking notification	ANONYMOUSSRILANKA	http://pastebin.com/uPrRmUeH
2014-Jun	Sunil Motors Corporation - Breached #OpSriLanka	Sunil Motors Corporation	Credential Dump	UGLEGION	http://pastebin.com/8rSbjirC
2014-Jun	Untitled(More Sri Lanka Government leak because they dont stop extremist attacking Muslim in Sri Lanka!)	Public Administration and Management	DB Dump	A GUEST	http://pastebin.com/EUcB4C6M
	AnonGhost on #OpSrilanka	Multiple targets	Website Defacement	HUSSEIN98D	http://pastebin.com/JKbLtFhF
2014 -Oct	HaxorsteinBD 443address,773 air route leaked	Multiple targets	DB Dump	HAXORSTEINBD	http://pastebin.com/fSc6bTty
2016-Apr	Panama leaks:- Datasheet 2	-	Information Leak	A GUEST	http://pastebin.com/WHr3CZgz
2016-Jun	Untitled	military	Information Leak	A GUEST	http://pastebin.com/4YBcUDm2