**YETI: Your everyday threat intelligence**

**Open-Source Threat Intel Platform Technical Review Report**

# Abstract – (by Keerthi)

This report highlights the detailed summary of the cybersecurity tool Yeti (Your Everyday Threat Intelligence), focusing on its sustainability for Small and Medium-sized Enterprises (SMEs) in developing countries. Yeti emerges as an open-source platform in response to the challenges faced by security analysts in managing scattered threat data. This platform addresses the need for a centralized repository, consolidating Indicators of Compromise (IOCs), Tactics, Techniques, and Procedures (TTPs) associated with cyber threats. Beyond aggregation, Yeti automates data enrichment through functionalities like IP geolocation and domain resolution, saving analysts valuable time. The user-friendly interface caters to both human analysts and integrates seamlessly with other security tools through a web API. This focus on automation extends to data exports, allowing users to define formats compatible with incident management, malware investigation, and threat monitoring systems. [1]

Designed for customization and extensibility, Yeti empowers analysts to prioritize analysis over data formatting. The platform fosters collaboration through a well-organized repository and leverages the power of the security community for ongoing development. In summary, Yeti offers a comprehensive solution for streamlined threat intelligence analysis, improving efficiency, collaboration, and overall security posture. [5]

Small and Medium-sized Enterprises (SMEs) in developing economies face a significant cybersecurity challenge. Limited resources often prevent them from implementing robust security measures, leaving them vulnerable to ever evolving cyberattacks. Yeti emerges as a free, open-source threat intelligence platform specifically designed to address this gap. By centralizing critical data like Indicators of Compromise (IOCs) and Tactics, Techniques, and Procedures (TTPs), Yeti empowers even resource-constrained SMEs to gain a comprehensive understanding of the cyber threat landscape. Its features, including centralized threat intelligence, automation for efficiency, a user-friendly interface, and customization and integration options, make it a cost-effective solution for SMEs, ensuring affordability for budget-conscious organizations. By automating tasks, simplifying analysis, and offering a centralized view of threats, Yeti empowers SMEs to effectively combat cyberattacks. This improved threat intelligence translates to better decision-making, prioritized actions, and a more robust overall cybersecurity posture, crucial for business continuity and growth in today's digital world. [4,6]

[Abstract – (by Keerthi) 1](#_Toc164648361)

[Tool Overview – (by Mitchell) 4](#_Toc164648362)

[● **Tool Description:** 4](#_Toc164648363)

[● **Key Features:** 4](#_Toc164648364)

[Installation and Setup – (by Misa) 4](#_Toc164648365)

[● **System Requirements:** 5](#_Toc164648366)

[● **Installation Process:** 5](#_Toc164648367)

[● **Configuration Needs:** 6](#_Toc164648368)

[Functional Capabilities 6](#_Toc164648369)

[**Core Functionalities:** (by Keerthi) 7](#_Toc164648370)

[**Accuracy and Reliability:** (by Mitchell) 8](#_Toc164648371)

[Usability and Accessibility - (by Keerthi) 8](#_Toc164648372)

[**Interface Design:** 8](#_Toc164648373)

[**Documentation and Support:** 8](#_Toc164648374)

[**Ease of Use:** 9](#_Toc164648375)

[Integration, Compatibility, and Scalability - (by Misa) 9](#_Toc164648376)

[● **Interoperability with Existing Tools:** 10](#_Toc164648377)

[● **APIs and Customization:** 10](#_Toc164648378)

[● **Scalability:** 11](#_Toc164648379)

[● **Flexibility:** 11](#_Toc164648380)

[Community Support and Sustainability – (by Keerthi) 13](#_Toc164648381)

[**Developer Community:** 13](#_Toc164648382)

[**Updates and Maintenance:** 13](#_Toc164648383)

[**Local and Global Support Networks:** 13](#_Toc164648384)

[######## 14](#_Toc164648385)

[Assessment of Gaps – (by Mitchell) 14](#_Toc164648386)

[● **Identified Gaps:** 14](#_Toc164648387)

[**Potential for Improvement:** 14](#_Toc164648388)

[● Use Cases and Practical Applications 15](#_Toc164648389)

[● **Examples of Successful Use:** (by misa) 16](#_Toc164648390)

[Conclusion and Recommendations for Development - (by Keerthi) 17](#_Toc164648391)

[● **Overall Suitability:** 17](#_Toc164648392)

[● **Recommendations for New Platform:** 17](#_Toc164648393)

[Reference 17](#_Toc164648394)

# Tool Overview – (by Mitchell)

## **Tool Description:**

* Yeti is a platform designed to enhance cyber threat intelligence analysis, digital forensics and incident response efforts. Its primary purpose is to streamline the gathering, analyzing, and sharing cyber threat intelligence data to better understand and mitigate cyber threats. [5]
* Yeti stores technical and tactical cyber threat intelligence from internal or external systems, serving as a backend for digital forensics and incident response -related queries (including Yara signatures, Sigma rules, and DFIQ), providing a web API for automation and enrichment tasks, and supporting the export of data in user-defined formats for ingestion by third-party applications such as SIEMs and digital forensics and incident response platforms. [5]

## **Key Features:**

* Bulk Search: Yeti enables users to perform bulk searches of observables (such as IP addresses, domains, hashes) to identify potential threats and understand their nature.
* Threat Analysis: Users can focus on specific threats and quickly access comprehensive lists of associated tactics, techniques, and procedures (TTPs), malware samples, and other digital forensics and incident response artifacts.
* Simplified Cyber Threat Intelligence Management: Yeti allows cyber threat intelligence analysts to concentrate on adding intelligence rather than worrying about machine-readable export formats, thanks to its streamlined data management capabilities.
* Customization and Integration: The platform facilitates the incorporation of custom data sources, analytics, and logic, making it adaptable to the specific needs and workflows of different organizations.

# Installation and Setup – (by Misa)

## **System Requirements:**

* + *Minimal Requirements:*

Operating System: Any system compatible with Docker (e.g., Linux, macOS, Windows)

Disk Space: At least 2 GB of free disk space.

RAM: Minimum 2 GB RAM

CPU: Dual-core processor

* + *Recommended Requirements:*

Operating System: Linux (e.g., Ubuntu, CentOS)

Disk Space: 10 GB or more of free disk space.

RAM: 4 GB RAM or more

CPU: Quad-core processor or higher

## **Installation Process:**

* + *Install Prerequisites:*

Ensure that you have the following tools installed on your system:

* + - 1. git
      2. Docker
  + *Download YETI Docker Containers:* Open command prompt in your windows PC and type the below given command after navigating to the folder where you want to clone the project.

git clone https://github.com/yeti-platform/yeti-docker

cd yeti-docker/prod

* + *Start Docker Containers:*

docker-compose up -d

This command will start the YETI containers and create a Docker network called yeti\_network.

* + *Create Admin User:* To access the YETI web interface, create an admin user with the following command:

docker-compose run --rm api create-user USERNAME PASSWORD -- admin

* + *Access YETI Web UI:*

Open your web browser and go to http://localhost:80/. Log in using the admin credentials you just created.

## **Configuration Needs:**

* + Configuring YETI for immediate use is straightforward, especially with the Docker setup. However, for specific customization or advanced configurations, you may need to adjust settings within the Docker containers or modify configuration files.

Link to Installation Manual: [YETI Installation Manual](https://yeti-platform.io/docs/getting-started/)

Link to Custom Configurations: [YETI Configuration Guide](https://yeti-platform.io/guides/)

By following these steps, you can easily install and set up YETI on your system.

# Functional Capabilities

## **Core Functionalities:** (by Keerthi)

**Addressing the Cybersecurity Gap for SMEs:**

SMEs in developing economies often face significant challenges in securing their digital infrastructure due to limited resources. Traditional security solutions can be expensive and complex to manage, leaving them vulnerable to cyberattacks. Yeti emerges as a free, open-source threat intelligence platform specifically designed to bridge this gap. [1,4,6]

**Core Functionalities Addressing SME Needs:**

* **Centralized Threat Intelligence:** Consolidates data from various sources into a single repository, improving threat visibility and streamlining analysis. This empowers SMEs to make informed decisions about potential threats without requiring a dedicated security team.
* **Automation for Efficiency:** Automates time-consuming tasks like domain resolution and IP geolocation, freeing up valuable analyst resources within an SME. Analysts can then focus on higher-level tasks like threat assessment, investigation, and response.
* **User-Friendly Interface:** Built with Bootstrap for an intuitive design, Yeti caters to users with varying technical expertise. This reduces the need for extensive training within an SME, making it easier to leverage the platform even with limited technical staff.
* **Focus on Analysis:** Minimizes manual data formatting through a machine-friendly web API. This allows analysts, even with limited experience, to focus on strategic threat assessment and response, maximizing their impact.
* **Cost-Effective Solution:** Freely available open-source platform ensures affordability for budget-conscious SMEs.

**Performance in Resource-Constrained Environments:**

* **Reduced Processing Requirements:** By leveraging automation and a web-based interface, Yeti minimizes the need for powerful hardware or complex software installations, making it suitable for resource-constrained environments. [2]
* **Scalability:** Yeti can potentially scale to meet the growing needs of an SME as its resource capabilities evolve.

**Performance Metrics:** (by Misa)

Assessing performance metrics, especially in resource-constrained environments, for YETI requires further investigation to provide accurate insights. Performance metrics typically include factors such as processing speed, memory usage, and scalability under varying workloads. Conducting performance testing in different environments and configurations would help determine YETI's efficiency and effectiveness in resource-constrained settings. Additionally, evaluating factors like response time, data ingestion rate, and system stability would provide valuable insights into YETI's performance.

## **Accuracy and Reliability:** (by Mitchell)

* Yeti is at its most effective when integrated with other software. Elastic Stack is a great example of software that can be combined with Yeti to achieve more accurate and reliable results.
* Yeti collects feeds from various sources, including the Hybrid Analysis Public Feed, a publicly accessible sandbox service utilized for analyzing the execution of suspicious files. This feed offers valuable insights, presenting data through pie charts that precisely depict the ratio between threat levels observed in Hybrid Analysis and the corresponding detection rates on VirusTotal. By presenting this information visually, users can effectively assess the severity of threats, categorizing them along a scale ranging from benign or unknown to suspicious or outright malicious. [10]
* Furthermore, utilizing Yeti this way is a great start for users to find unknown threats and APTs in a relatively short amount of time.
* Yeti empowers users to uncover threat-hosting domains by aggregating feeds containing domains associated with suspicious or malicious activities. Leveraging Yeti's indexed data, users can easily access the top 10 domains linked to malware downloads. Additionally, users can verify the current status of these domains to determine if they remain active threats. [10]
* These queries can allow users to focus on malicious domains during investigations and track active campaigns using the ‘online’ status. [10]

# Usability and Accessibility - (by Keerthi)

## **Interface Design:**

* Partially Suitable for Non-Technical Users: Yeti offers a user-friendly interface built on Bootstrap, which is generally considered easy to navigate. However, some features might require familiarity with cybersecurity concepts like observables, TTPs, and malware. [6,7]

## **Documentation and Support:**

* Clarity: The current information available about Yeti seems generally clear. The Yeti documentation: yeti-platform.io explains core functionalities and should be helpful for those with some technical background. [4]
* Comprehensiveness: While the documentation covers core functionalities, it might not be sufficient for users entirely new to cybersecurity concepts. Additional resources or tutorials specifically geared towards non-technical users could be beneficial. [6]

## **Ease of Use:**

Learning Curve: There's likely a learning curve for users unfamiliar with threat intelligence terminology. However, the platform offers functionalities that can be valuable even with a basic understanding: [2,4,8]

* Bulk searching observables: This can help identify potential threats without deep knowledge of specific indicators.
* Threat analysis for identified threats: Once a threat is flagged, the platform can provide associated details like TTPs and malware samples, giving a clearer picture without needing in-depth technical understanding of each detail.

While Yeti might require some familiarization for non-technical users, its functionalities can still be valuable for identifying and understanding potential threats. Here are some suggestions for SMEs/developing economies: [5]

* **Prioritize Training:** To Invest in basic cybersecurity training for users to get acquainted with core concepts like observables and TTPs. This will significantly improve their ability to leverage Yeti's features.
* **Consider Alternative Interfaces:** To Explore if there are simpler front-end interfaces built on top of Yeti that cater to users with lower technical expertise. These interfaces could provide a more user-friendly way to interact with the platform's core functionalities.
* **Community Support:** To Look for online communities or forums where Yeti users can share knowledge and troubleshoot issues.
* **Improved Documentation:** The Yeti community can contribute to improving the existing documentation by creating resources specifically geared towards non-technical users. This could include tutorials, explanations of key concepts in simpler terms, and step-by-step guides for using core functionalities.

The fact that Yeti is free and available on GitHub makes it accessible for organizations with limited budgets. While Yeti has a user-friendly interface, there is a learning curve for non-technical users. By implementing the recommendations above, SMEs in developing economies can bridge this gap and leverage Yeti's power to gain valuable threat intelligence, ultimately improving their overall cybersecurity posture.

# Integration, Compatibility, and Scalability - (by Misa)

## **Interoperability with Existing Tools:**

Yeti is engineered to seamlessly integrate ensuring compatibility with a wide range of tools and systems commonly used by SMEs. Leveraging Yeti's interoperability capabilities, we can integrate it with antivirus solutions, SIEM platforms, threat intelligence feeds, and more, without disrupting your current operations.

YETI excels with its API, which can be seamlessly integrated into various programming languages. However, due to its Python-centric design, integration is particularly straightforward in Python environments. This inherent compatibility with Python facilitates smoother integration processes, although YETI's API can still be utilized in other languages. This flexibility ensures that organizations can leverage YETI's capabilities within their existing tech stacks, with Python being the preferred language for optimal integration.

## **APIs and Customization:**

* + *Auto-generated Documentation:*

Yeti offers a fully capable REST API, and the auto-generated documentation is available at: http://<YETI\_HOSTNAME>/docs where hostname is the host on which application is running.

* + *Authentication Process:*

Each user in Yeti is provided with an API key, which serves as a refresh token for obtaining a JWT access token.

* + - Making authenticated requests to the API involves a three-step process:
      1. Obtain an API key for the user you want to authenticate as.
      2. Send a POST request to /api/v2/auth/api-token with the API key in the x-yeti-apikey header.
      3. Extract the access token from the response and reuse it in the Authorization header of subsequent requests.
    - Example Python Code for Authentication:

import requests

apikey = "5902c3f2e63a172e0da2a8e9162771b3c7e0d98b813804f44149c1cd15dbcc6e" #This is a sample api key and will be different for each user.

# Obtain access token

response = requests.post(

"http://localhost:8000/api/v2/auth/api-token",

headers={"x-yeti-apikey": apikey},

)

access\_token = response.json().get("access\_token")

# Send authenticated requests

response = requests.get(

"http://localhost:8000/api/v2/auth/me",

headers={"Authorization": f"Bearer {access\_token}"},

)

print("response:", response.json())

# Using requests Session object for easier management

yeti\_session = requests.Session()

yeti\_session.headers.update({"Authorization": f"Bearer {access\_token}"})

response = yeti\_session.get("http://localhost:8000/api/v2/auth/me")

print("response:", response.json())

* + - How to Use Yeti API:

Refer to the auto-generated documentation provided at the above URL for detailed information on available API endpoints, request parameters, and response formats.

Utilize the provided Python code snippet as a reference for authenticating requests to the Yeti API and accessing resources. Adjust the API endpoint URLs, request headers, and payload data as needed for your specific use case.

## **Scalability:**

* + Containerization and cloud support enable YETI's scalability, allowing for effective deployment and scaling across distributed environments. Components can scale independently because to its modular architecture, and database optimization techniques improve efficiency. Yet, YETI's scalability can change based on system settings and workload. It takes extensive performance testing to find possible bottlenecks and maximize scalability. Additional testing is necessary to ascertain YETI's precise scalability in real-world deployment circumstances.

## **Flexibility:**

* Yeti offers unparalleled flexibility, enabling organizations to tailor its capabilities to their unique needs and workflows. Whether you're configuring data ingestion pipelines, defining custom indicators, or designing analysis workflows, Yeti provides the flexibility needed to adapt to evolving cybersecurity challenges. Below is an example of configuring custom indicators in Yeti.

**import requests**

**# Base URL for Yeti API**

**yeti\_api\_url = "http://localhost:8000/api/v2"**

**# Example API key for authentication**

**apikey = "your\_yeti\_api\_key"**

**# Example custom indicator data**

**custom\_indicator\_data = {**

**"type": "IP",**

**"value": "192.168.1.100",**

**"description": "Example custom indicator",**

**# Add more relevant data as needed**

**}**

**# Create custom indicator in Yeti**

**response = requests.post(**

**f"{yeti\_api\_url}/indicators",**

**headers={"Authorization": f"Bearer {apikey}"},**

**json=custom\_indicator\_data,**

**)**

**if response.status\_code == 201:**

**print("Custom indicator created successfully.")**

**else:**

**print("Failed to create custom indicator.")**

# Community Support and Sustainability – (by Keerthi)

## **Developer Community:**

* Yeti has an active developer community on GitHub. The platform is open source, which means that contributors from around the world can collaborate, report issues, and contribute code. [4]
* Community support is crucial, especially for small and medium-sized enterprises (SMEs) without in-house expertise. The Yeti community provides assistance, shares knowledge, and helps troubleshoot issues.

## **Updates and Maintenance:**

* Yeti aims to provide regular updates to enhance its features, address security vulnerabilities, and improve performance. [6]
* Regular maintenance ensures that the platform remains reliable and secure over time. Users can expect bug fixes, performance improvements, and new features through updates.

## **Local and Global Support Networks:**

**Local Support**:

* Yeti’s community extends globally, but local support can vary based on geographical regions. [1]
* Users can connect with other Yeti practitioners through local cybersecurity meetups, conferences, and workshops.
* Local user groups may organize events where users can share experiences, best practices, and tips related to Yeti.

**Global Support**:

* Yeti’s online presence allows users to access global support networks.
* The official Yeti website provides documentation, guides, and resources for users worldwide. [5]
* Users can participate in online forums, mailing lists, and social media groups to engage with the broader Yeti community.

# Assessment of Gaps – (by Mitchell)

## **Identified Gaps:**

* Usability and user experience.
* Integration with Additional Tools and Platforms solutions, and threat hunting platforms.
* Enhanced Automation and Orchestration.
* Advanced Analytics and Machine Learning.
* Community Collaboration and Data Sharing.
* Scalability and Performance.
* Security and Privacy Considerations.

## **Potential for Improvement:**

* While Yeti offers powerful features, ensuring that the platform is intuitive and user-friendly is crucial. Improvements in the user interface, workflow optimization, and documentation can enhance usability and make it easier for analysts to leverage the platform effectively.
* While Yeti supports integration with security information and event management, and other security tools, expanding its compatibility with a broader range of tools and platforms could further enhance its value. This includes integration with popular threat intelligence feeds, endpoint detection and response solutions, and threat hunting platforms.
* While Yeti provides an API for automation, further enhancing automation capabilities and orchestration features can streamline cyber threat intelligence workflows even more. This could involve automating repetitive tasks, orchestrating complex investigation and response processes, and integrating with orchestration platforms like Security Orchestration, Automation, and Response solutions.
* Incorporating advanced analytics techniques, such as machine learning and artificial intelligence, can help Yeti better analyze and correlate large volumes of cyber threat intelligence data. This can improve threat detection accuracy, identify emerging threats, and provide more proactive intelligence to security teams.
* Encouraging community collaboration and data sharing among users can enrich the platform's intelligence and expand its coverage of threats. This could involve establishing a community-driven repository for sharing custom analytics, threat indicators, and best practices.
* As organizations deal with increasingly large and complex datasets, ensuring that Yeti can scale effectively and maintain performance is essential. Continuously optimizing the platform's architecture and resource utilization can help address scalability challenges and ensure responsive performance.
* Given the sensitive nature of cyber threat intelligence data, prioritizing security and privacy considerations is critical. This includes implementing robust access controls, encryption mechanisms, and compliance with relevant privacy regulations to protect sensitive information stored and processed within the platform.

# Use Cases and Practical Applications

**Relevant Scenarios for SMEs/developing economies:** (by Mitchell)

* Cyber Threat Detection and Response - SMEs often lack dedicated cybersecurity teams and resources, making them vulnerable to cyber threats. Yeti can assist SMEs by providing automated threat detection and response capabilities. By integrating Yeti with their existing security infrastructure, SMEs can detect and respond to threats more effectively, reducing the risk of data breaches and financial losses. [5]
* Malware Analysis - SMEs may encounter malware infections that can disrupt business operations and compromise sensitive data. Yeti's malware analysis capabilities enable SMEs to analyze and remediate malware infections quickly. By leveraging Yeti's data and analytics, SMEs can identify the source of malware infections, remove malicious files, and implement measures to prevent future attacks. [5]
* Enhancing National Cybersecurity - Developing economies often lack robust cybersecurity infrastructure and capabilities, making them vulnerable to cyber threats. Yeti can support national cybersecurity efforts by providing threat intelligence sharing capabilities, incident response coordination, and cyber threat analytics. By leveraging Yeti's capabilities, developing economies can strengthen their cybersecurity posture and protect critical infrastructure. [5]
* Promoting Economic Development - Developing economies increasingly rely on digital technologies for economic growth and innovation. Yeti can support economic development initiatives by helping SMEs and startups in developing economies improve their cybersecurity posture. By providing access to threat intelligence, malware analysis, and incident response capabilities, Yeti can empower SMEs to protect their digital assets, attract investment, and contribute to economic development. [5]

## **Examples of Successful Use:** (by misa)

*Threat Hunting Case Study: YETI and Elastic Stack Integration [10]*

* Background: Security researchers and threat hunters often face challenges in identifying and mitigating emerging threats effectively. Leveraging YETI's capabilities alongside Elastic Stack, researchers can streamline threat hunting processes and uncover previously undetected malware and malicious activities.
* Integration with Elastic Stack: To enhance the analysis capabilities of YETI, researchers integrated its data with Elastic Stack, known for its scalability and data visualization features. By migrating YETI's data into Elastic Stack using a custom integration script, researchers gained the ability to perform advanced correlations, visualize relationships, and draw actionable insights from the threat data.
* Threat Hunting Visualization: Utilizing the combined power of YETI and Elastic Stack, researchers conducted threat hunting activities to identify zero-detection malware samples and investigate potential threats. By leveraging Elastic Stack's visualization tools, such as Kibana, researchers tracked the prevalence of malware variants like Gozi over time, correlating spikes in activity with external events and indicators.
* Campaign Tracking: Through comprehensive analysis using YETI's indexed data and Elastic Stack's visualization capabilities, researchers tracked the prevalence of Gozi malware variants, identifying significant spikes in activity corresponding to specific campaigns. Cross-referencing with external sources, such as Google Trends and threat intelligence reports, researchers gained insights into the timing and context of these campaigns, enabling proactive threat mitigation strategies.
* Identifying Unknown Threats: By leveraging YETI's integration with the Hybrid Analysis Public Feed, researchers identified previously unknown threats with low detection rates on VirusTotal. Through targeted queries and analysis, researchers uncovered new malware samples and potential attack vectors, enabling proactive threat detection and response.

# Conclusion and Recommendations for Development - (by Keerthi)

## **Overall Suitability:**

* + While Yeti, as a threat intelligence platform, offers promising features such as centralized data management, automation, and a user-friendly interface, it may not be the ideal fit for our project, Deakin Threat Mirror. Several factors contribute to this assessment. Firstly, Yeti has a relatively small community with only 50 contributors, which indicates limited ongoing development and support. Secondly, our testing revealed that the platform is still not fully developed and has encountered errors, suggesting potential stability issues. Lastly, being an open-source solution, Yeti lacks comprehensive technical support and documentation, which could pose challenges for integration and troubleshooting.

## **Recommendations for New Platform:**

* + In addition to the outlined features and capabilities, it's crucial to consider alternative platforms that may better suit the needs of our project, Deakin Threat Mirror. Based on our research and evaluation, MISP (Malware Information Sharing Platform & Threat Sharing) emerges as a superior alternative to Yeti.
  + MISP offers several advantages over Yeti, including a larger and more established community of contributors and users. With its development dating back to 2011, MISP has undergone extensive refinement and has a proven track record in the cybersecurity community. Its maturity and robustness make it a reliable choice for SMEs seeking a comprehensive threat intelligence platform.
  + Furthermore, Intel MQ presents another viable alternative worth considering. Like MISP, Intel MQ boasts a larger community and a longer development history compared to Yeti. Its focus on message queueing for threat intelligence sharing provides SMEs with a scalable and efficient solution for managing and disseminating threat data.

# Reference

[1] S. Handa, "YETI: A Threat Intelligence Platform," Help Net Security, Oct. 12, 2023. [Online]. Available: <https://www.helpnetsecurity.com/2023/10/12/yeti-threat-intelligence/>. [Accessed: Apr. 22, 2024].

[2] "MISP vs. YETI," Cosive, [Online]. Available: <https://www.cosive.com/misp-vs-yeti>. [Accessed: Apr. 22, 2024].

[3] "YETI Analytics Plugins," SecurityTrails Blog, [Online]. Available: <https://securitytrails.com/blog/yeti-analytics-plugins>. [Accessed: Apr. 22, 2024].

[4] "Best Open Source Threat Intelligence Platforms and Feeds," ZenArmor, [Online]. Available: <https://www.zenarmor.com/docs/network-security-tutorials/best-open-source-threat-intelligence-platforms-and-feeds>. [Accessed: Apr. 22, 2024].

[5] "YETI Platform," [Online]. Available: <https://yeti-platform.io/>. [Accessed: Apr. 22, 2024].

[6] "YETI: A Threat Intelligence Platform," Security Online, [Online]. Available: <https://securityonline.info/yeti-threat-intelligence/>. [Accessed: Apr. 22, 2024].

[7] T. Chantzios, P. Koloveas, S. Skiadopoulos, N. Kolokotronis, C. Tryfonopoulos, V.G. Bilali, and D. Kavallieros, "The Quest for the Appropriate Cyber-threat Intelligence Sharing Platform," in DATA, 2019, pp. 369-376.

[8] M. Garralaga Alvarez, "Detect and visualize vulnerable systems using active vulnerability scans," Bachelor's thesis, Universitat Politècnica de Catalunya, 2019.

[9] H. J. Hadi, M. A. Riaz, Z. Abbas, and K. U. Nisa, "Cyber Threat Intelligence Model: An Evaluation of Taxonomies and Sharing Platforms," in Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence, River Publishers, 2023, pp. 3-33.

[10] C. Erlich “[Threat Hunting using YETI and Elastic Stack](https://medium.com/@chenerlich/threat-hunting-using-yeti-and-elastic-stack-29735c51195f)” , Medium, 2020 [Accessed: Apr. 22, 2024].