**DESIGNING AN EMAIL HEADER ANALYSIS TOOL FOR CYBER THREAT INTELLIGENCE**

AbdulAzeez Hassan T1, Suleiman Ahmad2, Raji AbdulGafar3, Maliki Danlami4

1 Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria

[abdulazeezhassant@futminna.edu.ng](mailto:abdulazeezhassant@futminna.edu.ng)

2 Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria

[ahmads@futminna.edu.ng](mailto:ahmads@futminna.edu.ng)

3 Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria

[a.raji@futminna.edu.dg](mailto:a.raji@futminna.edu.dg)

4 Department of Computer Engineering, Federal University of Technology, Minna, Nigeria

danlami.maliki@futminna.edu.ng

# **ABSTRACT**

This project proposes a design for an analysis of email headers as a way of obtaining cyber threat intelligence. The aim is to enable organizations to detect and prevent cyber-attacks more effectively by analysing the email headers of incoming messages. The analysis is conducted using the information found in the e-mail header. The proposed design incorporates an automated system that analyses the email headers and provides insights into the source, hop IP information, and security header details. Overall, the proposed email header analysis system has the potential to improve the cyber security posture of organizations and enhance their ability to respond to cyber threats.

**Keywords: -** *Email Forensics, Email Traffic Analysis, Open source forensics, digital forensics readiness,*

*forensics process.*

**INTRODUCTION**

Email, an abbreviated form of electronic mail, serves as a prevalent means of message exchange on the Internet. It stands as one of the internet's most commonly utilized applications, steadily growing in global usage and evolving into a crucial communication tool across commercial, academic, and personal domains, offering a straightforward and efficient mode of communication. Due to the popularity and ease of use of email, cybercriminals tend to exploit its use by sending spam in bulk, phishing messages, distributing child pornography, propagating malware, sending threatening/hate mail, and engaging in other criminal activities (Haggerty, et al., 2011).

Email, a widely used means of communication, has become a prime target for cyber threats, necessitating thorough analysis as part of threat detection efforts. The surge in electronic communication adoption by both individuals and organizations has led to a significant increase in the volume of emails processed and stored. When emails are utilized for illicit activities or become the focus of cyber attacks, they serve as crucial evidence in digital investigations. The effectiveness of such investigations relies heavily on the quality and timeliness of the digital evidence gathered (John, et al., 2020).

Each email transmission generates a wealth of additional data pertinent to that specific message, forming a digital trail that can be followed during investigative analysis. E-mail threat intelligence has been developed to scrutinize email content and origins, identifying key elements such as sender, recipient, timestamps, and message origin by deconstructing emails into their constituent parts: the email header and body (Chhabra, 2015).

This evidence obtained from digital sources can assist in the identification or tracking of individuals and devices engaged in email misuse. Analysing email data can be a time-consuming and complex process, particularly when dealing with large volumes of messages. Hence, there's a necessity to create and refine tools that streamline the collection and analysis of emails, facilitating the discovery of digital evidence.

## Description of the New System

The web-based email threat-hunting software offers a straightforward interface for uploading emails, analyzing the information within the email header, and exploring additional data gathered from the internet based on header parameters. Using this software, tasks that would typically take hunters hours to complete can now be accomplished within minutes, significantly enhancing threat-hunting efficiency.

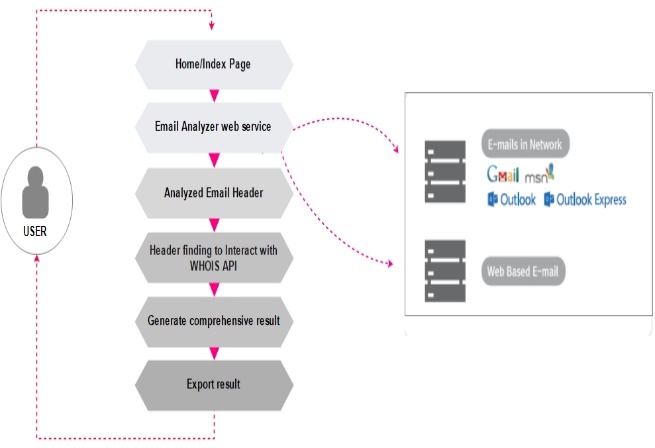
As depicted in Figure 3, users will engage with the proposed system through a web-based interface to submit original email messages for analysis and review the reports generated from both internet sources and internal threat hunter investigation networks.

Figure 3: Diagram depicting the working procedure of the proposed software

The proposed solution comprises a graphical user interface accessible via the web, an API for IP/WHOIS functionality, and an output of exported results.

## Related Works

In a study conducted by (Kumar1 et al. 2018) an online tool was introduced to analyse malicious activities within email traffic, highlighting the necessity for a structured email threat-hunting system that integrates the unique attributes of email as digital evidence into standard threat intelligence workflows. Scientific evaluation is essential to assess the credibility of emails, given their increasingly crucial role in analysis and evidential provision. The authors propose a potential solution to this issue, suggesting the development of a system capable of monitoring email traffic, processing, and generating reconstructed email messages ready for forensic investigation to detect instances of malicious activity and scrutinize email content.

Their proposed methodology centers on an open-source client-server application tasked with acquiring data from a network server, analyzing collected data, identifying malicious activities, and generating evidence reports upon detecting anomalies in email header traffic. However, due to the requirement of a specialized dedicated network server, which can be costly, the authors advocate extending the development to a web platform. This model effectively monitors email traffic and produces reconstructed email messages suitable for forensic investigation.

One common malicious attack discussed is email address spoofing, wherein the source address is manipulated to appear as if it originates from the intended recipient, while the actual sender is the attacker.

(Upadhyay & Kumar, 2018) proposed an algorithm designed to validate the content of email headers, focusing specifically on four key fields: R-SPF, DKIM, DKIM Signature, and DMARC. Named the "sender spoofing analysis algorithm," it aims to identify instances of sender spoofing in emails by conducting thorough evaluations of mail header fields, thus enabling the detection of spoofed emails. The algorithm involves creating and implementing data values for both legitimate and spoofed emails to assess the validity of email header fields.

The algorithm checks for four predefined conditions to detect email spoofing: the Message-ID entry should match the domain address of the FROM field entry, both DMARC and R-SPF must be marked as PASS, DKIM and R-SPF values must both contain PASS, with R-SPF containing the same domain name as in the From Field, and if DKIM and R-SPF are absent, the D's value must match the domain name of the FROM field under DKIM-signature. If these conditions are not met, the sent email is identified as fake, indicating sender spoofing.

To detect sender address spoofing across all email domains, the methodology outlined in this algorithm involves creating original email header datasets using Java API, then generating datasets of spoofed email headers using the same API, followed by email recognition and classification using Java API. To enhance the optimization of this algorithm, more sophisticated tools are recommended for complex header analysis, including the identification of machine IP addresses and the retrieval of corresponding log files.

In their research (Andrew et al. 2020), a comparative examination was carried out on cyber-threat intelligence sources, formats, and languages, encompassing both internal and external sources. The objective was to investigate how different formats and languages align with the data requirements of typical security operations. The findings revealed that the effectiveness of these sources, formats, and languages varies depending on the particular use case and needs. The study suggested incorporating origin and freshness/timestamp data into feeds and maintaining comprehensive threat data to mitigate challenges associated with the quality and dissemination of cyber-threat intelligence.

(Stadlinger et al. 2018), introduced a predictive visualization forensic email analysis tool, which employs responsive and interactive graph visualization of email data supplemented by statistical insights. Given the substantial volume of emails typically found in inboxes, the challenge of manually sifting through them impedes the rapid identification of relevant communication partners and the structures surrounding them, hindering accurate response and assessment. This obstacle significantly impacts forensic investigations, prompting the need for tools that facilitate swift comprehension of a suspect's email communication, identification of specific contacts, and recognition of communication patterns across single or multiple email accounts.

To address the task of identifying communication patterns in unidentified mailboxes, the software integrates various existing visualization methods, offering a dynamic and sensitive approach to analyzing extensive email data from multiple mailbox files. Implemented in Python, the tool utilizes JavaScript to deliver an interactive user interface via a local web server, comprising four distinct modules. The Initialization Module manages input data transfer to respective parsing modules, which extract and normalize data from diverse formats before storing it in a database. Subsequently, the graph generation module processes the stored data to generate a visual representation of communications between parties. Finally, leveraging the Flask micro-framework, the tool offers a web-based user interface using HTML5 and JavaScript, ensuring flexibility and responsiveness.

The tool presents results through forced-directed graphs and responsive charts, offering an innovative and dynamic visualization approach. However, there is a need for improvement in parsing speed performance through optimization of the input parser.

(Msongaleli et al 2018), proposed an email investigation algorithm tailored for criminal investigations and dispute resolutions. This three-tiered forensic email algorithm begins with evidence gathering, followed by email header analysis, email server log analysis, and image analysis of local computers. While email header analysis is commonly relied upon by investigators to determine the sender or recipient identity, it may provide limited information about the email's source. In cases where the origin of the contested email cannot be established from header analysis alone, it is advisable to review email server logs and the local computer images used for sending or receiving the email. To validate the credibility of this algorithm, it was applied to a case study involving a disputed email, successfully revealing the sender and the credibility of malicious and controversial emails. However, due to inherent weaknesses in Internet protocols, identifying the source of an email can be challenging, especially when recipients employ anonymous strategies such as sending emails via TOR or I2P. In such scenarios, prosecutors may demonstrate that the email originates from an unknown source without identifying the actual person, presenting an intriguing problem for future research.

(Nikolaos 2019), advocates an approach highlighting the usefulness of threat intelligence exchange for digital forensics, as it becomes a common practice for preventing or detecting security incidents. The study presents a quantitative methodology for identifying popular cyber threat intelligence elements and a lightweight approach to correlating them with potential forensic value, resulting in quick and accurate triaging and identification of malicious activity patterns. This model can assist organizations in improving their digital forensics and response posture, minimizing the time and cost of cybercrime incidents. The study also includes the results of cross-checking 3,649 IP entries of the local IoC database against 52,374 records contained in the audit log database, obtaining related objects, running the scoring algorithm, and presenting the results in approximately 8 minutes, which is an acceptable timeframe but can be improved upon.

## E-Mail Based Attacks

Email has evolved into a universally utilized service, serving over a billion individuals globally. However, it has also emerged as a significant venue for criminal activities targeting both users and organizations. Therefore, in order to safeguard both business and personal data, it is imperative to understand the various types of common email threats. This section elucidates different categories of email-based attacks, offering a deeper understanding of these malicious activities.

### **Spam**

Spam, an undesired electronic communication, can be defined differently based on either the lack of permission or the content of the email (Alazab & Roderic, 2016). It poses a significant challenge, leading to substantial bandwidth loss and costing service providers billions of dollars. As per (Kaur & Verma, 2017), spam emails are unauthorized and unwelcome messages containing various forms of undesirable content, including junk, viruses, malicious codes, advertisements, and threats to authenticated account holders. The proliferation of email spamming is accelerating due to the widespread adoption of email as a means of information exchange. An email may be classified as spam if it meets the following criteria:

1. Originating from an unfamiliar or unauthorized sender, constituting unsolicited email.
2. Sent in large volumes to numerous recipients, categorized as bulk mailing.
3. Lacking sender identification or concealing the sender's identity, known as anonymous email.

### **Mass mailing attacks**

Mass mailing attacks are utilized to disrupt or compromise email services, rendering them unusable. These attacks consume significant network resources and often serve as carriers for malware. Various techniques employed in mass mailing attacks include chain bombs, error message bombs, list linking, zip bombs, and email bombs.

As stated by (Kumar, Singh, & Singh, 2015), "email bombing is a form of SMTP DoS attack that inundates an inbox or email server with a large volume of spam messages, aiming to overwhelm the server with unwanted traffic, thus making it inaccessible to legitimate users." Email bombing entails sending numerous identical emails directly to the same email address using automated methods. In an email bomb attack, attackers may dispatch millions or even billions of emails to a single mailbox or multiple addresses, leading to a buffer overflow. Potential consequences of email bombing include loss of internet connectivity and tarnishing the reputation of the targeted organization.

### **Phishing**

Phishing involves the theft of identities by exploiting vulnerabilities in established technologies and human behavior. The attack typically begins with the phisher sending an email to the victim, appearing to be from a legitimate organization, containing malicious links. These links may lead the victim to a malicious website where they are prompted to provide personal credentials or install spyware on their computer. According to (Anthony, 2019), "email phishing encompasses various techniques, including deceptive phishing, spear phishing, whaling, and pharming." These phishing attacks can employ various mediums, such as email, text messaging, and voice calls, to carry out the attack.

### **E-mail Spoofing**

Email spoofing involves altering the source of an email to make it appear as if it originated from a different source (Iyer et al., 2017). This malicious tactic presents the source address as if the email came from the intended recipient, while the actual sender is an attacker. Spoofing email represents a significant threat to email security, with variations including spoofing date-time and spoofing sender email. In date-time spoofing, the attacker manipulates the date in the email header's 'Date' field, while in sender address spoofing, the origin of the email is changed to make it seem as though it originated from a different source (Gupta et al., 2014).

Regrettably, the current email transmission protocol (SMTP) lacks a built-in mechanism to prevent spoofing. It relies on email providers to implement SMTP extensions such as SPF, DKIM, and DMARC to authenticate the sender (Hu & Wang, 2015). However, the source information in a spoofed email header differs from that of a genuine email, and this disparity between spoofed email headers and genuine emails serves as crucial evidence for detecting email spoofing attacks.

### **Malicious attachment**

Malicious attachments pose a significant threat to email communication, as both executable and non-executable files can be attached to emails. Attackers may embed malicious programs within these files to cause damage or unauthorized use. Executable files, such as \*.exe files, contain compiled source code that can execute directly and independently on the operating system, while non-executable files are written in specific formats that cannot be executed directly and can only be read by specific software applications. Web-based client emails (such as Gmail and Yahoo) and organizational email servers typically prohibit the attachment of executable files due to the potential risk of malicious content (Muralidharan, T., & Nissim, N. 2022).

## Threat intelligence

Threat intelligence analysis is used to determine who, what, when, where, how, and why a cyber threat or attack occurred. The analysis may include investigating cyber crimes, identifying malicious behavior, addressing operational issues, fostering due diligence in the management of security records, and recovering from unintentional system damage. Threat intelligence analysis is evolving rapidly due to the constant advancements in technology, making it increasingly important for professionals to stay up-to-date with the latest techniques and tools. With the widespread use of technology, a multitude of devices have become vulnerable to cyber attacks, requiring thorough review and analysis in each case. The field of threat intelligence analysis has become crucial in the fight against cybercrime due to the growing threat of malicious activity in the digital landscape. (Gurkok, 2017).

Ken Zatyko defines digital forensics as the utilization of computer science principles and investigative techniques for legal purposes, encompassing the examination of digital evidence under authorized search authority, adhering to chain of custody protocols, validating findings with mathematical rigor, employing validated tools, ensuring repeatability, generating comprehensive reports, and potentially presenting expert testimony (Sammons, 2012). Meanwhile, Victor et al (Karie, Kebande, Venter, & Choo, 2019) characterize digital forensics as the application of scientifically derived and verified methodologies for the preservation, collection, validation, identification, analysis, interpretation, documentation, and presentation of digital evidence sourced from digital platforms. This process aims to aid in the reconstruction of criminal events or to anticipate unauthorized actions that may disrupt planned operations.

As technology continues to advance, the risk of cyber attacks has grown exponentially. From breaches of sensitive data to ransomware incursions, organizations are confronted with an array of threats that jeopardize their security and tarnish their reputation. In response to this escalating danger, a novel discipline known as threat intelligence has arisen.

Threat intelligence encompasses the gathering, analysis, and dissemination of data pertaining to cyber threats and perpetrators. This data comprises indicators of compromise (IOCs), such as malware signatures, IP addresses, and domain names, along with contextual details about the threat, including the tactics, techniques, and procedures (TTPs) employed by the attacker.

The significance of threat intelligence cannot be overstated. By aggregating information about known threats and adversaries, organizations can preemptively fortify their defenses against potential attacks and mitigate the impact of successful breaches. Additionally, threat intelligence offers valuable insights into emerging threats and novel attack methodologies, enabling organizations to maintain a proactive stance against evolving risks

One of the primary advantages of threat intelligence lies in its capacity to automate the identification and response to potential threats. By integrating threat intelligence into security systems, organizations can swiftly detect and block malicious activities, thereby minimizing the likelihood of successful attacks. Moreover, threat intelligence facilitates the prioritization of security measures, enabling organizations to concentrate their resources on addressing the most pressing threats and vulnerabilities (Cybersecurity and Infrastructure Security Agency CISA, 2021).

Another significant benefit of threat intelligence is its ability to foster collaboration and information sharing among different organizations. Through collaborative efforts to exchange threat intelligence, organizations can bolster their collective defenses and enhance their ability to detect and respond to threats effectively. This collaborative approach is particularly crucial in industries where an attack on one organization can reverberate across the entire sector.

Computing devices, which serve as sources of data, are essential elements in the investigation process, with threat hunters leveraging technical tools, whether proprietary or open-source, to extract evidence. Threat hunting techniques involve employing analytical and investigative methods to identify, collect, analyze, and retain evidence in various forms. Utilizing advanced threat hunting techniques, numerous tools and methodologies, such as ThreatConnect, MISP, and AlienVault, have been developed to examine data obtained through commercially available or open-source licensing agreements. The goal of employing traditional threat hunting methods is to systematically compile and analyze data from computing devices to uncover potential evidence. Furthermore, during investigations, each piece of evidence is scrutinized using one or more threat intelligence tools to identify associated artifacts and assess their significance (Mohammed et al, 2016).

## E-Mail Threat Hunting

Email crimes present a significant threat to both individuals and businesses, as they are increasingly utilized by cybercriminals for illicit purposes. Consequently, email forensics has gained importance within threat hunting, prompting a surge in the development of software applications aimed at aiding this endeavor. According to Lili et al. (Xie et al, 2015), research related to email threat hunting has primarily focused on two aspects: the structural components of the network and the content of the emails. Content-based analysis includes spam filtering, author identification, mining of email users' behavioral patterns, header analysis, and authenticity verification.

Email threat hunting encompasses the analysis of emails exchanged between different parties (Chauhan & Gupta, 2017). The process of email threat intelligence primarily entails evidence collection, analysis, and presentation. A crucial aspect involves extracting and analyzing email-related data stored across various devices (Chen, 2016).

Investigation Techniques for Email Threat Hunting Various approaches are employed to investigate email-related crimes and incidents. These approaches are outlined below and succinctly defined:

#### **Header Analysis**

Header analysis serves as the primary analytical technique, involving the examination of metadata within the email header. This metadata can be manipulated by altering the sender's identity entries. Therefore, conducting a thorough analysis of the email header is crucial, as it aids in identifying various email-related crimes such as spam, scams, phishing, email spoofing, and data leaks from internal organizational networks.

#### **Server Investigation**

This method involves examining duplicates of delivered emails and database logs to ascertain the origin of an email message. If retrieving emails directly from customers (either senders or receivers) proves challenging, emails can be retrieved from servers (such as Proxy or ISP), as many service providers store backups of all emails post-transmission. Additionally, logs maintained by servers can be reviewed to identify the computer address associated with the email transaction.

**Network Device Investigation**

In this cyber threat intelligence method, logs maintained by network devices like switches, routers, and firewalls are utilized to investigate the source of an email message. This approach presents challenges, particularly when ISPs or proxies do not retain logs, rendering primary evidence inaccessible.

#### **Software Embedded Analysis**

The sender's email software, utilized for composing the email, may contain information regarding the sender, as well as any attached files or documents. This data can be incorporated in the form of Transport Neutral Encapsulation Format (TNEF) within custom headers or within MIME content. Delving into these provided details unveils crucial insights into the sender's preferences and the recipient's choices, which could aid in gathering evidence on the client side.

#### **Sender Mail Fingerprints**

The "Received" field traces, in reverse sequence, data generated by mail servers that previously managed an email. The term "X-Mailer" or "User-Agent" field denotes the applications and versions employed by clients to transmit the email.

#### **Volatile Memory Analysis**

Recent studies have investigated the extraction of spoofed emails from volatile memory. As computers progress, evidence related to emails, particularly header information, can be retrieved from volatile memory.

#### **Attachment Analysis**

Email attachments have emerged as the primary vector for virus and malware attacks, making the investigation of attachments essential in any email-related inquiry. Various software tools exist to extract email-related data, including attachments, from hard disk drives. Investigators can also upload documents to online databases like VirusTotal to ascertain whether the file contains malware.

#### **E-Mail Threat Hunting Tools**

The significant volume of data generated through email necessitates the development of tools to aid investigators in analyzing the origin and content of email messages, facilitating effective and efficient investigations into email-based crimes. These tools play a crucial role in identifying the source and destination of messages, monitoring message pathways, and detecting email-based attacks by offering user-friendly browser-based automated reports and other functionalities.

A comparative study conducted by (Devendran et al., 2015) assessed forensic email tools based on nine different criteria, revealing that these tools are independently designed with varying features. Aid4Mail, for instance, has the capability to analyze emails stored on both local computers and remote email servers. Notably, Paraben EMX stands out as the only tool that provides detailed examination of email attachments in addition to headers and body content. Both Paraben EMX and Aid4Mail offer superior recovery features, enabling the restoration of emails/files from deleted folders. Paraben supports a wider range of email formats and approximately 750 MIME types, making it a versatile option for email format support. While most tools are compatible with the Windows platform, only a few support Linux. DFF and Aid4Mail offer extended device analysis, including pen drives. Given the varying complexities of email crime scenarios, investigators require a diverse array of tools to conduct email investigations effectively.

### **E-mail Structure**

An email message typically comprises two primary components: the header, positioned at the outset of the email, and the body, which contains the actual content of the email. The header contains metadata pertaining to the email message, including details such as the sender's email address, the email subject, recipient email addresses, and time-related information. Structured fields within the header include 'From', 'To', 'Subject', 'Date', 'CC', 'BCC', and 'Return-To', with certain fields, such as "From" and "Date", being mandatory in every email message. Email header analysis is crucial in digital investigations as it provides substantial information about the email. Additionally, the number of headers in an email can vary and is contingent upon the server that dispatched the email (Bhaskari & Avadhani, 2013).

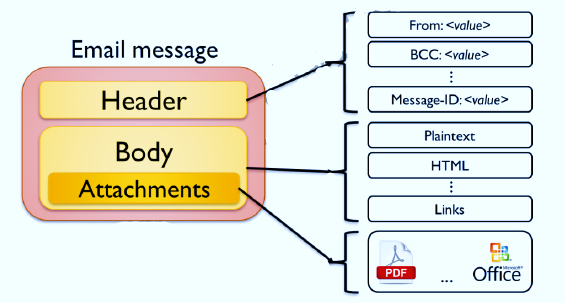


Figure 1: E-mail message structure

The body of an email typically comprises the message itself along with additional content, such as multimedia and executable files permitted by the email server. This section may contain plain text or incorporate multimedia elements in Hyper Text Markup Language (HTML) and Multipurpose Internet Mail Extensions (MIME) encoded attachments. Attachments are stored within the body, and HTML-formatted messages may include embedded multimedia files. While any type of file can be attached to an email, most email servers prohibit the attachment of executable files (e.g., \*.exe) due to associated risks.

In cyber threat intelligence investigations, email headers and contact lists often hold greater significance than email content as they identify a user's associates and provide insights into communication frequency (McCarrin, Green, & Gera, 2018). Email threat hunting investigations entail examining both the header and body, which involves scrutinizing the sender's email address, message ID, message initiation protocol (HTTP and SMTP), and the sender's IP address. Existing web-based platforms for email header analysis offer these functionalities, yet they may have limitations such as slow parsing speed, a disjointed user interface, and suboptimal output presentation. To enhance investigation and exploration, current tools need to be reimagined using a modular forensic processing framework that implements the "Visibility, Filter, and Report" model.

### **Analysis and Limitation of the Existing System**

Email threat hunting investigations entail analyzing both the header and body of emails. This examination should encompass scrutinizing the sender's email address, message ID, message initiation protocols (such as HTTP and SMTP), and the sender's IP address. While existing web-based platforms for email header analysis offer these capabilities, they may present challenges due to limitations such as slow parsing speeds, a fragmented user interface, and suboptimal output formatting. To address these challenges, there is a need to reimagine current tools using a modular forensic processing framework that adheres to the "Visibility, Filter, and Report" model. This approach aims to streamline investigation and exploration processes.

### **Justification for the New System**

Threat hunters require tools that can efficiently handle large volumes of email data to streamline investigations and yield dependable outcomes. This project entails designing and developing a web-based email threat hunting software that aims to:

1. Offer a user-friendly interface for parsing email headers and rendering them in a readable format.
2. Detect hop delays, hop countries, and the email's source.
3. Enhance the efficiency of retrieving and generating results from extracted email headers.
4. Provide comprehensive results gathered from the internet, suitable for use as digital evidence.
5. Enable the generation of results in PDF format for future analysis or utilization.

## Description of the New System

The web-based email threat hunting software offers a straightforward interface for uploading emails, analyzing the information extracted from email headers, and exploring additional findings sourced from the internet based on header parameters. Tasks performed using this software can be completed within minutes, significantly reducing the time it typically takes for hunters to manually sift through email headers. This enhances efficiency in threat hunting endeavors.

As depicted in Figure 3, users will engage with the proposed system via its web-based interface to submit original email messages for analysis and review the reports generated, sourced either from the internet or from the internal networks, by expert threat hunter investigators.

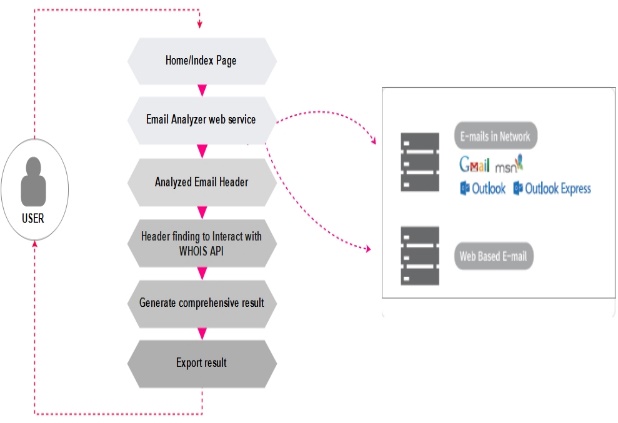


Figure 3: Diagram depicting the working procedure of the proposed software

The suggested system comprises a graphical user interface accessible through the web, an API for IP/WHOIS, and an output result that is generated.

## System Requirement for Development

System Requirements delineate the characteristics and functionalities of a system or software application. These encompass a range of components aimed at defining the desired attributes necessary to meet the intended requirements of the system users. Functional requirements outline what the system is expected to accomplish, while non-functional requirements detail the essential attributes and qualities the system must possess.

1. Functional Requirements
   1. The proposed system must analyze each e-mail header and promptly deliver relevant results.\
   2. Results must consistently adhere to the "Visibility, Filter, and Report" model.
2. Non-Functional Requirements
   1. Usability: The system interface must be intuitive and user-friendly, facilitating seamless interaction.
   2. Security: To maintain data integrity, the system will generate a hash with the report to prevent tampering.
   3. Performance: The system should efficiently handle concurrent user requests while optimizing response times.
   4. Availability: System functionality and services must be accessible at all times during use.
   5. Scalability: The system should exhibit sufficient scalability to accommodate growth without compromising efficiency, ensuring it can serve more clients, process additional information, and handle increased workload seamlessly.

## Design of the Proposed System

The architecture of a web application outlines the connections and dependencies among its components, encompassing databases and middleware systems within the web environment. It illustrates a sequence of steps initiated by a user who seeks a particular URL through their browser, which then sends a request. This request traverses the network, prompting the server to transmit data back to the browser for page rendering. Finally, the user interacts with the displayed page. The software development process will adhere to the N-tier architecture, ensuring the system's adaptability and flexibility are maximized.

### **The Proposed System Design (N-tier architecture)**

N-tier architecture is a well-established framework within the software industry, offering scalability, security, fault tolerance, reusability, and maintenance solutions for web applications. In a typical 3-tier architecture model, the presentation layer is deployed through various devices such as desktops, laptops, tablets, or mobile devices, either via a web browser or a web application utilizing a web server. The application layer is usually hosted on one or more application servers, which can be situated in the cloud or on dedicated workstations, depending on the application's complexity and processing requirements. The data layer typically comprises relational databases, large data sources, or other database systems, stored either on-site in a data center or in the cloud. For the proposed software, the 3-tier architecture will delineate the architectural implementation of the system;

#### **Presentation Tier**

The user interface typically presents visual content and information in a user-friendly manner, accessible through a web browser or cloud-based software. It relies on various web technologies to achieve this, including:

1. HTML (Hypertext Markup Language): This serves as the core markup language for the World Wide Web, defining the structure of web pages and providing meaning to web content.
2. CSS (Cascading Style Sheets): These style sheets are used to define the presentation of HTML documents, determining aspects such as colours, layouts, and fonts.
3. JavaScript: This lightweight, interpreted programming language enhances web pages by enabling dynamic behavior, storing information, and managing website requests and responses.
4. Bootstrap: An open-source CSS framework that facilitates responsive, mobile-first web development, providing pre-designed templates and components for building user interfaces.

#### **Application Tier**

The application tier comprises the functional business logic that forms the foundation of an application's operation.

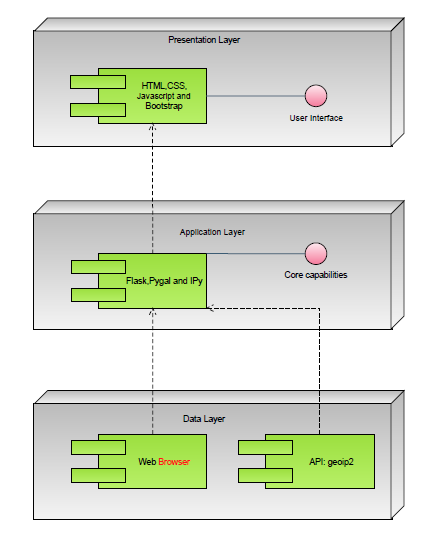


Figure 4: The UML diagram of the proposed system

The fundamental functionalities. For the envisioned software, the programming languages and framework utilized in the backend development are as follows:

1. Flask: Flask is a Python micro-framework utilized for web development. This programming language offers simplicity, extreme flexibility, absolute control over web application structuring, and fine-grained control of dependencies.
2. Pygal: Pygal is an open-source program designed for creating vector graphics for websites.
3. IPy: IPy is a Python class used for address and network handling, specifically for IPv4 and IPv6 addresses. The IP class facilitates comfortable parsing and handling for most notations used in IPv4 and IPv6 addresses and networks. It can detect, analyze, and differentiate between IPv4 and IPv6 addresses using various methods of expression.

#### **Data Tier**

The data tier encompasses the database or data storage system and data access components. In the proposed software, Python dependencies are used for development, and data access is facilitated through API calls made by the application layer. The API employed in the software is the geoip2 object, which serves as a wrapper for the MaxMind geoip2 Python library. This API is utilized to obtain the geographical location of identified IP addresses.

When interacting with the geoip2 API, the software takes a domain name or IP address extracted from the header findings as input. Subsequently, the API determines the appropriate Whois database or domain to query, sends requests to the relevant servers using a large pool of IP addresses, analyses the responses, and returns the results in a human-readable format in real time.

The API offers essential data points for the domain address identified in the analysed e-mail header, including registrant name, organization, registrar information, and domain reputation. This gathered information can aid in detecting spam, malicious websites, intrusions, and other online misbehaviours. Digital forensics experts can leverage the Whois data to compile easily interpretable lists of domains potentially linked to cybercrime, thereby bolstering cyber defence measures. Such information proves particularly valuable in mitigating phishing and other malware-enabled attacks.

Top of Form

.

### **Flowchart**

A flowchart is a visual depiction of an algorithm, workflow, or process, using connected symbols and arrows to illustrate the sequence of operations. Figure 5 presents the flowchart of the proposed system, illustrating how the system operates;

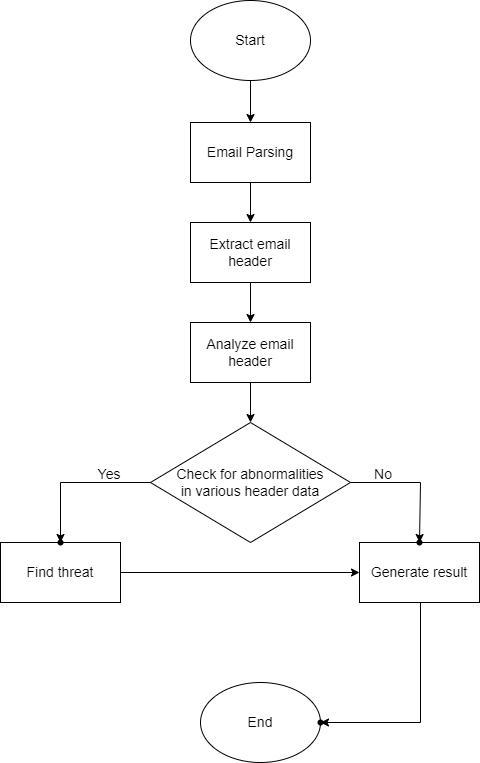
****

Figure 5: Flowchart diagram for the proposed system

### **Use Case Diagram**

A use case diagram is a type of behavioural diagram that illustrates a collection of actions, known as use cases, that a system can undertake in collaboration with external users, also known as actors. It typically includes actors, use cases, and associations, and delineates the system's boundary.

1. **Actors**: Represent the external entities interacting with the system, such as individuals or external systems.
2. **Use Cases**: These depict the operational functionalities of the system, illustrating actions performed by actors. For instance, actions like "browsing the cart," "selecting items for purchase," or "completing a purchase."
3. **Associations**: These are depicted by solid lines connecting actors to use cases, indicating the interactions between actors and the system functionalities.
4. **System Boundary**: This represents the extent or scope of the system with which the actors interact.

The use case diagram consists of one actor and four use cases, as illustrated in Figure 6, depicting the interaction between users and the system's functionalities;

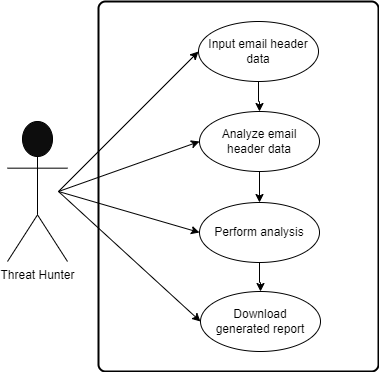
****

Figure 6: Use case diagram for system users

## Data Collection

Information gathering and data collection for this research will involve sourcing from a diverse range of materials, including publications, case studies, articles, journals, and website resources. The gathered data will undergo thorough analysis to ensure the research yields comprehensive results, which will subsequently be utilized in the report-writing process.

**SYSTEM IMPLEMENTATION AND TESTING**

## System Requirement

Prior to developing the proposed system, it is essential to identify and design appropriate techniques for the research project. Core requirements were carefully analyzed and gathered to identify the necessary software and hardware dependencies essential for ensuring the completeness and efficiency of the developed software application, aligning with the study's aims and objectives.

### **Software Requirement**

The software needed for the proposed system includes:

1. Application server: This software enables the provision of services to clients over the internet. Gunicorn, a Python WSGI HTTP Server for UNIX, fulfills this role. It operates on a pre-fork worker model and is compatible with various web frameworks. Gunicorn is known for its simplicity, efficiency, and compatibility with different web frameworks, making it resource-efficient and fast.
2. Web Browsers: These applications function on the application layer to render a graphical user interface from the webserver. They act as clients accessing the server and are executed on the client's machine. Compatible browsers for this project include Chrome, Mozilla Firefox, Microsoft Edge, and Opera.

### **Hardware Requirement**

The computing device requirements for optimal functioning of the proposed system are outlined as follows:

1. A processor (CPU) with a frequency of 2 gigahertz (GHz) or higher.
2. At least 3GB of RAM.
3. Monitor resolution of 1024 x 768 or higher.
4. A minimum of 50GB of free space on the hard disk.

## System Overview: Different Interfaces of the Proposed System

The developed web-based e-mail threat hunting software offers fundamental investigative capabilities essential for e-mail threat analysis. Aligned with standard practices in e-mail investigations, the system facilitates the following functionalities:

1. Retrieval of registration details for a domain or IP address.
2. Identification of DNS and mail servers linked to a domain.
3. Provision of current location information (currently limited to country) for an IP address or canonical hostname.
4. Visual representation of hop count.
5. Extraction of diverse header parameters, including security headers and X headers.

### **Homepage**

Users will engage with the proposed system via its web-based user interface, through which they can submit an original e-mail message for analysis and review the reports generated, sourced from either the Internet or the internal networks of digital investigation experts.

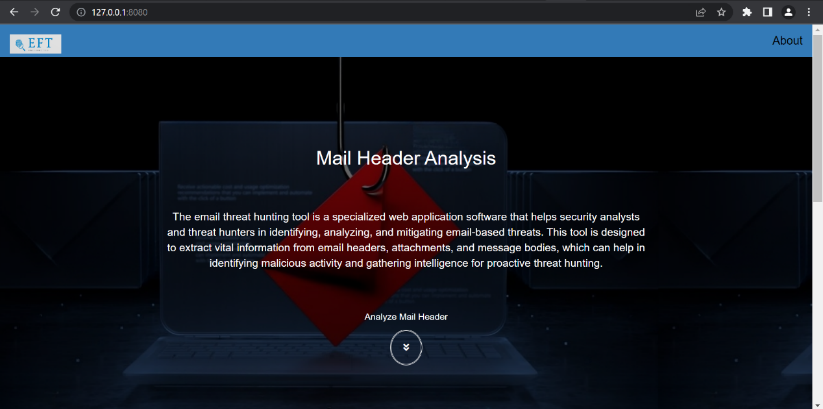


Figure 7: Homepage

The homepage serves as the primary landing page for users of the developed system, providing essential information and navigation options. It features a prominent Call-To-Action (CTA) element, directing users to access other system functions. Figure 7 illustrates the homepage with a brief overview of the developed system, while Figure 8 offers a more detailed view, emphasizing its unique features.

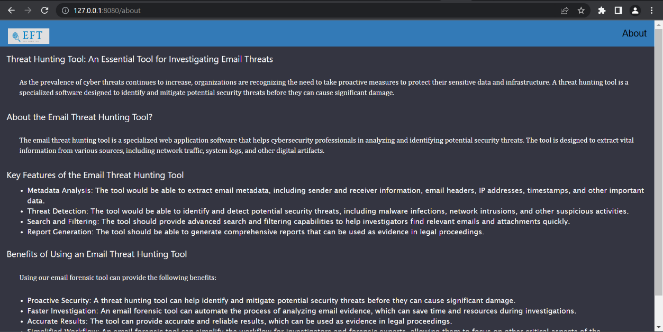


Figure 8: About Section of the homepage

### **E-mail analyser web service**

Figure 9 depicts the section where users of the developed system are prompted to input the email header obtained from various sources, such as web-based email services or desktop email clients. Within this section, a module is integrated to extract the email header, and a mechanism is in place to analyze the email header data upon clicking the "analyze" button, as illustrated in Figure 9

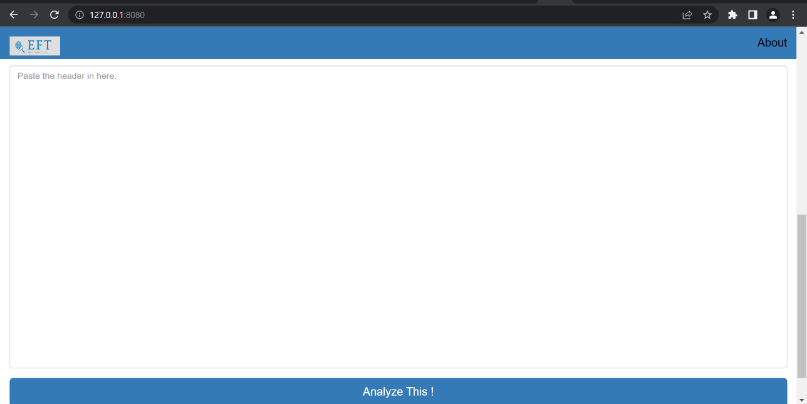


Figure 9: Required field for header data

### **Parse E-mail Header data**

The header of an email contains essential metadata about the message, including details such as the sender's email address, the subject line, recipient addresses, and timestamp information. Structured fields within the header typically include 'From', 'To', 'Subject', 'Date', 'CC', 'BCC', and 'Return-To', although certain mandatory fields like "From" and "Date" must be present in every email message. In situations requiring email investigation, users of the system gather a copy of the email header file and input it into the designated field of the system, as illustrated in Figure 10;

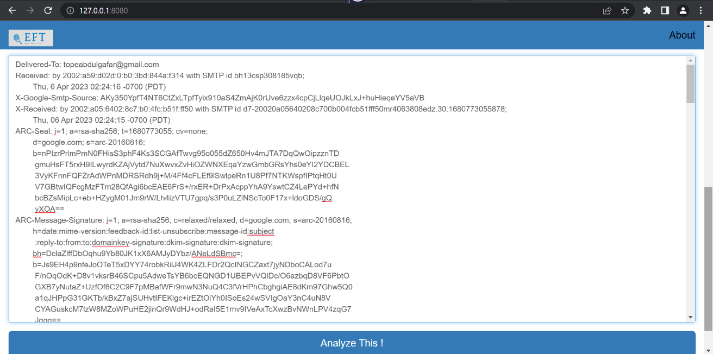


Figure 10: Collected header data

### **Results of Analysed e-mail**

Figure 11 showcases the results and summary of the processed email header. A distinguishing aspect of the developed system is its inclusion of a freshly computed hash for each generated report. This feature aims to guarantee the verifiability, reliability, and integrity of the reports, providing assurance that they remain unaltered and authentic.

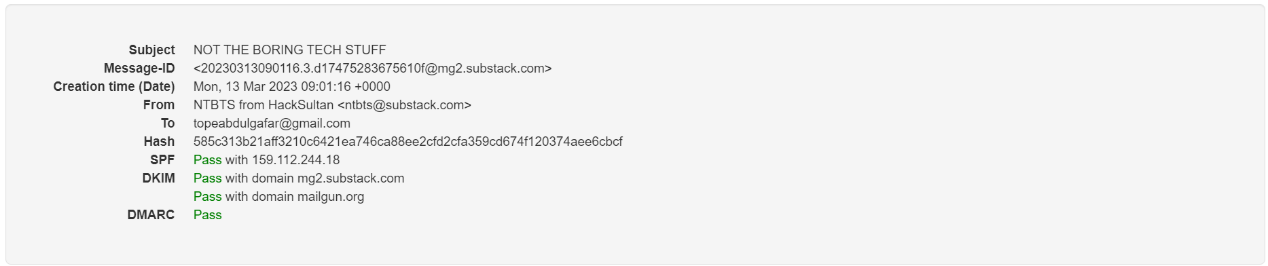
****

Figure 11: Summary of analysed e-mail

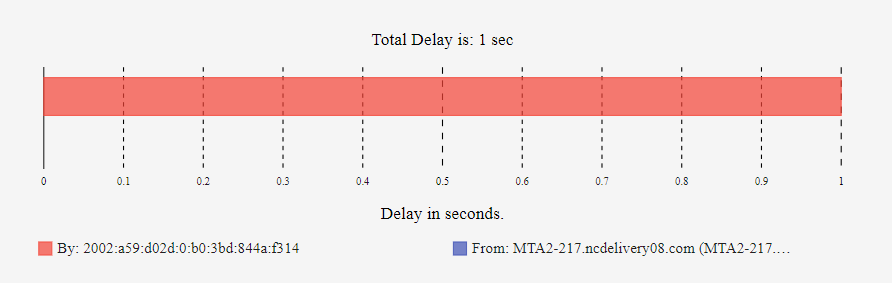


Figure 12: Graphical display of hop count

The hop count in an email pertains to the number of mail servers through which a message traverses from the sender to the recipient along its route. By examining the "Received:" header of an email message, which outlines the message's complete path, one can determine the hop count by tallying the total number of hosts listed therein. Hop counts are particularly important in emails to prevent mail loop problems. Figure 12 illustrates the graphical representation of the hop count analysis, while Figure 13 showcases the ordered list of the analysed header hop count.



Figure 13: List display of hop count

As depicted in Figure 14, the system presents various information that can facilitate a reverse lookup process to gather insights into a user's logical and physical location. Through the developed system, it is possible to ascertain the geolocation of the computer, offering details such as country, region/state, city, latitude/longitude, telephone area code, and location-specific zip code, albeit with differing levels of accuracy.

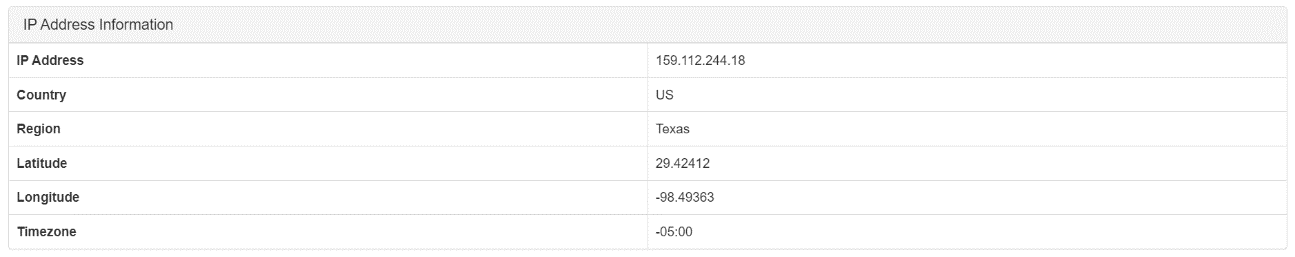


Figure 14: Collated IP Information

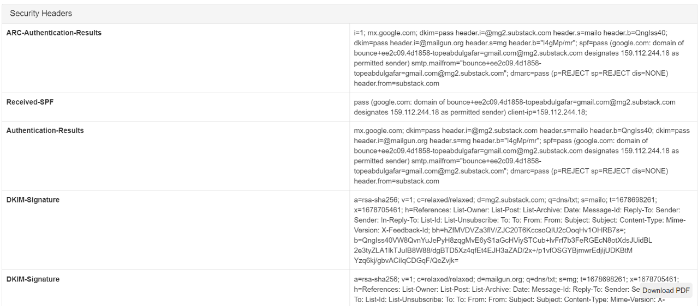
Security headers establish a system wherein handlers of an email message can verify the authenticity of the message as it traverses between them toward its destination. At each stage of the handling process, an authenticated status record is generated and attached to the message, allowing the final recipient to make informed decisions regarding the message's bl.

Figure 15: Displayed Security headers

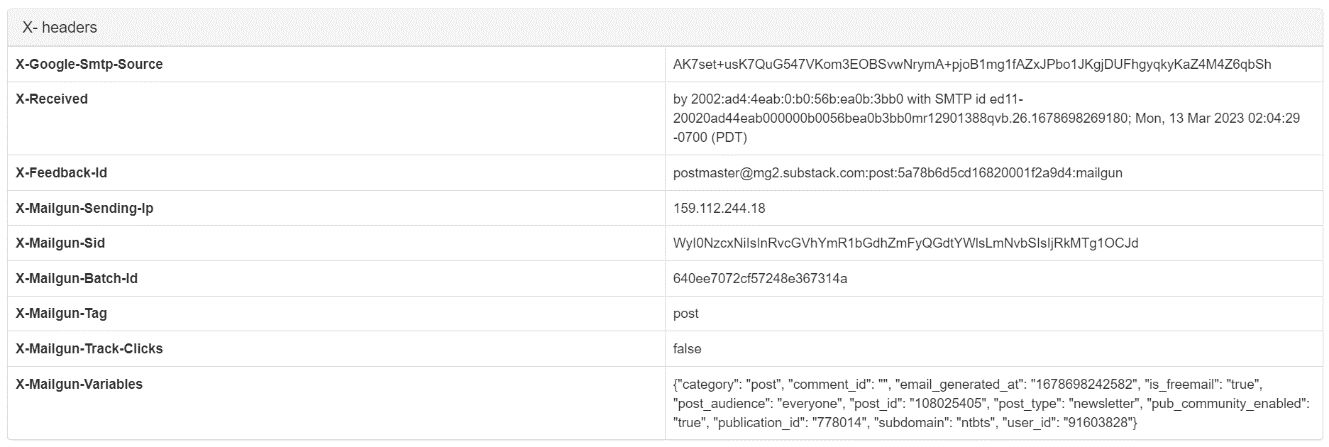
****

Figure 16: Displayed X-headers

X-headers are additional email headers appended to the message alongside standard headers like "From" and "Subject," tailored to the sender's requirements. Email service providers may also include X-headers for purposes such as authentication results or spam filter data. Traditionally, X-headers begin with "X" to indicate that their values are experimental or extensions of standard headers. Figure 14 displays the segment containing various X-headers collected from the analyzed email

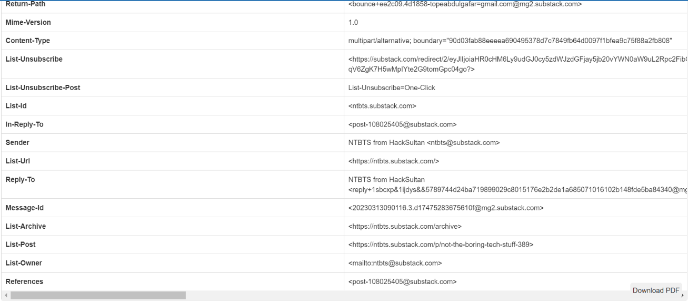


Figure 17: Other identified headers

## SYSTEM EVALUATION

This section outlines the specifications of the developed web-based application, emphasizing the importance of drafting requirements tailored to threat hunting. It is crucial to define requirements that address the integrity and operational processes of the tool;

### **Application Core Requirements (A-CR)**

1. A-CR-01: The tool must collect data from a user endpoint.
2. A-CR-02: The tool must log all actions and processes.
3. A-CR-03: The tool must parse and analyze email headers.
4. A-CR-04: The tool must generate hashes for ingested data.
5. A-CR-05: The tool must display analyzed email headers.
6. A-CR-06: The tool must provide hashes for analyzed email headers to ensure integrity.
7. A-CR-07: The tool must generate reports for analyzed email headers.

### **Application Core Test Assertions (A-CA)**

1. A-CA-01: The tool must log and hash each report generated during its use.

Rationale: This measure ensures the verifiability, reliability, and integrity of the reports, providing evidence that they have not been altered.

1. A-CA-02: The tool must identify instances where an internet connection is unavailable to retrieve web-based resources.

Rationale: This functionality ensures uninterrupted analysis of e-mail headers and prevents unexpected disruptions during tool usage.

### **Application Test Cases (SC-TC)**

1. A-TC-01: Initiate the application and observe its successful launch.
2. A-TC-02: Validate the presence of logs and hashes to confirm report integrity.
3. A-TC-03: Ensure the analysis process is executed and the report is correctly displayed.
4. A-TC-04: Submit improperly parsed e-mail data and verify the system's ability to identify inaccuracies in the header information.
5. A-TC-05: Confirm the capability to download the generated report and obtain the document.

### **Compliance Matrix**

The testing phase follows the NIST Threat Hunting Tool Testing Program, which outlines specific criteria for each tool. This program includes requirements specifications, test assertions, and test cases. The results of applying these testing criteria to the tool are documented, along with their compliance matrix, as shown in Table 4.1:

**Table 4.1:** Compliance Matrix Result

|  |  |  |  |
| --- | --- | --- | --- |
| **S/NO** | **Requirement** | **Test Case** | **Result of Test Assertion** |
| 1 | A-CR-01, A-CR-02 | A-CA-02 | A-TC-01 |
| 2 | A-CR-03 | ---Check--- | A-TC-04 |
| 3 | A-CR-04, A-CR-06 | A-CA-01 | A-TC-01, A-TC-02 |
| 4 | A-CR-05, A-CR-07 | ---Check--- | A-TC-03, A-TC-05 |

## Conclusion

The developed system serves as an efficient raw e-mail parser designed specifically for digital investigations. It offers a user-friendly interface for uploading e-mails, extracting information from e-mail headers, and further exploring additional details gathered from the internet based on header parameters. Notably, each generated report is accompanied by a newly computed hash to guarantee the authenticity and integrity of the reports, ensuring they remain unaltered.

## Recommendations

1. Enhancements to the developed system could introduce proactive cyber threat hunting capabilities through the following measures:
2. Implement a machine learning model trained on datasets comprising both malicious and benign email headers, employing selected algorithms and techniques.
3. Automate the report generation process to eliminate manual intervention and streamline the workflow.
4. Integrate a paid version of the Whois API to access comprehensive and detailed information about all parties involved in email transmissions.
5. Elevate the application development efforts to transform the existing tool into a comprehensive threat hunting suite, offering advanced functionalities and capabilities.

## Suggestions for Further Work

Investigating email headers is crucial for gathering evidence, as it allows for the extraction of valuable hidden personal information contained within emails. To advance research in this area, it would be beneficial to develop web-based software that integrates header analysis with other email cyber threat hunting techniques. This integrated approach would enable more comprehensive investigations into email-related activities and facilitate the production of evidence that aligns with various threat hunting processes and standards.

# REFERENCES

Anthony, B. (2019). *Social Engineering:The human element of cybersecurity* (Utica College). Retrieved from https://search.proquest.com/docview/2228120220?accountid=169469

Banday, M. T. (2011). Technology Corner : Analysing E-Mail Headers for Forensic Investigation. *Journal of Digital Forensics, Security and Law*, *6*(2), 49–64. https://doi.org/https://doi.org/10.15394/jdfsl.2011.1095

Bhardwaj, A., & Goundar, S. (2017). Security challenges for cloud-based email infrastructure. *Network Security*, *2017*(11), 8–15. https://doi.org/10.1016/S1353-4858(17)30094-6

Bhaskari, N. S. D. L., & Avadhani, P. S. (2013). *Inverted Pyramid Approach for E-Mail Forensics Using Heterogeneous Forensics Tools. CSI Communications*. (July), 21–23.

Chhabra, G. S., & Bajwa, D. S. (2015). Review of E-mail System , Security Protocols and Email Forensics. *International Journal of Computer Science & Communication Networks*, *5*(3), 201–211.

Crocker, D. (2009). *Internet Mail Architecture*. Retrieved from http://trustee.ietf.org/license-info

Guo, H., Jin, B., & Qian, W. (2013). Analysis of Email Header for Forensics Purpose. *International Conference on Communication Systems and Network Technologies*, 340–344. https://doi.org/10.1109/CSNT.2013.78

Haggerty, J., Karran, A. J., Lamb, D. J., & Mark, T. J. (2011). A framework for the forensic investigation of unstructured email relationship data relationship data A Framework for the Forensic Investigation of Unstructured Email Relationship Data. *International Journal of Digital Crime and Forensics (IJDCF)*, *3*(3), 1–18. https://doi.org/10.4018/jdcf.2011070101

Iyer, R. P., Atrey, P. K., Varshney, G., & Misra, M. (2017). Email Spoofing Detection Using Volatile Memory Forensics. *IEEE Conference on Communications and Network Security (CNS): The Network Forensics Workshop Email*, 619–625.

Johannes, S., & Dewald, A. (2017). A Forensic Email Analysis Tool Using Dynamic Visualization A FORENSIC EMAIL ANALYSIS TOOL USING. *Journal of Digital Forensics, Security and Law Article*, *12*(1), 7–14.

McCarrin, M., Green, J., & Gera, R. (2018). Creating and understanding email communication networks to aid digital forensic investigations Michael McCarrin, Janina Green 1 and Ralucca Gera. *The Journal of Digital Forensics, Security and Law: JDFSL*, 1–6.

Msongaleli, L. D., & Kerem, K. (2018). Electronic Mail Forensic Algorithm for Crime Investigation and Dispute Settlement. *International Symposium on Digital Forensic and Security (ISDFS)*, *3*, 1–5.

Tiwari, L. K., & Samaddar, S. G. (2011). E-mail Forensics For Real Life Application in Evidence Building. *International Journal of Advanced Research in Computer Science*, *2*(3), 287–293.

Upadhyay, R., & Kumar, C. (2018). Legalistic Evaluation of E-Mail Headers Based on Address Spoofing. *International Journal of Advance Engineering and Research Development*, *5*(1), 104–113.

Xie, L., Liu, Y., & Chen, G. (2015). A Forensic Analysis Solution of the Email Network Based on Email Contents. *International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 1613–1619.

Zinzuvadia, K., & Gardas, N. (2018). Open Source Based Online Forensic Tool for Analysis of Malicious Activity over Email Traffic. *International Journal of Advance Research and Innovative Ideas in Education*, *4*(3), 522–529. Retrieved from www.ijariie.com

Muralidharan, T., & Nissim, N. (2022). Improving malicious email detection through novel designated deep-learning architectures utilizing entire email. Neural Networks, 157. https://doi.org/10.1016/j.neunet.2022.09.002

Kumar Birthriya, S., & Jain, A. K. (2021). A Comprehensive Survey of Phishing Email Detection and Protection Techniques. Information Security Journal: A Global Perspective, 1–30. https://doi.org/10.1080/19393555.2021.1959678

Bin Abd Razak, S., & Bin Mohamad, A. F. (2013). Identification of spam email based on information from email header. 2013 13th International Conference on Intellient Systems Design and Applications. doi:10.1109/isda.2013.6920762

Pavanello, F., Giaimo, M., Phoenix, W., & Cagol, S. (2023). OSINT-based Email Analyzer for Phishing Detection. https://www.utupub.fi/bitstream/handle/10024/174329/OSINT-based%20Email%20Analyzer%20for%20Phishing%20Detection.pdf