

A Trellix® Project Report on

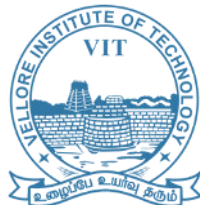
ePO Policy Comparison Tool: Streamlining Policy Analysis and Threat Detection for Enhanced Security Management

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology *in* Computer Science and Engineering

by

**Sharadindu “Sharad” Adhikari
19BCE2105**



VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

May, 2023

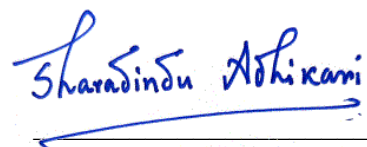
DECLARATION

I hereby declare that the thesis entitled “ePO Policy Comparison Tool: Streamlining Policy Analysis and Threat Detection for Enhanced Security Management” submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering to VIT, is a record of bonafide work carried out by me, under the supervision of Mr. Ashish Jain.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: May 30, 2023



Signature of the Candidate



We bring security to life.

+91 80 6656 9000 Main
+91 80 6656 9098 Fax

Registered office:

Musarubra Software India Private Limited
Embassy Golf Link Business Park, Pine Valley,
2nd Floor, Off Indiranagar Kormangala Int Ring Road
Bangalore, Karnataka, India - 560071
CIN: U72900KA2021FTC146803

30-May-2023

TO WHOM IT MAY CONCERN

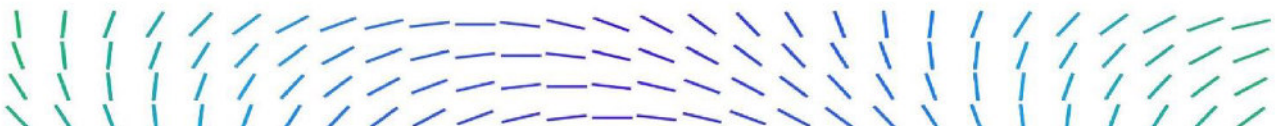
This is to certify that Mr. Sharad Adhikari (50000760) is a full-time employee (40hours/week) of Musarubra Software India Private Limited (Trellix) since 09-Jan-2023 and His current title is Graduate Intern Technical.

Please be advised that this reference is given in the strictest confidence and that the information cannot be used for anything besides the verification process.

And whilst every effort has been made to ensure that the above details are correct, the information is given on the strict basis that the company will accept no financial or other liability which may arise from the use of or reliance upon the information provided in this letter.

Regards,

Mahipal Nair
VP, Human Resources
For and on behalf of Musarubra Software India Private Limited



ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to Mr. Ashish Jain, my mentor, and Mr. Abhijit Bindage, my Manager, at Team DLP, Trellix Bengaluru. Their constant guidance, continual encouragement, and understanding have been invaluable throughout my internship. I am especially grateful for their teachings in patience, which have greatly contributed to my professional growth. Working with them has not only been a valuable learning experience but also a privilege to collaborate with intellectual and experienced professionals in the field of computer science.

I would also like to extend my gratitude to Dr. G Viswanathan, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. Rambabu Kodali, and Dr. Ramesh Babu K from the School of Computer Science and Engineering. Their provision of a conducive work environment and their inspirational support during my academic tenure have been instrumental in shaping my learning journey.

I am jubilant in expressing my heartfelt thanks to Dr. Vairamuthu S, Sr. Associate Professor and HOD Scope, as well as the entire teaching staff and university members who have played pivotal roles in my educational pursuit. Their enthusiasm, coupled with timely encouragement, has motivated and empowered me to acquire the necessary knowledge for the successful completion of my studies. I am also grateful to my parents for their unwavering support.

I am indebted to my friends who have consistently persuaded and encouraged me to undertake and accomplish this task. Finally, I would like to express my gratitude and appreciation to all those who have provided direct or indirect assistance, contributing to the successful completion of this project.

Place: Vellore

Date: May 30, 2023

Sharadindu Adhikari

EXECUTIVE SUMMARY

During my internship at Trellix, I worked on a project that involved developing an ePolicy Orchestrator (ePO) Policy Comparison tool. The goal of this tool was to automate the conversion of .json output files into easily readable formats and present the differences in a tabulated manner for users to analyze.

The project utilized Java, Spring Boot, and Maven for the backend, while the frontend was implemented using Angular. This technology stack provided a robust and scalable foundation for the development of the ePO Policy Comparison tool. The key functionalities of the tool included converting complex .json output files into layman-readable formats, comparing policies, and highlighting the differences for users to identify. This streamlined the policy comparison process and made it more accessible for non-technical users.

To ensure efficient development and deployment processes, several tools and technologies were integrated into the project. This included utilizing Artifactory for artifact management and TeamCity for continuous integration and deployment (CI/CD) pipelines. These tools enhanced collaboration, version control, and automated the build and deployment processes.

Additionally, the tool incorporated various threat levels of endpoint security detection. It classified threats based on their severity and provided administrators with clear visibility into the threat landscape. The system would ping and alert the responsible administrator in control of the ePO whenever a significant threat was detected.

Overall, the ePolicy Orchestrator Policy Comparison tool developed during my internship at Trellix demonstrated my proficiency in Java, Spring Boot, Maven, and Angular. Everything, from the brainstorming to developing the tool has been highly explained throughout this report, in addition to the additional features that have been added afterwards, and how ePO bridges the entire suit of products of Trellix. The tool has been developed into a very user-friendly interface for comparing policies, which converts .json files into easily understandable formats, and enhances the overall security management process by highlighting and classifying different threat levels. The integration of tools such as Artifactory and TeamCity facilitated efficient development and deployment, ensuring a smooth workflow throughout the project.

INDEX

Declaration	<i>ii</i>
Internship Certificate	<i>iii</i>
Acknowledgements	<i>iv</i>
Executive Summary	<i>v</i>
Table of Contents (Index)	<i>vi</i>
List of Figures	<i>viii</i>
List of Tables	<i>ix</i>
List of Acronyms (Abbreviations)	<i>x</i>
Symbols & Notations	<i>xi</i>
 Chapter 1. INTRODUCTION & OVERVIEW	 1
1.1. About Trellix as a Company	1
1.2. Core Values	2
1.3. Lines of Businesses (LOBs)	2
1.4. Domain of Work (Business Unit)	3
1.5. Key Features of DLP®	4
1.6. How Trellix DLP® Works	4
1.7. How Trellix DLP® Endpoint & Device Control Protect Sensitive Content	6
 Chapter 2. BACKGROUND	 9
2.1. Workflow for protecting sensitive data with Trellix DLP®	9
2.1.1. Classifying the data	10
2.1.2. Tracking how and when sensitive content is used	11
2.1.2.1. Content fingerprinting	11
2.1.2.1.1. Support for persistent fingerprinting information	11
2.1.2.2. Registered documents	12
2.1.2.3. Manual Classification	12
2.1.3. Protecting sensitive data with rules and policies	13
2.1.3.1. Rules and rule sets	13
2.1.3.2. Policies	14
2.1.3.3. Reviewing & managing incidents to fine-tune the policies	14
2.2. How policy components make up a policy	15
2.3. Two Important Use Cases	16
2.3.1. Data loss prevention policy workflow to block email attachments with sensitive data	17
2.3.2. Add custom add-in to monitor sent emails in new Outlook for macOS	20
2.4. Objectives	20
 Chapter 3. TOOLS AND TECHNOLOGIES	 22
3.1. Introduction	22
3.2. Agile Development	23

3.3. The Angular Framework	26
3.4. TypeScript	27
3.5. Java	29
3.6. API Development with Spring Boot	31
3.7. Git	33
3.8. Maven	35
3.9. Artifactory	37
3.10. TeamCity	38
3.11. Confluence and Jira	39
3.12. RDP	40
Chapter 4: MY WORK, LEARNINGS, EXPERIMENTS, & RESULTS	42
4.1. My Work (& The Motivation Behind It)	42
4.1.1. Development Workflow	42
4.1.1.1. Front-end Development with Angular	42
4.1.1.2. Back-end Development with Spring Boot	42
4.1.1.3. Integration and Data Management	42
4.1.2. Best Practices	46
4.1.3. Summary	47
4.2. Incidents, events, and cases	47
4.2.1. Incidents and operational events	47
4.2.1.1. Logging events with Syslog	48
4.2.1.2. Stakeholders	48
4.2.2. Monitoring and reporting events	49
4.2.3. DLP Incident Manager/DLP Operations	49
4.2.3.1. User Information	50
4.2.3.2. How the Incident Manager works	50
4.2.3.2.1. DLP Incident Manager	51
4.2.3.3. Incident tasks/Operational Event tasks	52
Chapter 5. INDUSTRIAL EXPERIENCE	53
5.1. Prior Skillset	53
5.2. Self-Evaluation	54
Chapter 6. CONCLUSION & FUTURE WORK	56
6.1. Diagnostics	56
6.1.1. Diagnostic Tool	56
6.1.2. Tuning Policies	57
6.2. Conclusion	58
6.3. Scope for Future Work	59
APPENDICES	61
REFERENCES	64

LIST OF FIGURES

Fig. 1. Working of DLP Suite	6
Fig. 2. Protection of sensitive content (by DLP's suit of products)	7
Fig. 3. How network discovery works, and How DLP locates it	8
Fig. 4. The Trellix DLP Protection Process	9
Fig. 5. How policy components make up a policy	15
Fig. 6. Adding a new definition of an existing policy	18
Fig. 7. Logging into the RDP	43
Fig. 8. Trellix ePolicy Orchestrator (formerly, McAfee ePO)	44
Fig. 9. ePO Dashboard	44
Fig. 10. ePO Before (before the creation of Policy Comparison Mechanism)	45
Fig. 11. ePO After (after the creation of Policy Comparison Mechanism)	45
Fig. 12. The DLP Incident Manager	51
Fig. 13. The DLP Operations	52

LIST OF TABLES

Table 1. How different components of policies make up an assigned policy	16
Table 2. Operational Event (Incident) tasks	52
Table 3. Diagnostic Tools in ePO Policy Comparison Module	57
Table 4. Explained in layman terms, all the glossaries used in the report	63

LIST OF ACRONYMS (ABBREVIATIONS)

1. ePO	- ePolicy Orchestrator
2. CI/CD	- Continuous Integration/Continuous Deployment
3. JSON	- JavaScript Object Notation
4. API	- Application Programming Interface
5. UI	- User Interface
6. JVM	- Java Virtual Machine
7. IDE	- Integrated Development Environment
8. SQL	- Structured Query Language
9. REST	- Representational State Transfer
10. CRUD	- Create, Read, Update, Delete
11. LDAP	- Lightweight Directory Access Protocol
12. SCM	- Source Code Management
13. QA	- Quality Assurance
14. HTML	- Hypertext Markup Language
15. HTTPS	- Hypertext Transfer Protocol Secure
16. BPM	- Business Process Management
17. SDK	- Software Development Kit
18. SSL	- Secure Sockets Layer
19. TLS	- Transport Layer Security
20. LDAP	- Lightweight Directory Access Protocol
21. OS	- Operating System
22. CPU	- Central Processing Unit
23. RAM	- Random Access Memory
24. XSS	- Cross-Site Scripting
25. CSRF	- Cross-Site Request Forgery
26. SSO	- Single Sign-On
27. SAML	- Security Assertion Markup Language
28. WAF	- Web Application Firewall

SYMBOLS & NOTATIONS

1. -> - Represents an arrow indicating direction or flow.
2. {} - Curly brackets used to enclose blocks of code or define sets.
3. [] - Square brackets used for array indexing or to denote optional parameters.
4. () - Parentheses used for grouping expressions or function calls.
5. = - Equals sign used for assignment or comparison.
6. + - Plus sign used for addition or concatenation.
7. - - Minus sign used for subtraction or negation.
8. * - Asterisk used for multiplication or wildcard matching.
9. / - Forward slash used for division or directory paths.
10. % - Percent sign used for modulo or percentage calculation.
11. ! - Exclamation mark used for negation or logical NOT operation.
12. < - Less than sign used for comparison or HTML tags.
13. > - Greater than sign used for comparison or HTML tags.
14. & - Ampersand used for bitwise AND or URL encoding.
15. | - Vertical bar used for bitwise OR or logical OR operation.
16. # - Hash symbol used for comments or to denote a section in a document.
17. \$ - Dollar sign used to represent currency or variable in some programming languages.
18. @ - At symbol used for email addresses or social media mentions.
19. ~ - Tilde used for negation or as a prefix for home directory.
20. : - Colon used to denote a case label or separator in URLs.

Chapter 1.

INTRODUCTION & OVERVIEW

1.1. About Trellix as a Company

Trellix is a cybersecurity company that was formed in 2021 through the merger of McAfee Enterprise and FireEye. The company's mission is to "build resilient and confident organizations through living security." Trellix offers a wide range of cybersecurity products and services, including endpoint security, network security, cloud security, and threat intelligence. The company's customer base includes businesses of all sizes, from small businesses to large enterprises. Trellix's domain is cybersecurity, and its target market is businesses of all sizes. The company's products and services are designed to help businesses protect themselves from a wide range of cybersecurity threats, including malware, ransomware, and data breaches.

Trellix is a leading provider of cybersecurity solutions. The company's products and services are used by businesses of all sizes to protect themselves from a wide range of cybersecurity threats. Trellix is committed to providing its customers with the best possible security solutions, and the company is constantly innovating to stay ahead of the latest threats.

Trellix's products and services work by combining a variety of technologies, including machine learning, artificial intelligence, and threat intelligence. This allows the company to provide its customers with a comprehensive and unified view of their security posture. Trellix's products and services are also designed to be easy to use and manage, even for businesses with limited IT resources.

Here are some of Trellix's most popular products:

- **Trellix XDR** is a unified security platform that combines endpoint security, network security, cloud security, and threat intelligence. XDR provides a comprehensive view of an organization's security posture and helps to identify and respond to threats more quickly.
- **Trellix Endpoint Security** is a next-generation antivirus solution that uses machine learning and artificial intelligence to protect endpoints from a wide range of threats. Endpoint Security also includes features such as device control, application control, and data loss prevention.
- **Trellix Network Security** is a suite of network security products that protect networks from a variety of threats, including malware, ransomware, and data breaches. Network Security includes products such as firewalls, intrusion detection systems, and web application firewalls.
- **Trellix Cloud Security** is a suite of cloud security products that protect cloud-based workloads from a variety of threats, including malware, ransomware, and data breaches.

Cloud Security includes products such as cloud firewalls, intrusion detection systems, and cloud access security brokers.

- **Trellix Threat Intelligence** is a service that provides organizations with access to threat intelligence data from Trellix's global threat intelligence network. Threat Intelligence can be used to identify and respond to threats more quickly.

In addition to these products, Trellix also offers a variety of services, including consulting, training, and support. Trellix's services can help organizations to implement and manage their security solutions more effectively.

1.2. Core Values

The company's core values are:

- **Open:** Trellix is committed to being open with its customers, partners, and employees. The company believes that open communication is essential for building trust and collaboration.
- **Tenacious:** Trellix is relentless in its pursuit of excellence. The company is committed to continuous improvement and innovation.
- **Curious:** Trellix is always looking for new ways to improve its products and services. The company believes that curiosity is essential for staying ahead of the latest threats.
- **Fun:** Trellix is a fun and engaging place to work. The company believes that a positive work environment is essential for attracting and retaining top talent.

These core values guide Trellix in everything it does. The company is committed to providing its customers with the best possible security solutions, and it is constantly innovating to stay ahead of the latest threats.

1.3. Lines of Businesses (LOBs)

The company's lines of businesses include:

- **Endpoint security:** Trellix offers a variety of endpoint security products, including next-generation antivirus, endpoint detection and response (EDR), and application control. These products help to protect endpoints from malware, ransomware, and other threats.
- **Network security:** Trellix offers a variety of network security products, including firewalls, intrusion detection systems (IDS), and web application firewalls (WAFs).

These products help to protect networks from a variety of threats, including malware, ransomware, and data breaches.

- **Cloud security:** Trellix offers a variety of cloud security products, including cloud firewalls, IDS, and WAFs. These products help to protect cloud-based workloads from a variety of threats, including malware, ransomware, and data breaches.
- **Threat intelligence:** Trellix offers a variety of threat intelligence products, including threat feeds, threat analysis, and threat hunting. These products help organizations to identify and respond to threats more quickly.
- **Managed security services:** Trellix offers a variety of managed security services, including security consulting, security operations, and security education. These services can help organizations to implement and manage their security solutions more effectively.

Trellix's products and services are used by businesses of all sizes to protect themselves from a wide range of cybersecurity threats. The company is committed to providing its customers with the best possible security solutions, and it is constantly innovating to stay ahead of the latest threats.

1.4. Domain of Work (Business Unit)

I work on a team that develops and maintains Trellix's data loss prevention (DLP) products. DLP products are designed to help organizations protect sensitive data from unauthorized access, disclosure, or destruction. My team uses a variety of technologies, including machine learning, artificial intelligence, and natural language processing, to develop DLP products that are effective at detecting and preventing data leaks.

We offer a suite of DLP products that can be used to protect data at rest, in motion, and in use. Our products can be deployed on-premises or in the cloud, and they can be used to protect data in a variety of environments, including email, web browsing, file sharing, and mobile devices.

Our DLP products are used by a wide range of organizations, including Fortune 500 companies, government agencies, and educational institutions. We are committed to providing our customers with the best possible DLP solutions, and we are constantly innovating to stay ahead of the latest threats.

Each of this Trellix DLP product protects different types of data in your network.

- **Trellix Data Loss Prevention Endpoint for Windows** — Content-based agent solution that inspects user actions. It scans data-in-use on endpoints and blocks or encrypts unauthorized transfer of data identified as sensitive or confidential. The

Endpoint Discovery feature scans local file system and email storage files and applies rules to protect sensitive content.

- **Trellix Data Loss Prevention Endpoint for Mac** — Offers similar protection for Macintosh computers running macOS operating systems.
- **Trellix Device Control** — Controls the use of removable media on endpoints. Trellix Device Control contains a subset of the protection rules in Trellix DLP Endpoint for Windows and Mac.
- **Trellix Data Loss Prevention Discover** — Scans network file, Box, SharePoint, and database repositories to identify and protect sensitive data by copying or moving the files, or by applying an RM policy. Registration scans extract fingerprint information from file repositories for file classification and store the signatures in a registered documents database.
- **Trellix Data Loss Prevention Prevent** — Works with your web proxy or MTA server to protect web and email traffic.
- **Trellix Data Loss Prevention Monitor** — Passively scans unencrypted network traffic for potential data loss incidents.

1.5. Key Features of DLP®

Trellix DLP provides comprehensive protection for all potential leaking channels, including removable storage devices, the cloud, email, instant messaging, web, printing, clipboard, screenshot, and file-sharing applications.

Compliance enforcement — Ensure compliance by addressing day-to-day user actions, such as emailing, cloud posting, and downloading to removable media devices.

Advanced protection — Apply fingerprinting, classification, and file tagging to secure sensitive, unstructured data, such as intellectual property and trade secrets.

Scanning and discovery — Scan files and databases stored on local endpoints, shared repositories, or the cloud to identify sensitive data.

User education — Provide real-time feedback through educational pop-up messages to help shape corporate security awareness and culture.

Centralized management — Integrate with **Trellix ePO - On-prem** software to streamline policy and incident management.

1.6. How Trellix DLP® Works

Trellix DLP products identify sensitive data or user activity, take action on policy violations, and create incidents of violations.

Installing all Trellix DLP products allows you to use the full feature set of the product suite. The following diagram shows a simplified network where all Trellix DLP products and Trellix ePO - On-prem are deployed.

1. Administrators create policies in Trellix ePO - On-prem and deploy them to Trellix DLP Endpoint for Windows and Trellix DLP Endpoint for Mac clients.

- a. Users create, save, and copy files or emails.
- b. Trellix DLP Endpoint client applies policies and either blocks or allows user actions.
- c. Applying the policies creates incidents that are sent to DLP Incident Manager for reporting and analysis.

2. Trellix DLP Discover scans files from local or cloud repositories and local databases, collecting file metadata.

- a. Trellix DLP Discover receives classifications and policies from Trellix DLP to apply during classification or remediation scans.
- b. DLP Server software creates registered documents databases for use in policies for Trellix DLP Discover, Trellix DLP Prevent, and Trellix DLP Monitor.
- c. Incidents from remediation scans are sent to DLP Incident Manager for reporting and analysis.

3. Trellix DLP Prevent receives email from MTA servers and web traffic from web proxy servers. It analyzes the email messages and web traffic, applies the Trellix DLP policies, and sends incidents and evidence to DLP Incident Manager.

4. Trellix DLP Monitor analyzes network traffic, then creates incidents or saves evidence for the supported protocols. It applies network communication protection rules, web protection rules, or email protection rules.

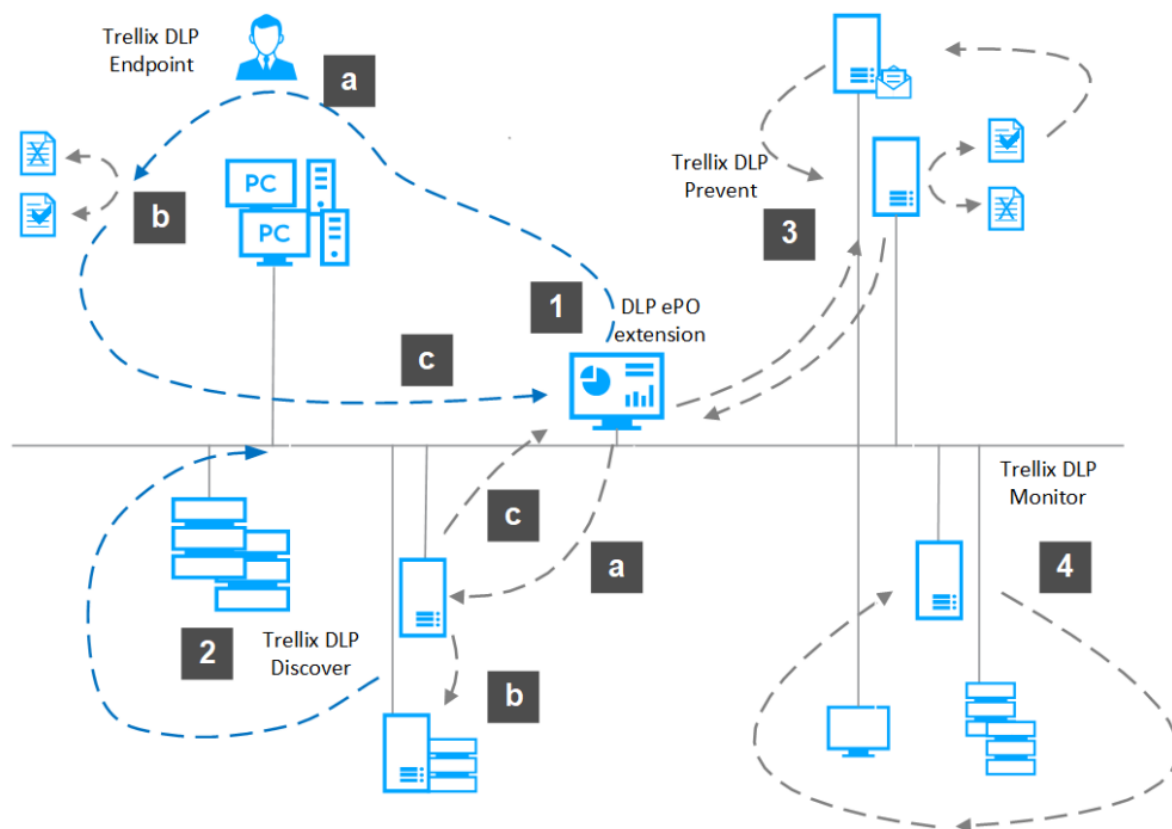


Fig.1. Working of DLP Suite

1.7. How Trellix DLP® Endpoint and Device Control protect sensitive content

Trellix Device Control controls sensitive content copied to removable devices. Trellix DLP Endpoint also inspects enterprise users' actions on sensitive content when emailing, using cloud applications, and posting to websites or network shares.

The Trellix DLP Endpoint client software is deployed as a Trellix Agent plug-in, and enforces the policies defined in the Trellix DLP policy. It audits user activities to monitor, control, and prevent unauthorized users from copying or transferring sensitive data and generates events recorded by the Trellix ePO - On-prem Event Parser.

Events generated by the Trellix DLP Endpoint client software are sent to the Trellix ePO - On-prem Event Parser, and recorded in tables in the Trellix ePO - On-prem database. Events are stored in the database for further analysis and used by other system components.

1. Create policies consisting of definitions, classifications, and rule sets (groups of Trellix Device Control, Data Protection, and Discovery rules) in the DLP Policy Manager and Classification consoles in Trellix ePO - On-prem.

2. Deploy the policies to the endpoints.
3. Collect incidents from the endpoints for monitoring and reporting.

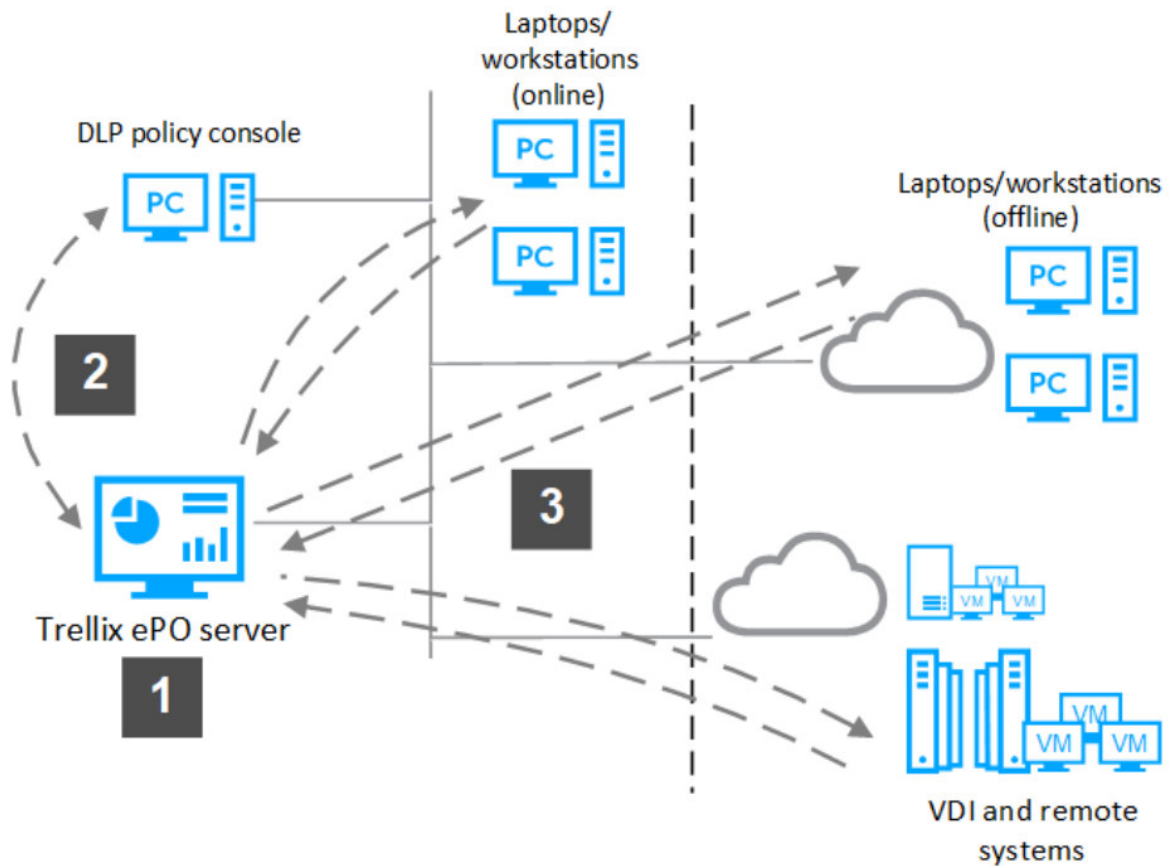


Fig.2. Protection of sensitive content (by DLP's suit of products)

Trellix DLP Endpoint for Windows safeguards sensitive enterprise information using four layers of protection:

1. Trellix Device Control rules control information copied to external drives.
2. Data protection rules control data as it is used or copied to files and emails.
3. Endpoint discovery scans local file and email repositories for sensitive information.
4. Web application control rules block specified URLs by name or by reputation.

Trellix DLP Endpoint for Mac safeguards sensitive enterprise information using three layers of protection:

1. Trellix Device Control rules control information copied to external drives.
2. Data protection rules control data as it is used or copied to files.
3. Endpoint discovery scans local file repositories for sensitive information.

Trellix DLP Endpoint safeguards sensitive enterprise information:

- Applies policies that consist of definitions, classifications, rule sets, endpoint client configurations, and endpoint discovery schedules.
- Monitors the policies and blocks actions on sensitive content, as needed.
- Encrypts sensitive content before allowing the action.
- Creates reports for review and control of the process, and can store sensitive content as evidence.

We can apply different device and protection rules, depending on whether the managed computer is online (connected to the enterprise network) or offline (disconnected from the network). Some rules also allow you to differentiate between computers within the network and those connected to the network by VPN.

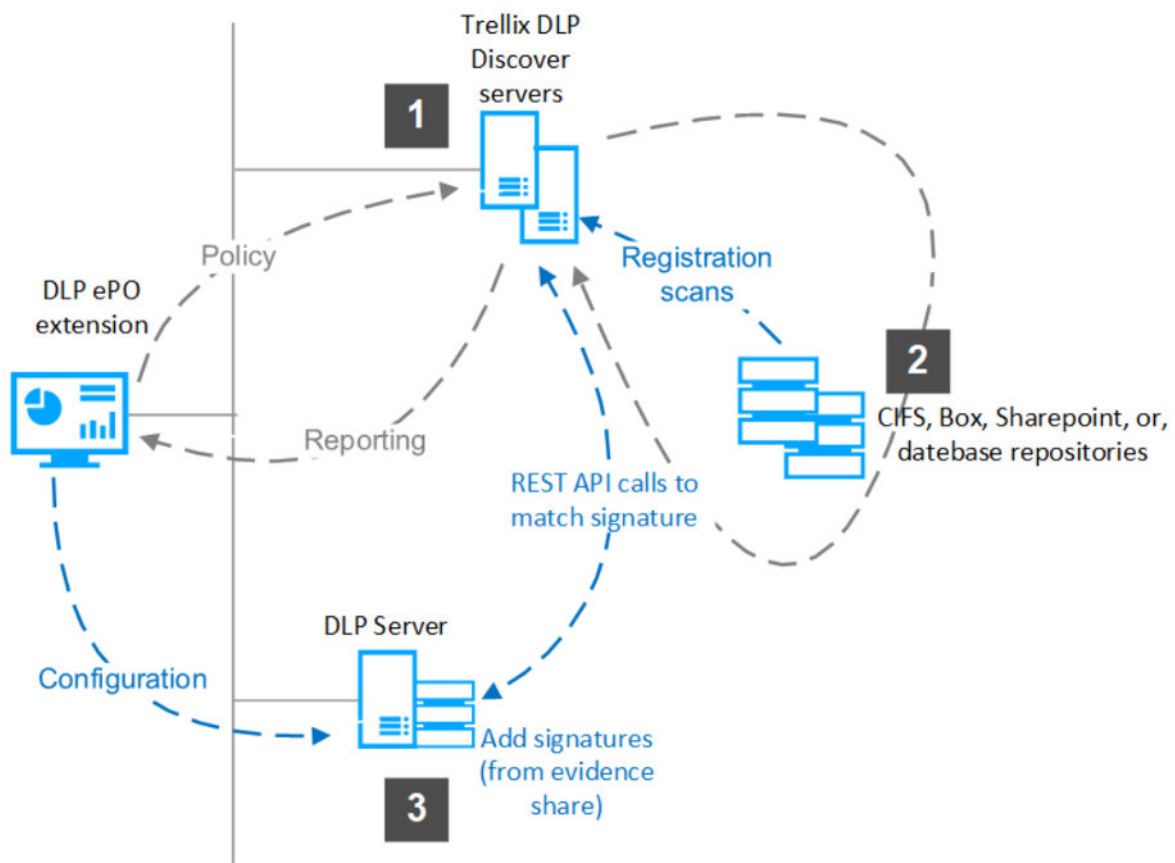


Fig.3. How network discovery works, and How DLP locates it.

Chapter 2. BACKGROUND

2.1. Workflow for protecting sensitive data with Trellix DLP

Trellix DLP features and policy components make up a protection process that fits into this overall workflow.

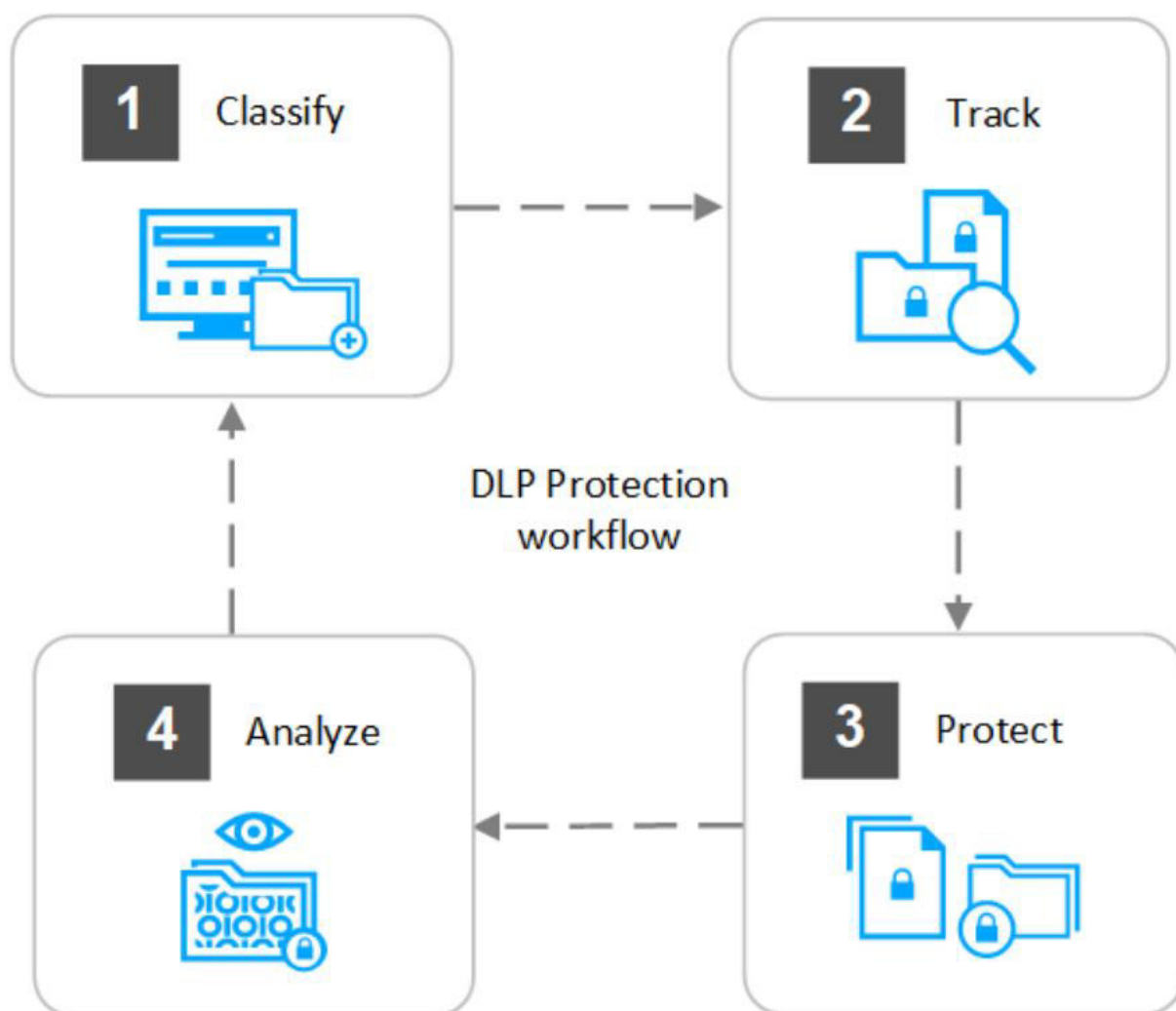


Fig.4. The Trellix DLP protection process

We use Trellix DLP to monitor the data and user actions on the network. We can use predefined rules or create a basic policy.

1. Create classifications from the Classification console. Classify and define sensitive data by configuring classifications and definitions. For more information, see [Classifying your data](#)
2. Track how and where the files containing sensitive content are used with tags or registered documents. For more information, see [Tracking how and when sensitive content is used](#).
3. Protect sensitive data by applying rules with DLP Policy Manager. Protect data with scans and rules. Configure the action to take when sensitive data is discovered, accessed, or transmitted. For more information, see [Protecting sensitive data with rules and policies](#).
4. Analyze the Trellix DLP incidents from DLP Incident Manager. Review incidents and analyze scan results to see potential policy violations. Use this information to begin creating an effective policy. You can manage the incidents by grouping and working with incidents, which can be escalated to other departments, such as legal or Human Resources. You can also create reports with dashboards and queries. For more information, see [Reviewing and managing incidents to fine-tune your policies](#).

2.1.1. Classifying the data

To protect sensitive content, start by defining and classifying sensitive information that needs to be protected.

Content is classified by defining classifications and classification criteria. Classification criteria defines the conditions on how data is classified. Methods to define criteria include:

- **Advanced patterns** — Regular expressions combined with validation algorithms, used to match patterns such as credit card numbers. Advanced patterns are ranked according to a score, meaning, the number of times the sensitive expressions need to appear in the content for the rule to be triggered. The Classifications editor includes several built-in advanced patterns for ensuring compliance with government regulations and simplifying detection of personal information. You can also create your own advanced patterns.
- **Dictionaries** — Lists of specific words or terms, such as medical terms for detecting possible HIPAA violations.
- **Keywords** — A string value that defines sensitive data. You can add multiple keywords for content classifications. Keywords are not consistent across classifications. If you need to use consistent keywords across classifications, use a dictionary.
- **File size** — The size of the file to detect the sensitive data. You can also define a file size range.
- **True file types** — The true file type to determine which files to identify the sensitive data. True file type helps detect attachment violations when file extensions are renamed

and sent as attachments. For example, a .cpp file saved as a .txt file can be detected using the true file type classification criteria.

- **File extension** — The file types to detect the sensitive data, such as MP3 and PDF. Source or destination location — URLs, network shares, or the application or user that created or received the content.
- **Location in file** — The section of the file to look for the sensitive content; Header, Footer, Body or within the first characters. Specifying the number of characters for the within first (characters) option in a classification looks for the sensitive content in the Header, that is, in the first part of the first page in a document.
- **Microsoft Word documents** - Header, body and footer is identified.
PowerPoint documents - WordArt is considered Header, everything else is identified as Body.
Other documents - Only Body is applicable.

2.1.2. Tracking how and when sensitive content is used

Trellix DLP can track content based on storage location or the application used to create it.

The mechanisms used to track content are:

- Content fingerprinting — Supported on Trellix DLP Endpoint for Windows and Trellix DLP appliances.
- Registered documents — Supported on all Trellix DLP products except Trellix DLP Endpoint for Mac.
- Manual classifications — Created by Trellix DLP Endpoint and Trellix DLP Endpoint for Mac users, but supported on all Trellix DLP products.

2.1.2.1. Content fingerprinting

Content fingerprinting is a technique for identifying and tracking content. The administrator creates a set of content fingerprinting criteria. The criteria define either the file location or the application used to access the file, and the classification to place on the files. The Trellix DLP Endpoint client tracks any file that is opened from the locations, or by the applications, defined in the content fingerprinting criteria and creates fingerprint signatures of these files in real time when the files are accessed. It then uses these signatures to track the files or fragments of the files. You can define content fingerprinting criteria by application, UNC path (location), or URL (web application).

2.1.2.1.1. Support for persistent fingerprint information

Content fingerprint signatures are stored in a file's extended file attributes (EA) or alternate data streams (ADS). When such files are accessed, Trellix DLP Endpoint software tracks data transformations and maintains the classification of the sensitive content persistently, regardless of how it is being used. For example, if you open a fingerprinted Word document, copy a few paragraphs of it into a text file, and attach the text file to an email message, the outgoing text file has the same signatures as the original document.

For file systems that do not support EA or ADS, Trellix DLP Endpoint software stores signature information as a metafile on the disk. The metafiles are stored in a hidden folder named ODB\$, which the Trellix DLP Endpoint client software creates automatically.

2.1.2.2. Registered documents

The registered documents feature is based on pre-scanning all files in specified repositories (such as the engineering SharePoint) and creating signatures of fragments of each file in these repositories. Trellix DLP Endpoint and the network Trellix DLP products use registered documents, but differ in the way the signatures of files are distributed.

Manual registration in Trellix DLP Endpoint for Windows — Signatures of the files are uploaded to Trellix ePO - On-prem from when you manually upload files and create a package. These signatures are made available and downloaded by the endpoints from the shared location. The Trellix DLP Endpoint client is then able to track any content copied from one of these documents and classify it according to the classification of the registered document signature.

Manual registration in Trellix DLP network products — Signatures of the files are uploaded to Trellix ePO - On-prem from Trellix DLP when you manually upload files and create a package. A package of these signatures of files is saved in an evidence share. These signatures are made available and downloaded by the appliances from the shared location. The appliance is then able to track any content copied from one of these documents and classify it according to the classification of the registered document signature.

Automatic registration in Trellix DLP network products — Trellix DLP Discover runs registration scans on file repositories. The signatures created are stored in signature databases on servers designated as DLP Servers. Trellix DLP Discover uses them to create classification and remediation scans. Trellix DLP Prevent and Trellix DLP Monitor use them to define rules.

Registered documents use extensive memory, which might affect performance, because each document that the Trellix DLP software inspects is compared to all registered document signatures to identify its origin.

2.1.2.3. Manual Classification

When working with manual classification, you have the option of applying content fingerprints or content classifications to files. Manually applied content fingerprinting is identical to the automatically applied fingerprinting described previously.

Manually applied content classifications embed a physical tag in the file which can be used to track the file wherever it is copied, but do not create signatures. Content copied from these files into other files can't be tracked.

Manual classification is supported on Microsoft Windows and macOS computers. If you try to classify a file type that doesn't support tagging (for example, TXT files), an error message displays.

2.1.3. Protecting sensitive data with rules and policies

Create rules to identify sensitive data and take appropriate action.

2.1.3.1. Rules and rule sets

Rules are made up of conditions, exceptions, and actions. Conditions contain multiple parameters — such as classifications — to define the data or user action to identify. Exceptions specify parameters to exclude from triggering the rule. Actions specify how the rule behaves when a rule is triggered, such as blocking user access, encrypting a file, and creating an incident. Rules are organized into rule sets. A rule set can contain any combination of rule types.

- **Data Protection rules** — Data protection rules are used to prevent unauthorized distribution of classified data. When you try to copy classified data, or attach it to an email, Trellix DLP intercepts the attempt and uses the data protection rules to determine which action to take. For example, if the rule action requires a business justification, Trellix DLP Endpoint halts the attempt and displays a dialog box. When the user inputs the justification for the attempt, processing continues.
 - Trellix DLP Endpoint uses several rules to inspect user actions. It scans data-in-use on endpoints and blocks unauthorized transfer of data identified as sensitive or confidential.
 - Trellix DLP Prevent uses web and email protection rules to monitor and take action on communication from an MTA server or web proxy server.
 - Trellix DLP Monitor can apply the network communication protection, email protection, or web protection rules to analyze supported traffic on your network.
 - Trellix Device Control uses only removable storage data protection rules.

- **Device Control rules** — Device Control rules monitor and potentially block the system from loading physical devices such as removable storage devices, Bluetooth, Wi-Fi, and other plug-and-play devices. Device Control rules consist of device templates and reaction specifications, and can be assigned to specific user groups by filtering the rule with user group definitions.
- **Application control rules** — Application control rules block the application rather than blocking the content. For example, a web application control rule blocks a specified URL by name or by reputation.
- **Discovery rules** — Discovery rules are used for file and data scanning. Endpoint Discovery is a crawler that runs on managed computers. It scans the local endpoint file system and the local email (cached) inbox and PST files. Local file system discovery rules define whether the content is to be quarantined, encrypted, content fingerprinted, or have an RM policy or classification applied. Local emails can be quarantined or content fingerprinted. These rules can also define whether an incident is reported, and whether to store the file or email as evidence included in the incident.

2.1.3.2. Policies

Policies contain active rule sets and are deployed from Trellix ePO - On-prem to the Trellix DLP Endpoint client software, Trellix DLP Discover server, Trellix DLP Prevent, or Trellix DLP Monitor. Trellix DLP Endpoint policies also contain policy assignment information and definitions.

2.1.3.3. Reviewing and managing incidents to fine-tune the policies

We can review, analyze, and manage incidents for policy violations that have occurred. These functions include:

- **Incident management** — Incidents are sent to the Trellix ePO - On-prem Event Parser and stored in a database. Incidents contain the details about the violation, and can optionally include evidence information. You can view incidents and evidence as they are received in the DLP Incident Manager console.
- **Case management** — Group-related incidents into cases for further review in the DLP Case Management console.
- **Evidence collection** — For rules that are configured to collect evidence, a copy of the data or file is saved and linked to the specific incident. This information can help determine the severity or exposure of the event. Evidence is encrypted using the AES-256 algorithm before being saved.
- **Hit highlighting** — Evidence can be saved with highlighting of the text that caused the incident. Highlighted evidence is stored as a separate encrypted HTML file.
- **Reports** — Reports, charts, and trends are created in Trellix ePO - On-prem dashboards.

2.2. How policy components make up a policy

Trellix DLP products use a similar workflow for creating policies. A policy consists of rules, grouped into rule sets. Rules use classifications and definitions to specify what Trellix DLP detects. Rule reactions determine the action to take when data matches the rule.

This is the workflow for creating and applying policies:

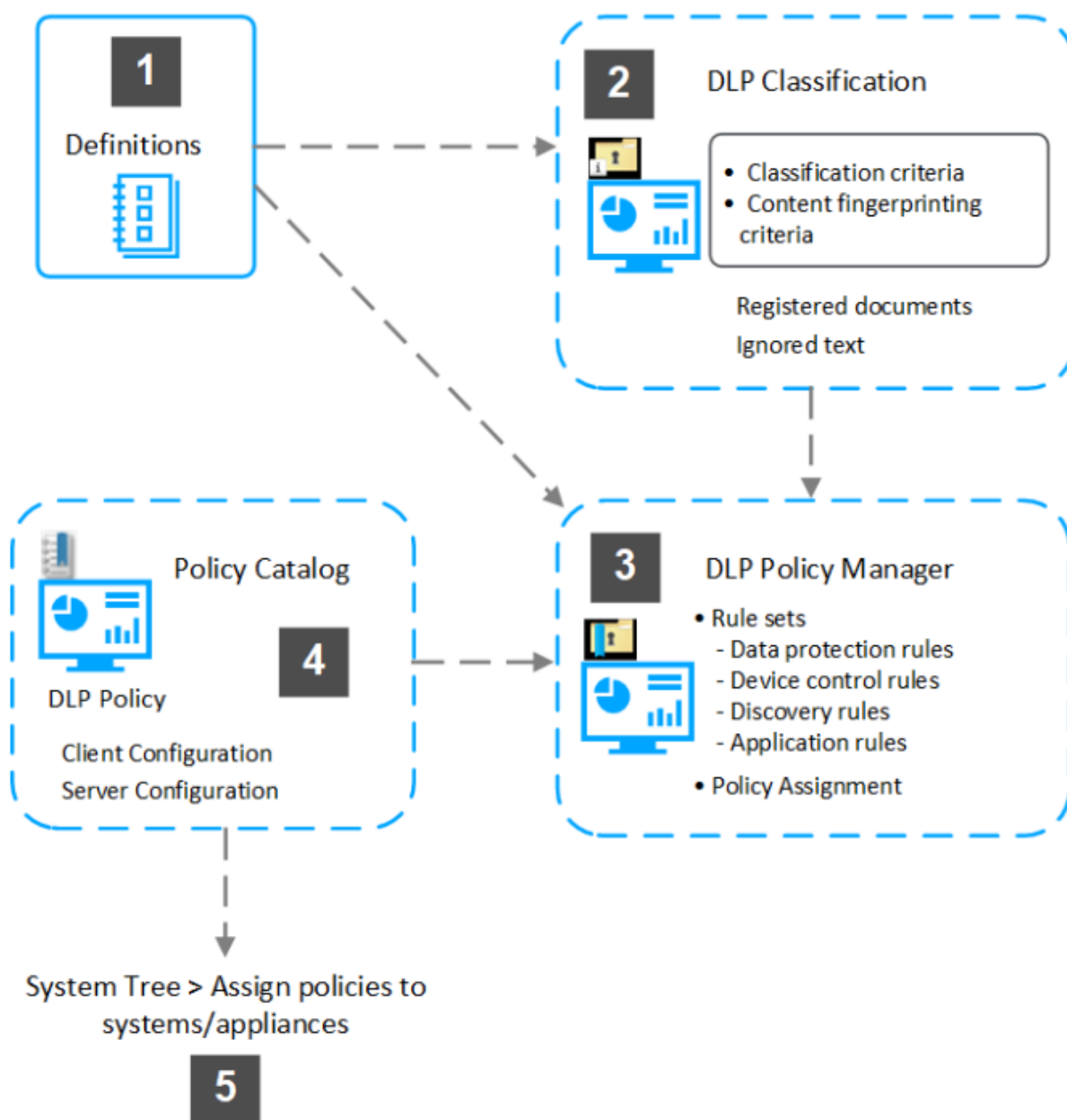


Fig.5. How policy components make up a policy

1	Create definitions — Used to create classifications and rules.
2	<p>Create a classification — Define categories of sensitive data by creating classifications. Classifications are used by rule sets to define the data protection rule.</p> <ul style="list-style-type: none"> • Content fingerprinting criteria and ignored text — Supported in Trellix DLP Endpoint for Windows only. • Registered documents — Created in the Classification console (manual registration) for Trellix DLP Endpoint for Windows clients and Trellix DLP appliances. • Classification criteria — Used to create definitions to identify sensitive text patterns, dictionaries, and keywords. Not supported with Trellix Device Control.
3	Create a rule set and Create a rule — Rule sets can combine multiple data protection rules for improved coverage of data protection. Create a rule and its actions and add it to a rule set.
4	Assign rule sets to policies — A policy can have multiple rule sets. Before you apply rule sets to a policy, activate the rule sets. Assign the rule sets to the policy and then apply the required rule sets to the policy
5	Assign and push a policy to a system — Trellix DLP policies are assigned to endpoints and Trellix DLP appliances from System Tree. You can use the Wake Up Agents option or the Break inheritance and assign the policy and settings below option to push a policy to a system.

Table.1. How different components of policies make up an assigned policy

The options and availability for these components vary depending on which Trellix DLP product the user use.

2.3. Two Important Use Cases

2.3.1. Data loss prevention policy workflow to block email attachments with sensitive data

This section describes all step-by-step tasks that you need to perform to create and apply a policy to block email attachments with sensitive data. Trellix DLP Prevent can use file extension or true file type in classification criteria to block email attachments with sensitive data. You can also use these classifications to block attachments sent in web posts or network.

Consider that you have some architecture and planning diagrams of your business solution in Autocad and Visio formats, which you have saved in a shared location. The administrator wants to block all Autocad or Visio attachments with sensitive data from going outside the network. You can use these steps to block attachments for these specific extensions and prevent any data loss.

1. Create a definition or choose a built-in definition. (See Step 1 in the following task)
2. Create a classification and classification criteria. (See Step 2 in the following task)
3. Create a rule set and rules. (See Step 3 in the following task)
4. Assign rule sets to policy. (See Step 4 in the following task)
5. Assign and push policy to system. (See Step 5 in the following task)

Task

1. Choose a built-in definition or create a definition to include phrases that must be tracked.
 - a. In Trellix ePO - On-prem, go to Menu → Data Protection → Classification and select Definitions.
 - b. Select Dictionary, click Action → New Item, give the dictionary definition a name and an optional description, then click Action → Add. You can also use an existing dictionary.
 - c. In Phrase, type the word internal, then set the Score as 1 and select Case Sensitive to only match on the keyword when it is lowercase. To add multiple phrases, you can click Save and New.
 - d. Click Save.

Add	
Phrase :	internal
Score (+/-) :	1
Start With :	<input checked="" type="checkbox"/>
End With:	<input checked="" type="checkbox"/>
Case Sensitive:	<input type="checkbox"/>

Save and New Save Cancel

Fig.6. Adding a new definition of existing policy

2. Create a classification and classification criteria. You can also edit an existing classification.
 - a. Select Classification.
 - b. Click Actions, then click New Classification and type a unique name and an optional description. Select Save Classification in the Actions menu.
 - c. In the right pane, click Actions, then select New Content Classification Criteria and type the classification criteria name.
 - d. In the Data conditions properties, click to select Dictionary and add the dictionary that you created.
 - e. In the File conditions properties, click to select File Extensions and True File Type. Classification criteria with true file type helps detect attachment violations when file extensions are renamed and sent as attachments. For example, a .cpp file saved as .txt file can be detected using the true file type classification criteria.
 - f. For the File Extensions property, click the select icon (...) to open the Choose from existing values window. Choose all file types that you want to block. To add more values, click +.
 - g. For the True File Type property, click the select icon (...) to open the Choose from existing values window. Choose all file types that you want to block. To add more values, click +.
 - h. Click Save.
3. Create a rule set that includes an Email Protection rule and add the classification criteria that you created.
 - a. Go to Menu → Data Protection → DLP Policy Manager.

b. Click Actions → New Rule Set. Enter a name for the rule set and provide a description for your reference. Click OK. A new rule set is created. Click the rule set and then create a new rule.

c. Click Actions → New Rule → Email Protection rule.

d. Type the rule name and optionally enter the description. Select the state as Enabled and click the checkbox to select Trellix Network DLP to enforce the policy on.

e. In the Conditions tab, in Classification of, select one of the attachments (*) and contains one of (OR), and then select the classification criteria you created.

f. Set the Recipient to any recipient (ALL).

g. In the Reactions tab, set the reaction you want to take when the rule gets triggered. Set the Action to Block and return email to sender, then select the appropriate User Notification. Click Save.

4. Assign rule sets to a policy. Before you assign rule sets to a policy, activate the rule set.

a. Go to Menu → Policy → Policy Catalog.

b. In the Product drop down list, select Data Loss Prevention <version> and select DLP Policy.

c. Click the Edit link of the policy you want to update.

d. In the policy page, go to Active Rule Sets → Actions, then click Activate Rule Set. The Activate Rule Set window opens.

e. Select the checkboxes of one or more rule sets that you want to apply to the policy.

f. Click OK and click Apply Policy. Trellix DLP displays the status of the policy applied.

5. Assign and push the policy to Trellix DLP appliances.

a. Go to Menu → Systems → System Tree.

b. Select the checkbox of one or more Trellix DLP Prevent appliances (target system) that you want to assign the policy to.

c. Click Wake Up Agents to push the policy to Trellix DLP appliances immediately. The Wake Up Trellix Agent window opens.

d. Next to Wake-up call type, select whether to send an Agent Wake-Up Call or a SuperAgent Wake-Up Call.

e. Accept the default Randomization (0–60 minutes) or type a different value. If you type 0, agents respond immediately.

f. Click OK to send the wake-up call to the target appliances.

Alternatively, you can use the Break inheritance and assign the policy and settings below option to push the policy. For information, see Assign and push a policy to a system.

Results

The appliance is now set with policy to block email attachments with sensitive data.

2.3.2. Add custom add-in to monitor sent emails in new Outlook for macOS

This section explains how to add a custom add-in to monitor the sent emails on your macOS.

Before we begin

- Install Trellix Data Loss Prevention
- Make sure that the Python HTTPS web server named outlook_monitor is always running.
- Set the self-signed certificate to Always Trust in the System Keychain using the Keychain Access application or by running the `sudo security add-trusted-cert -d -r trustRoot -k /Library/Keychains/System.keychain/usr/local/McAfee/DlpAgent/bin/outlook_monitor.bundle/localhost.pem` command in the command line interface.

For Admin-managed accounts, the custom add-in must be deployed using Microsoft Office 365 Admin Center or Microsoft Exchange Admin Center. You can learn more about deploying add-ins at [Deploy add-ins in the Microsoft 365 admin center](#).

To add the custom add-in to our managed account:

Task

1. In Microsoft Outlook, go to Menu → Tools → Get Add-ins.
2. On the left pane, click My add-ins → Custom add-ins → Add a custom add-in → Add from a file....
3. Browse to the file path `/usr/local/McAfee/DlpAgent/bin/outlook_monitor.bundle/dist/` .xml and click Install.

Results

The custom add-in monitors email headers, email bodies, and attachments for any sensitive content.

2.4. Objectives

The objective of this thesis report is to shed light on the extensive learnings, valuable experiences, and technical knowledge that I have gained during my time as an intern. It encompasses not only the comprehensive trainings I have undergone but also the significant contributions I have made as a dedicated team member in various ongoing company projects.

Through meticulous documentation, I aim to provide detailed descriptions of the work I have been involved in, presenting a systematic and organized account of my journey. This report serves as a testament to the skills I have acquired and honed throughout my internship, showcasing my growth and development in a professional setting.

As an intern, I have had the opportunity to immerse myself in diverse workflows, each presenting unique challenges and learning opportunities. By delineating these workflows, I aim to illustrate the depth and breadth of my involvement, demonstrating my ability to adapt and contribute effectively to the team's objectives. Additionally, this report serves as a reflection of the learning curve I have navigated, showcasing the progress I have made in terms of technical proficiency, problem-solving, and collaborative skills.

Furthermore, this thesis report goes beyond mere enumeration of tasks and accomplishments. It delves into the underlying methodologies, frameworks, and technologies I have encountered and utilized. By providing contextual information and insights into the rationale behind my decisions and actions, I aim to convey a deeper understanding of the challenges faced and the strategies employed to overcome them.

In summary, this thesis report encapsulates the essence of my internship experience, encapsulating the myriad experiences, knowledge gained, and skills honed throughout this transformative journey. It is my hope that this report not only serves as an informative account but also as an inspiration for future interns and a testament to the value of hands-on learning in a professional environment.

Chapter 3.

TOOLS AND TECHNOLOGIES

3.1. Introduction

As an intern, I was part of the Data Loss Prevention® team, which is responsible for protecting sensitive data from unauthorized access, use, disclosure, disruption, modification, or destruction. They do this by implementing a variety of security controls, including:

- **Data classification:** The DLP team works with business units to identify and classify sensitive data. This helps to ensure that appropriate security controls are put in place to protect the data.
- **Data discovery:** The DLP team uses a variety of tools to discover sensitive data that is stored on-premises, in the cloud, and in mobile devices. This helps to ensure that all sensitive data is protected, even if it is not stored in a traditional data center.
- **Data protection:** The DLP team uses a variety of security controls to protect sensitive data, including encryption, access controls, and auditing. This helps to ensure that unauthorized users cannot access or modify sensitive data.
- **Data breach response:** The DLP team has a plan in place to respond to data breaches. This plan includes steps to identify the breach, contain the breach, and recover from the breach.

The DLP team at Trellix is committed to protecting the sensitive data of its customers. They work tirelessly to stay ahead of the latest threats and to implement the most effective security controls. As a result, Trellix is one of the most trusted providers of data loss prevention solutions in the world.

Here are some of the specific tasks that the DLP team at Trellix performs:

- Develop and implement data loss prevention policies and procedures.
- Monitor and analyze data flows to identify potential data loss threats.
- Investigate and respond to data loss incidents.
- Provide training and education to employees on data loss prevention best practices.
- Work with other teams within Trellix to ensure that data loss prevention is integrated into all aspects of the business.

The DLP team at Trellix plays a critical role in protecting the sensitive data of Trellix customers. By working tirelessly to stay ahead of the latest threats and to implement the most

effective security controls, the DLP team helps to ensure that Trellix customers can focus on their business without worrying about data loss.

As a part of this team, I worked on the ePO (ePolicy Orchestrator), which was primarily written in Java and Perl. It uses the Tomcat container to host its various components, along with Microsoft SQL and IBM DB2 Servers. This section highlights my learnings as well as technical knowledge that I have gained and experienced as an intern, through either training (as part of trainings conducted by campus hire team) or my contributions as a team member in ongoing company projects.

3.2. Agile Development

In my team, we followed agile development, which is a software development methodology that emphasizes flexibility, collaboration, and iterative delivery. It was created as a response to the traditional waterfall model, which followed a linear and sequential approach to software development. In contrast, agile development focuses on delivering working software in short iterations called sprints.

One of the key principles of agile development is customer collaboration. Agile teams actively involve customers or stakeholders throughout the development process. By engaging with customers early and frequently, the team gains a better understanding of their needs and can quickly adapt to changing requirements. This collaboration ensures that the final product meets the customer's expectations and provides value to the end-users.

Another important aspect of agile development is iterative delivery. Instead of waiting until the end of the development cycle to deliver a complete product, agile teams aim to deliver functional increments of the software at regular intervals. This iterative approach allows for continuous feedback, which enables the team to make improvements and adjustments as they go. It also provides an opportunity to demonstrate progress to stakeholders and gather valuable input early on.

Agile development relies heavily on self-organizing cross-functional teams. These teams are composed of individuals with different skill sets and expertise, such as developers, testers, designers, and product owners. By working together closely and collaboratively, team members can collectively make decisions, share knowledge, and overcome challenges. This collaborative environment fosters creativity, innovation, and a sense of ownership among team members.

To facilitate the agile development process, teams often utilize frameworks such as Scrum or Kanban. Scrum is an iterative and incremental framework that divides the work into time-boxed sprints. Each sprint consists of planning, execution, review, and retrospective phases. Kanban, on the other hand, visualizes the workflow using a Kanban board, allowing teams to

manage and optimize their work in progress. These frameworks provide structure and guidance while still allowing teams to adapt their practices to suit their specific needs.

Continuous improvement is a fundamental principle of agile development. Through regular retrospectives, teams reflect on their processes, identify areas for improvement, and implement changes. This focus on continuous learning and adaptation helps teams become more efficient, productive, and responsive to customer needs.

Agile development also promotes transparency and visibility. Teams often use tools like burndown charts, task boards, and progress reports to track and communicate their work. This transparency enables stakeholders to have a clear understanding of the project's status, progress, and any potential obstacles. It fosters trust among team members and stakeholders and promotes open and effective communication.

Agile development consists of various components, principles, and practices that work together to deliver software in an iterative and collaborative manner. Let's delve into its different aspects:

1. Components of Agile Development:

- a. Customer Collaboration: Engaging customers or stakeholders throughout the development process to gather requirements, provide feedback, and ensure customer satisfaction.
- b. Self-Organizing Teams: Cross-functional teams that work collaboratively, make decisions collectively, and take ownership of their work.
- c. Iterative Delivery: Breaking down the development process into small, incremental iterations to deliver working software at regular intervals.
- d. Continuous Improvement: Regularly reflecting on the process, identifying areas for improvement, and making changes accordingly.
- e. Adaptive Planning: Embracing change by continuously refining and adjusting project plans based on customer feedback and evolving requirements.

2. Agile Principles and Practices:

- a. Agile Manifesto: The Agile Manifesto outlines the core values and principles of agile development, emphasizing individuals and interactions, working software, customer collaboration, and responding to change.
- b. Scrum Framework: A popular agile framework that employs time-boxed iterations called sprints, with specific roles (e.g., Scrum Master, Product Owner) and ceremonies (e.g., Daily Standups, Sprint Review) to guide the development process.
- c. Kanban Method: An agile framework that visualizes the workflow using a Kanban board, limiting work in progress, and enabling teams to optimize their workflow.
- d. User Stories: Brief, user-focused descriptions of software functionality that capture requirements from a user's perspective.

- e. Continuous Integration and Delivery: Practices that automate the integration and deployment of code changes, enabling frequent and reliable software releases.
- f. Test-Driven Development (TDD): A development approach that emphasizes writing automated tests before writing the actual code, ensuring code quality and facilitating code refactoring.

3. Phases of Agile Development:

- a. Initiation: Defining the project vision, identifying stakeholders, and establishing the initial requirements.
- b. Planning: Collaboratively creating a prioritized product backlog, estimating effort, and defining sprint goals.
- c. Execution: Conducting iterative sprints, where development tasks are performed, and working software increments are delivered.
- d. Review: Demonstrating completed work to stakeholders, gathering feedback, and incorporating it into subsequent sprints.
- e. Retrospective: Reflecting on the completed sprint, identifying process improvements, and planning for the next sprint.

4. Advantages of Agile Development:

- a. Flexibility: Agile allows for adapting to changing requirements and market conditions, enhancing product relevance.
- b. Faster Time-to-Market: Iterative delivery enables early and frequent releases, allowing for quicker feedback and faster product iterations.
- c. Customer Satisfaction: Regular customer collaboration ensures that the software meets their evolving needs and expectations.
- d. Transparency: The use of visual tools and frequent communication fosters transparency among team members and stakeholders.
- e. Continuous Improvement: Agile promotes a culture of learning and adaptation, leading to improved processes and higher-quality software.

5. Disadvantages of Agile Development:

- a. Uncertainty: Agile relies on evolving requirements, which can introduce uncertainty and make long-term planning challenging.
- b. Scope Creep: Frequent changes in requirements may lead to scope creep if not managed effectively, potentially impacting timelines and budgets.
- c. Resource Requirements: Agile development requires active involvement from stakeholders and dedicated team members, which may strain resources.
- d. Documentation: Agile places less emphasis on comprehensive documentation, which can be a challenge for maintaining knowledge and onboarding new team members.
- e. Team Dependency: Agile heavily relies on collaboration and cross-functional teams, and any disruptions or conflicts within the team can impact productivity.

In summary, agile development is a software development methodology that emphasizes customer collaboration, iterative delivery, self-organizing teams, and continuous improvement. By embracing flexibility, adaptability, and transparency, agile teams are able to deliver high-quality software that meets customer needs in a dynamic and rapidly changing environment.

3.3. The Angular Framework

Angular is a popular open-source framework developed by Google for building dynamic web applications. It is based on TypeScript, a superset of JavaScript that adds strong typing and additional features to the language. Angular follows the component-based architecture and provides a comprehensive set of features and tools to simplify the development process.

Key features of Angular:

1. **Components:** Angular uses a component-based architecture where the application is divided into reusable and independent components. Each component encapsulates its own HTML template, styles, and behavior, making it easier to manage and reuse code.
2. **Directives:** Angular provides built-in directives like `ngIf`, `ngFor`, and `ngStyle`, which allow developers to manipulate the DOM and add dynamic behavior to the application.
3. **Templates:** Angular's templates are written in HTML with additional syntax and features provided by Angular. Templates enable developers to define the structure and layout of the user interface.
4. **Dependency Injection (DI):** Angular has a powerful dependency injection system that helps manage dependencies between components and services. DI facilitates code reusability, testability, and modularity.
5. **Reactive Extensions (RxJS):** Angular leverages RxJS, a library for reactive programming, to handle asynchronous operations, such as HTTP requests, event handling, and data streaming.
6. **Routing:** Angular offers a robust routing module for building single-page applications (SPAs). It allows developers to define routes, handle navigation, and load different components based on the current URL.
7. **Forms:** Angular provides two form-building approaches: template-driven forms and reactive forms. Template-driven forms rely on Angular directives to handle form validation and submission, while reactive forms offer a more flexible and reactive approach by using explicit form controls.
8. **Services:** Services in Angular are used to encapsulate business logic, data retrieval, and shared functionality. They promote separation of concerns and code reuse across different components.
9. **Angular CLI:** The Angular Command Line Interface (CLI) is a powerful tool that automates various development tasks, such as project setup, component generation, testing, and deployment.

Advantages of Angular:

1. **Powerful and Comprehensive:** Angular offers a rich set of features and tools that cover all aspects of web application development, including data binding, routing, forms, and testing.
2. **Scalability:** Angular's modular architecture and dependency injection system make it highly scalable, allowing applications to grow and evolve without sacrificing performance.
3. **TypeScript Integration:** Angular is built with TypeScript, which brings benefits such as static typing, enhanced tooling, and improved code maintainability.
4. **Enhanced Performance:** Angular leverages features like Ahead-of-Time (AOT) compilation and lazy loading of modules to optimize application performance and reduce load times.
5. **Active Community and Support:** Angular has a large and active community of developers, providing extensive documentation, tutorials, and support. Regular updates and improvements are driven by Google and the Angular team.
6. **Cross-Platform Development:** With technologies like NativeScript and Ionic, Angular allows developers to build mobile applications for iOS and Android platforms using a single codebase.

Disadvantages of Angular:

1. **Learning Curve:** Angular has a steep learning curve, especially for developers new to TypeScript and the framework's concepts, which may require additional time and effort to become proficient.
2. **Complexity:** The comprehensive nature of Angular can sometimes introduce complexity, especially for small-scale projects where a lighter framework may be more suitable.
3. **Code Size:** Angular applications tend to have a larger bundle size compared to other frameworks, which may impact initial loading times, especially for low-bandwidth or mobile users.
4. **Version Migration:** Upgrading Angular versions can be challenging, as major updates may introduce breaking changes that require code adjustments and additional testing.

In summary, Angular is a feature-rich framework for building dynamic web applications. It offers a comprehensive set of tools, follows a component-based architecture, and leverages TypeScript for enhanced productivity and maintainability. While it comes with a learning curve and potential complexity, Angular provides a robust solution for scalable and high-performance web application development.

3.4. TypeScript

TypeScript is an open-source programming language developed and maintained by Microsoft. It is a superset of JavaScript that adds optional static typing, class-based object-oriented programming, and additional features to the language. TypeScript compiles down to plain JavaScript, making it compatible with all JavaScript environments and browsers.

Key features of TypeScript:

1. **Static Typing:** TypeScript introduces static typing, allowing developers to explicitly define variable types during development. This helps catch errors and provides improved tooling support, such as code completion, refactoring, and type checking at compile time.
2. **Object-Oriented Programming:** TypeScript supports object-oriented programming paradigms like classes, interfaces, inheritance, and access modifiers (public, private, and protected). This enables developers to write more structured and maintainable code.
3. **Type Inference:** TypeScript's type inference system automatically infers the type of variables based on their initialization values. This reduces the need for explicit type annotations while still providing type safety.
4. **Enhanced JavaScript Features:** TypeScript includes features from the latest ECMAScript (JavaScript) standards, such as arrow functions, template literals, destructuring, async/await, and modules. These features improve developer productivity and code readability.
5. **Compatibility with JavaScript Ecosystem:** Since TypeScript is a superset of JavaScript, existing JavaScript code can be gradually migrated to TypeScript. TypeScript can utilize existing JavaScript libraries and frameworks seamlessly.
6. **Advanced Tooling and IDE Support:** TypeScript integrates well with popular code editors and IDEs, providing features like intelligent code completion, refactoring, and error checking. Tools like the TypeScript Compiler (tsc) and TypeScript Language Service enable efficient code compilation and analysis.
7. **Optional Static Typing:** TypeScript allows developers to choose when and where to use static typing. It supports gradual typing adoption, so existing JavaScript codebases can be gradually typed without rewriting everything from scratch.
8. **Rich Configuration Options:** TypeScript provides various configuration options through the tsconfig.json file. Developers can customize compiler options, specify target environments, enable/disable strict mode, and control module resolution.

Advantages of TypeScript:

1. **Type Safety:** Static typing helps catch errors during development, reducing the likelihood of runtime errors and improving code reliability.
2. **Code Maintainability:** TypeScript's object-oriented features and static typing make code easier to understand, refactor, and maintain. It promotes modularity, reusability, and scalable codebases.

3. **Tooling and Productivity:** TypeScript's advanced tooling support and IDE integration enhance developer productivity. Features like code completion, refactoring, and error checking speed up development workflows.
4. **Early Detection of Errors:** TypeScript's static type checking catches errors at compile time, allowing developers to identify and fix issues before running the code.
5. **Improved Collaboration:** Static typing and clear type annotations facilitate collaboration among team members by providing better code documentation and understanding.
6. **ECMAScript Compatibility:** TypeScript supports modern JavaScript features and ECMAScript standards, enabling developers to leverage the latest language enhancements while targeting older JavaScript environments.

Disadvantages of TypeScript:

1. **Learning Curve:** Developers with a background in JavaScript may require some time to learn TypeScript's syntax, features, and type system.
2. **Development Overhead:** Writing explicit type annotations and adhering to TypeScript's type system can add some additional development overhead, especially for small projects or rapid prototyping.
3. **Build Process:** TypeScript code needs to be compiled to JavaScript before it can be executed. This adds an extra build step to the development process.
4. **Community and Library Support:** While TypeScript has gained significant popularity, the availability of TypeScript-specific libraries and frameworks might be relatively less compared to the broader JavaScript ecosystem. However, TypeScript can seamlessly work with JavaScript libraries.

Overall, TypeScript offers a powerful and productive development experience by combining static typing, object-oriented programming, and modern JavaScript features. It provides type safety, improved code maintainability, and enhanced tooling, making it a compelling choice for large-scale applications and collaborative development environments.

3.5. Java

Java is a widely used object-oriented programming language developed by Sun Microsystems (now owned by Oracle) in the mid-1990s. Known for its simplicity, platform independence, and robustness, Java has become one of the most popular programming languages in the world. It is used extensively for developing a wide range of applications, from desktop software to enterprise systems and Android mobile applications.

Key features of Java:

1. **Platform Independence:** Java follows the "write once, run anywhere" principle. It uses the Java Virtual Machine (JVM), which allows Java programs to run on any operating system or hardware that has a compatible JVM implementation. This platform independence makes Java highly portable.
2. **Object-Oriented Programming:** Java is a fully object-oriented language, supporting concepts such as encapsulation, inheritance, polymorphism, and abstraction. Object-oriented programming promotes modularity, code reusability, and maintainability.
3. **Strong Standard Library:** Java provides a rich set of standard libraries and APIs, known as the Java Class Library (JCL) or Java Development Kit (JDK), which offers ready-to-use classes and methods for common programming tasks. The JCL covers areas such as input/output, networking, database connectivity, multithreading, and user interface development.
4. **Memory Management:** Java handles memory management automatically through its built-in garbage collection mechanism. Developers do not need to manage memory explicitly, reducing the risk of memory leaks and other memory-related issues.
5. **Exception Handling:** Java has robust exception handling mechanisms that allow developers to catch and handle exceptional conditions, preventing application crashes and providing graceful error recovery.
6. **Multithreading:** Java supports multithreading, allowing developers to create concurrent and parallel programs. Multithreading enables applications to perform multiple tasks simultaneously, enhancing performance and responsiveness.
7. **Security:** Java has built-in security features, including a security manager, bytecode verification, and the ability to run Java applets in a sandboxed environment. These features help prevent unauthorized access, code execution, and other security vulnerabilities.
8. **Rich Ecosystem:** Java has a vast ecosystem with a large and active community of developers. It offers a wide range of frameworks, libraries, tools, and resources that facilitate application development, testing, and deployment.

Advantages of Java:

1. **Platform Independence:** Java's ability to run on multiple platforms makes it highly versatile and allows for widespread adoption.
2. **Robustness and Reliability:** Java's strong type checking, exception handling, and automatic memory management contribute to creating stable and reliable applications.
3. **Scalability:** Java's modular architecture and support for distributed computing make it well-suited for building scalable and enterprise-level applications.
4. **Broad Industry Adoption:** Java is extensively used in various industries, including finance, healthcare, telecommunications, and e-commerce, ensuring a wide range of job opportunities and a mature ecosystem.
5. **Large Standard Library:** The extensive standard library simplifies development by providing pre-built components and utilities for common tasks, saving time and effort.
6. **Strong Community Support:** Java has a thriving community that offers support, documentation, tutorials, and numerous open-source libraries and frameworks.

Disadvantages of Java:

1. **Performance:** While Java's performance has significantly improved over the years, it is still not as efficient as lower-level languages like C++. However, the performance gap is often negligible for most applications.
2. **Learning Curve:** Java has a steep learning curve, especially for beginners without prior programming experience. It has a large syntax and concepts to grasp.
3. **Memory Consumption:** Java's automatic memory management can result in higher memory consumption compared to languages that allow manual memory control.
4. **Verbosity:** Java's syntax can be verbose, requiring more lines of code compared to some other programming languages. However, modern frameworks and libraries help mitigate this issue.

In summary, Java is a versatile and widely adopted programming language known for its platform independence, object-oriented nature, and extensive standard library. It offers robustness, scalability, and a large ecosystem, making it a popular choice for a wide range of applications.

3.6. API Development with Spring Boot

API development with Spring Boot is a popular and efficient approach for building robust and scalable web APIs. Spring Boot, built on top of the Spring Framework, provides a comprehensive set of tools and libraries that simplify the development process and promote best practices. Here's an overview of key aspects and features of API development with Spring Boot:

1. **RESTful Architecture:** Spring Boot promotes the development of RESTful APIs, adhering to the principles of Representational State Transfer (REST). RESTful APIs use standard HTTP methods (GET, POST, PUT, DELETE) to perform operations on resources, making them widely adopted and easy to consume.
2. **Spring MVC:** Spring Boot leverages Spring MVC (Model-View-Controller) framework for handling incoming requests, managing routing, and generating responses. It allows developers to define REST endpoints, handle data binding, and apply validation rules.
3. **Dependency Management:** Spring Boot simplifies dependency management through its built-in dependency management system. It uses Maven or Gradle as build tools, automatically resolving and configuring the required dependencies based on the project's configuration.
4. **Auto-configuration:** One of the key features of Spring Boot is its auto-configuration capability. It automatically configures the application based on the dependencies present in the classpath, reducing the need for explicit configuration. This enables developers to quickly start building APIs with minimal setup.

5. **Embedded Servlet Container:** Spring Boot provides an embedded servlet container (such as Apache Tomcat, Jetty, or Undertow) that allows developers to package the application as an executable JAR file. This makes deployment and distribution of the API simpler and more portable.
6. **Data Access with Spring Data:** Spring Boot integrates seamlessly with Spring Data, which provides an abstraction layer for interacting with databases. It supports various database technologies, such as MySQL, PostgreSQL, MongoDB, and more. Spring Data's repositories and query methods streamline data access and minimize boilerplate code.
7. **Security:** Spring Security is a powerful security framework that can be easily integrated into Spring Boot APIs. It allows developers to secure endpoints, implement authentication and authorization mechanisms, and protect against common security vulnerabilities.
8. **Error Handling:** Spring Boot provides mechanisms for handling exceptions and errors in a consistent and centralized manner. Developers can define global exception handlers and customize error responses to provide meaningful feedback to API consumers.
9. **Testing:** Spring Boot has robust support for testing APIs. It offers unit testing and integration testing capabilities, allowing developers to write test cases to verify the behavior of API endpoints and ensure the correctness of the application.
10. **Actuator:** Spring Boot Actuator provides production-ready features to monitor and manage APIs. It offers endpoints that expose useful information about the application, such as health checks, metrics, logging, and more. Actuator helps with application monitoring, troubleshooting, and performance optimization.

Advantages of API development with Spring Boot:

1. **Rapid Development:** Spring Boot's auto-configuration and convention-over-configuration approach significantly reduce boilerplate code, enabling developers to quickly build APIs with minimal setup.
2. **Extensive Ecosystem:** Spring Boot benefits from a vast ecosystem of libraries, frameworks, and community support. It provides solutions for various requirements, such as data access, security, caching, and messaging.
3. **Modular and Testable Code:** Spring Boot promotes modular and loosely coupled code through its dependency injection and inversion of control features. This makes the codebase easier to maintain, test, and extend.
4. **Scalability and Performance:** Spring Boot's integration with technologies like Spring Cloud and Spring Integration allows seamless scaling and handling of high loads. It provides performance optimizations and efficient resource utilization.
5. **Documentation and Community Support:** Spring Boot has comprehensive documentation and an active community that offers guidance, support, and a wealth of resources. This makes it easier for developers to learn, troubleshoot, and stay updated with best practices.

Disadvantages of API development with Spring Boot:

1. **Learning Curve:** Spring Boot has a learning curve, especially for developers who are new to the Spring ecosystem. Understanding the various concepts and configurations may require some initial effort.
2. **Overhead for Simple APIs:** Spring Boot's powerful features and auto-configuration may introduce some overhead for small or simple APIs. However, the benefits usually outweigh the overhead for medium to large-scale projects.

In summary, API development with Spring Boot offers a powerful and streamlined approach to building RESTful APIs. It provides a wide range of features, robustness, scalability, and a thriving ecosystem, making it a popular choice for developing production-ready APIs.

3.7. Git

Git is a distributed version control system (DVCS) widely used for tracking changes in source code during software development. It was created by Linus Torvalds, the creator of Linux, in 2005. Git allows multiple developers to collaborate on a project, keeping track of changes, merging modifications, and managing different versions of the codebase.

Key concepts in Git:

1. **Repository:** A Git repository is a central storage location where all the project's files and version history are stored. It contains the complete history of the project and allows developers to track changes and collaborate.
2. **Commits:** A commit represents a snapshot of the project at a specific point in time. It captures changes made to the files, including additions, deletions, and modifications. Each commit is identified by a unique hash code.
3. **Branches:** Git uses branches to create independent lines of development. Each branch represents a different version of the codebase, allowing developers to work on separate features or bug fixes without interfering with each other. Branches can be merged back into the main branch (usually called "master" or "main") once the changes are tested and ready.
4. **Remote Repositories:** Git enables collaboration by supporting remote repositories. Remote repositories are copies of the project stored on servers or online platforms (like GitHub, GitLab, or Bitbucket). Developers can push their changes to remote repositories to share their work with others or pull changes made by others into their local repository.
5. **Pull and Push:** Pulling and pushing are actions that synchronize changes between local and remote repositories. Pulling fetches the latest changes from a remote repository and merges them into the local branch. Pushing sends local commits to the remote repository, making them available to other team members.

6. **Merge and Conflict Resolution:** When merging branches or pulling changes, Git automatically combines the changes made by different developers. In some cases, conflicts may arise when two or more changes conflict with each other. Developers need to resolve these conflicts manually by selecting the desired changes or modifying the code accordingly.

Advantages of Git:

1. **Distributed and Offline Work:** Git is a distributed version control system, meaning that each developer has a complete copy of the project's history. This allows developers to work offline and independently, reducing dependencies on a central server.
2. **Collaboration and Branching:** Git supports efficient collaboration by enabling developers to work on different branches and merge their changes seamlessly. It promotes parallel development and helps manage concurrent work.
3. **Version History and Tracking:** Git keeps a detailed history of all changes made to the project, providing a comprehensive record of modifications. Developers can easily track who made specific changes, revert to previous versions, and analyze the evolution of the codebase.
4. **Branching and Feature Development:** Git's branching capabilities facilitate the creation of feature branches, allowing developers to work on new features or experimental changes without affecting the main branch. This promotes flexibility, code isolation, and ease of testing.
5. **Security and Data Integrity:** Git uses cryptographic hashing to ensure the integrity of the codebase. Each commit is identified by a unique hash, making it resistant to tampering or data corruption. Additionally, multiple copies of the repository provide redundancy and protection against data loss.
6. **Community and Ecosystem:** Git has a large and active community, which means ample documentation, resources, and support available to developers. It also integrates well with various development tools and platforms.

Disadvantages of Git:

1. **Learning Curve:** Git has a steep learning curve, especially for developers new to version control systems. Understanding its concepts, commands, and workflows may require some initial effort.
2. **Complexity:** Git offers a wide range of features and commands, making it a powerful but complex tool. Developers may need time to become proficient and comfortable with its functionalities.

In summary, Git is a distributed version control system that allows developers to track changes, collaborate on projects, and manage different versions of the codebase. It provides powerful

features, flexibility, and a robust ecosystem, making it a popular choice for software development teams.

3.8. Maven

Maven is a popular build automation and dependency management tool used primarily for Java projects. It was developed by the Apache Software Foundation and provides a structured and efficient approach to building, packaging, and managing software projects. Maven simplifies the build process, resolves dependencies, and promotes best practices in project organization. Here's an overview of key aspects and features of Maven:

1. **Project Object Model (POM):** Maven uses a Project Object Model, defined in an XML file called the POM. The POM describes the project's configuration, dependencies, build settings, and plugins. It serves as the central configuration file for the project and allows for consistency and reproducibility across different development environments.
2. **Dependency Management:** Maven handles dependency management by automatically resolving and downloading required libraries and frameworks from remote repositories. Developers specify dependencies in the POM, and Maven ensures that the required dependencies and their transitive dependencies are available during the build process. This simplifies the management of external libraries and makes the project more portable.
3. **Build Lifecycle:** Maven defines a standard build lifecycle consisting of phases and goals. Phases represent different stages of the build process, such as compilation, testing, packaging, and deployment. Goals are specific tasks performed within each phase. Developers can execute different phases and goals based on their needs, or they can rely on the default lifecycle provided by Maven.
4. **Build Plugins:** Maven supports a wide range of build plugins that extend its functionality. Plugins provide additional tasks and features, such as generating documentation, running tests, code analysis, deployment, and more. Developers can easily configure and include plugins in the POM to automate various aspects of the build process.
5. **Convention over Configuration:** Maven follows a convention-over-configuration approach, which means that it uses sensible defaults and standard project structures. By adhering to these conventions, developers can minimize the need for explicit configuration. This promotes consistency across projects and simplifies project setup.
6. **Repository Management:** Maven supports the use of local and remote repositories to store and share project artifacts. Local repositories cache downloaded dependencies and plugins on the developer's machine, while remote repositories serve as central locations for sharing and distributing artifacts with team members or the wider community. Maven integrates with popular remote repository managers like Nexus and Artifactory.
7. **Modular and Multi-Module Projects:** Maven enables developers to organize projects into modules, allowing for better code separation and modularity. Multi-module projects can have a hierarchical structure, where each module represents a subproject

with its own POM and build configuration. Maven simplifies the management and interdependencies between modules, making it easier to handle large-scale projects.

8. **Test and Reporting:** Maven provides built-in support for running tests and generating test reports. It integrates with testing frameworks like JUnit and generates reports in various formats, including HTML, XML, and plain text. These reports provide valuable insights into test coverage, failures, and performance.
9. **Integration with IDEs:** Maven integrates well with popular Integrated Development Environments (IDEs) like Eclipse, IntelliJ IDEA, and NetBeans. IDEs can import Maven projects, automatically configure project settings, and provide features like code completion, build triggers, and dependency management within the IDE environment.

Advantages of Maven:

1. **Dependency Management:** Maven simplifies the management of project dependencies, ensuring that the required libraries are easily resolved, downloaded, and included in the build process.
2. **Consistent Project Structure:** Maven's convention-over-configuration approach encourages a standard project structure, making it easier for developers to understand and navigate different projects.
3. **Build Automation:** Maven automates the build process, allowing developers to focus on coding rather than manually configuring and executing complex build scripts.
4. **Standardization and Reproducibility:** Maven's POM enforces standardization and provides a single source of truth for project configuration. This ensures that builds are reproducible across different environments and avoids configuration-related issues.
5. **Extensibility:** Maven's plugin system allows developers to extend its functionality and customize the build process to fit specific project requirements.
6. **Community and Ecosystem:** Maven has a large and active community, with a vast number of plugins, resources, and documentation available. Developers can leverage this ecosystem for support, best practices, and additional functionality.

Disadvantages of Maven:

1. **Learning Curve:** Maven has a learning curve, especially for developers new to build automation tools. Understanding the concepts, POM configuration, and build lifecycle may require some initial effort.
2. **Performance Overhead:** Maven's dependency resolution process and its reliance on remote repositories can introduce some performance overhead, especially in large projects or when working with slow or unstable network connections.

In summary, Maven simplifies the build process and dependency management for Java projects. It promotes best practices, provides a structured approach to project organization, and integrates well with IDEs and external tools. Maven's strengths lie in its dependency

management capabilities, convention-over-configuration approach, and extensive plugin ecosystem, making it a popular choice for Java developers.

3.9. Artifactory

Artifactory is a universal artifact repository manager developed by JFrog, a company specializing in DevOps and software release management. It serves as a central hub for managing and storing software artifacts such as binary files, Docker images, packages, libraries, and other build outputs. Artifactory supports various package formats and integration with popular build tools, making it a versatile solution for artifact management in software development and deployment pipelines.

Key Features of Artifactory:

1. **Artifact Management:** Artifactory provides a centralized location for storing, organizing, and managing software artifacts. It supports popular package formats like Maven, Gradle, npm, NuGet, PyPI, RubyGems, Docker, and more.
2. **Repository Management:** Artifactory allows users to create and manage multiple repositories within a single instance. Repositories can be set up for different package formats, local caching, remote proxying, and virtual aggregations.
3. **Dependency Management:** Artifactory enables efficient dependency management by caching remote artifacts locally. This reduces external network calls and improves build performance.
4. **Metadata and Search:** Artifactory captures extensive metadata for artifacts, including versioning information, build details, licensing, and custom properties. The metadata can be used for search, filtering, and organizing artifacts.
5. **Security and Access Control:** Artifactory provides robust security features, allowing administrators to define access permissions, authentication methods, and secure repositories. It integrates with external user management systems like LDAP, Active Directory, and SAML.
6. **Build Integration:** Artifactory seamlessly integrates with popular build tools and CI/CD systems such as Jenkins, TeamCity, Bamboo, and others. It can trigger builds, resolve dependencies, and deploy artifacts as part of the build pipeline.
7. **High Availability and Replication:** Artifactory supports clustering and high availability configurations, ensuring continuous availability of artifacts. It also provides replication capabilities for distributing artifacts across multiple Artifactory instances or geographical locations.
8. **Artifactory Query Language (AQL):** AQL is a powerful query language provided by Artifactory to search and filter artifacts based on various criteria like name, version, properties, and more.
9. **Artifactory User Plugins:** Artifactory allows users to extend its functionality by writing custom user plugins using Groovy. This enables automation, custom workflows, and integration with external systems.

10. RESTful API: Artifactory exposes a comprehensive RESTful API, enabling programmatic access to its features. This API can be used for automation, integration with external tools, and building custom applications on top of Artifactory.

Artifactory is widely adopted by organizations of all sizes and industries due to its flexibility, scalability, and comprehensive feature set. It helps streamline the software development lifecycle, improve build and deployment efficiency, and ensure the reliability and traceability of artifacts throughout the process.

3.10. TeamCity

TeamCity is a popular continuous integration and delivery (CI/CD) server developed by JetBrains. It provides a powerful and flexible platform for automating and managing the software build, test, and deployment processes. TeamCity is designed to help development teams streamline their workflows, increase productivity, and improve the overall quality of their software projects.

Key features of TeamCity:

1. **Build Automation:** TeamCity allows you to automate the build process for your software projects. It supports a wide range of build tools, languages, and frameworks including Java, .NET, Ruby, Python, Node.js, and more. TeamCity can compile source code, run tests, package artifacts, and generate build reports.
2. **Continuous Integration:** TeamCity enables continuous integration by automatically building and testing your code whenever changes are committed to the version control system. It provides real-time feedback on build status, test results, and code coverage, helping to identify issues early in the development process.
3. **Distributed Build System:** TeamCity supports distributed builds, allowing you to distribute the build workload across multiple build agents and run builds in parallel. This helps to reduce build times and optimize resource utilization.
4. **Build Pipelines and Dependencies:** TeamCity allows you to define complex build pipelines with dependencies between different build configurations. You can specify triggers, dependencies, and conditions to create a flexible and automated workflow for building, testing, and deploying your applications.
5. **Integration with Version Control Systems:** TeamCity integrates seamlessly with popular version control systems such as Git, Subversion, Mercurial, Perforce, and others. It can monitor repositories for changes and trigger builds automatically.
6. **Test Automation:** TeamCity provides extensive support for running automated tests as part of the build process. It integrates with testing frameworks and tools like NUnit, JUnit, MSTest, Selenium, Cucumber, and more. Test results are aggregated and displayed in the TeamCity UI for easy analysis.

7. **Code Quality Analysis:** TeamCity offers built-in code inspection and analysis tools, including support for static code analysis, code coverage, and duplicate code detection. It can integrate with external code quality tools like SonarQube, PMD, and ReSharper.
8. **Continuous Deployment:** TeamCity facilitates continuous deployment by integrating with deployment tools and platforms such as Octopus Deploy, AWS CodeDeploy, Docker, Kubernetes, and others. It allows you to define deployment pipelines and automate the release of your applications.
9. **Notifications and Collaboration:** TeamCity provides various notification mechanisms, including email notifications, build status badges, and integration with collaboration tools like Slack and Microsoft Teams. This helps to keep the development team informed about build and test results.
10. **Extensibility:** TeamCity offers a plugin system that allows you to extend its functionality. You can develop custom plugins or leverage existing plugins from the JetBrains plugin repository to integrate with external tools and customize your CI/CD workflows.

TeamCity is widely used by development teams across different industries and scales, ranging from small startups to large enterprises. Its user-friendly interface, robust feature set, and extensive integration options make it a popular choice for automating software build and deployment processes.

3.11. Confluence and Jira

Confluence and Jira are two powerful software tools developed by Atlassian that work together to enhance collaboration, knowledge sharing, and project management within organizations.

Confluence serves as a centralized knowledge base and collaboration platform, allowing teams to create, organize, and share content. It provides a space for creating documentation, meeting notes, project plans, and technical specifications. With Confluence, teams can collaborate in real-time, comment on pages, and receive notifications about updates and discussions. It supports rich content creation with text formatting, images, attachments, and embedded multimedia. Confluence also offers page templates, ensuring consistency and quick start for various types of documentation.

Jira, on the other hand, is a robust project management and issue tracking tool. It enables teams to plan, track, and manage their work efficiently. With Jira, teams can create and track issues, tasks, bugs, and user stories. It supports different project management methodologies, including agile methodologies like Scrum and Kanban. Jira provides customizable workflows that reflect team-specific processes, allowing for status updates, transitions, and approvals. It also offers agile boards, backlogs, and sprints for managing agile workflows, along with features like story points, burndown charts, and velocity tracking. Jira supports collaboration through comments, mentions, and attachments, facilitating communication and transparency within the team.

The integration between Confluence and Jira strengthens the collaboration and project management capabilities of both tools. Users can link relevant Jira issues, projects, and boards to Confluence pages, creating a seamless flow of information and context. This integration allows teams to embed Jira reports, charts, and dashboards directly into Confluence pages, providing a comprehensive view of project progress and metrics. Conversely, Confluence pages can include Jira issue trackers, displaying relevant issues and their status. This tight integration enhances cross-functional collaboration, facilitates knowledge sharing, and ensures alignment between documentation and project management activities.

By combining Confluence and Jira, teams can streamline their workflows, improve communication, and increase productivity. They can leverage Confluence's documentation capabilities to create project plans, requirements, and design documents, while using Jira to manage and track the execution of those plans. The integration between these two tools bridges the gap between knowledge management and project management, creating a cohesive environment for teams to collaborate, document, and deliver successful projects.

3.12. RDP

RDP, which stands for Remote Desktop Protocol, is a proprietary protocol developed by Microsoft that allows users to remotely connect to and control a computer or server over a network. It enables users to access the graphical user interface (GUI) and interact with the remote system as if they were physically present at that computer.

Key Features and Functionality of RDP:

1. **Remote Access:** RDP enables users to establish a remote connection to a computer or server from any location with network connectivity. This allows for remote administration, troubleshooting, and accessing resources hosted on the remote system.
2. **Graphical User Interface (GUI):** RDP provides access to the full GUI of the remote system. Users can view and control the remote desktop, launch applications, open files, and perform tasks as if they were physically present at the remote machine.
3. **Desktop Sharing and Collaboration:** RDP allows multiple users to connect simultaneously to a remote system, facilitating collaboration and remote assistance. Users can share their desktops, transfer files, and work together on projects in real-time.
4. **Secure Communication:** RDP employs strong encryption to secure the communication between the local client and the remote server. This ensures that sensitive data and user credentials transmitted over the network are protected from unauthorized access.
5. **Print and File Sharing:** RDP supports the sharing of local printers and files with the remote system. Users can print documents from the remote server to their local printers and transfer files between the local and remote machines.

6. **Audio and Video Streaming:** RDP allows for the streaming of audio and video content from the remote system to the local client. This is useful for multimedia applications, remote presentations, and accessing media files hosted on the remote server.
7. **Clipboard and Device Redirection:** RDP enables clipboard sharing between the local and remote systems, allowing users to copy and paste text and files across the remote connection. It also supports device redirection, which enables local devices such as USB drives, printers, and smart cards to be used within the remote session.
8. **Multiple Platforms and Operating Systems:** RDP is available for various platforms, including Windows, macOS, Linux, and mobile devices. It supports connections between different operating systems, allowing users to access Windows-based machines from non-Windows devices.
9. **RemoteApp:** RemoteApp is a feature of RDP that allows individual applications or desktops hosted on a remote server to be accessed remotely, without the need to display the full remote desktop environment. This enables users to run specific applications seamlessly on their local devices.

RDP is widely used for remote administration, technical support, software development, and accessing resources hosted on remote servers. It provides a convenient and secure means of remotely accessing and controlling computers or servers, improving productivity and enabling collaboration in distributed work environments.

Chapter 4. MY WORK, LEARNINGS, EXPERIMENTS, & RESULTS

4.1. My Work (& The Motivation Behind It)

I have utilized Maven, Spring Boot, and Angular to develop an ePolicy comparison tool for Data Loss Prevention (DLP) product suit of Trellix. The motivation & object of this project is to provide a user-friendly interface that showcases a tabulated difference between old and new policies of a specific industry. The current output in JSON format poses challenges for non-technical users, hence the need for a more accessible solution.

4.1.1. Development Workflow

4.1.1.1. Front-end Development with Angular

The front-end development process involves creating responsive user interfaces using Angular. Angular provides a robust framework for managing the application's state, implementing the necessary components, and handling user interactions. The Angular CLI (Command Line Interface) facilitates scaffolding, testing, and bundling of the application. With Angular, the project can provide an interactive and intuitive user experience, allowing users to compare and analyze ePolicy differences efficiently.

4.1.1.2. Back-end Development with Spring Boot

The back-end development revolves around Spring Boot, which serves as the foundation for the application's server-side logic. Spring Boot enables the creation of RESTful APIs, which are responsible for handling requests from the front-end and interacting with the database or external services. It provides seamless integration with various data persistence technologies, such as relational databases or NoSQL solutions, ensuring efficient data retrieval and storage. Additionally, Spring Security can be utilized to secure the application and implement access control measures.

4.1.1.3. Integration and Data Management

The integration of the front-end and back-end is achieved through well-defined APIs. Angular's HttpClient module is used to send HTTP requests to the server-side API endpoints developed using Spring Boot. These endpoints handle the requests, perform the required data processing, and retrieve the necessary information from the database or external systems. The retrieved data is then transformed into the desired format for comparison and presented to the user.

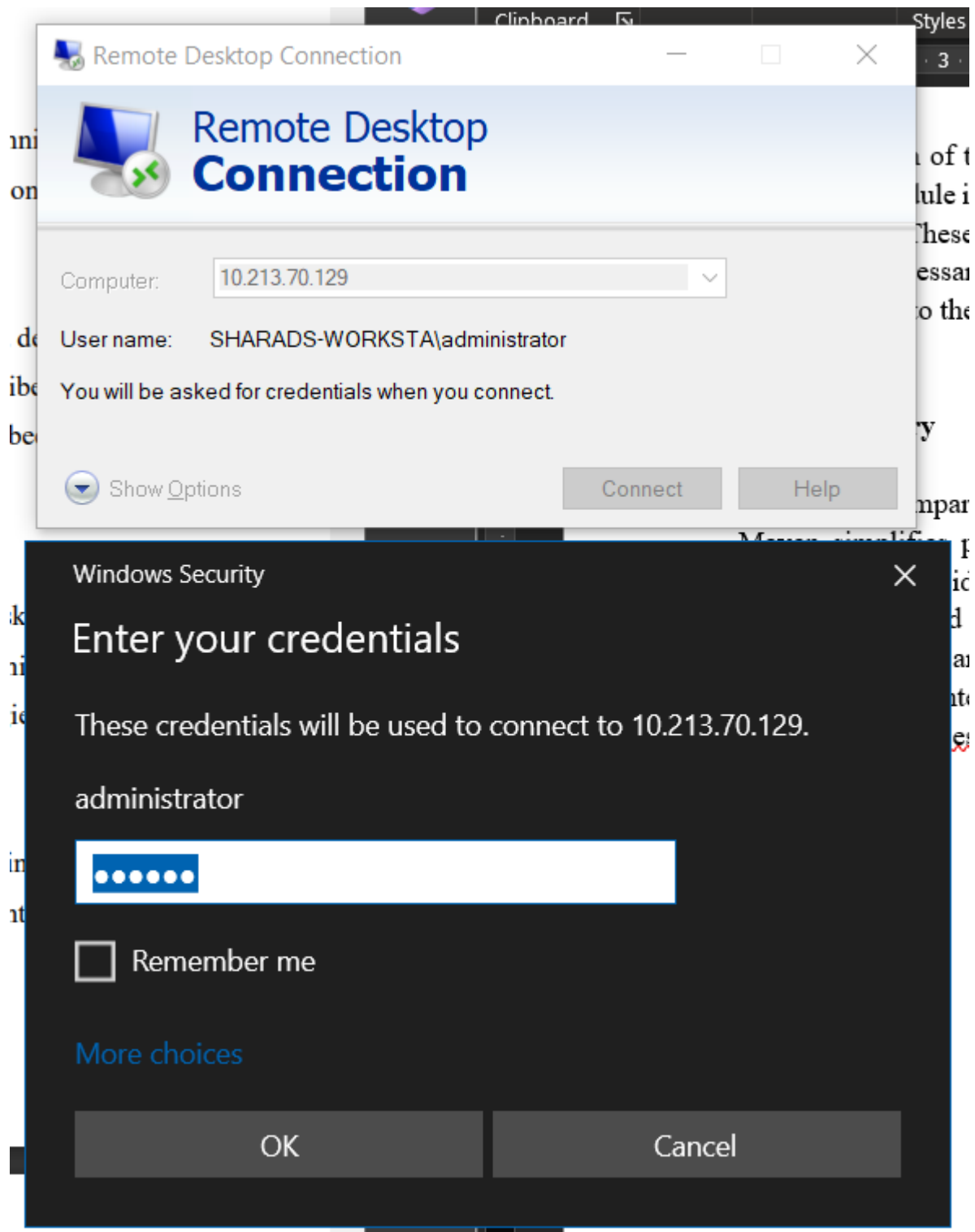


Fig.7. Logging into the RDP

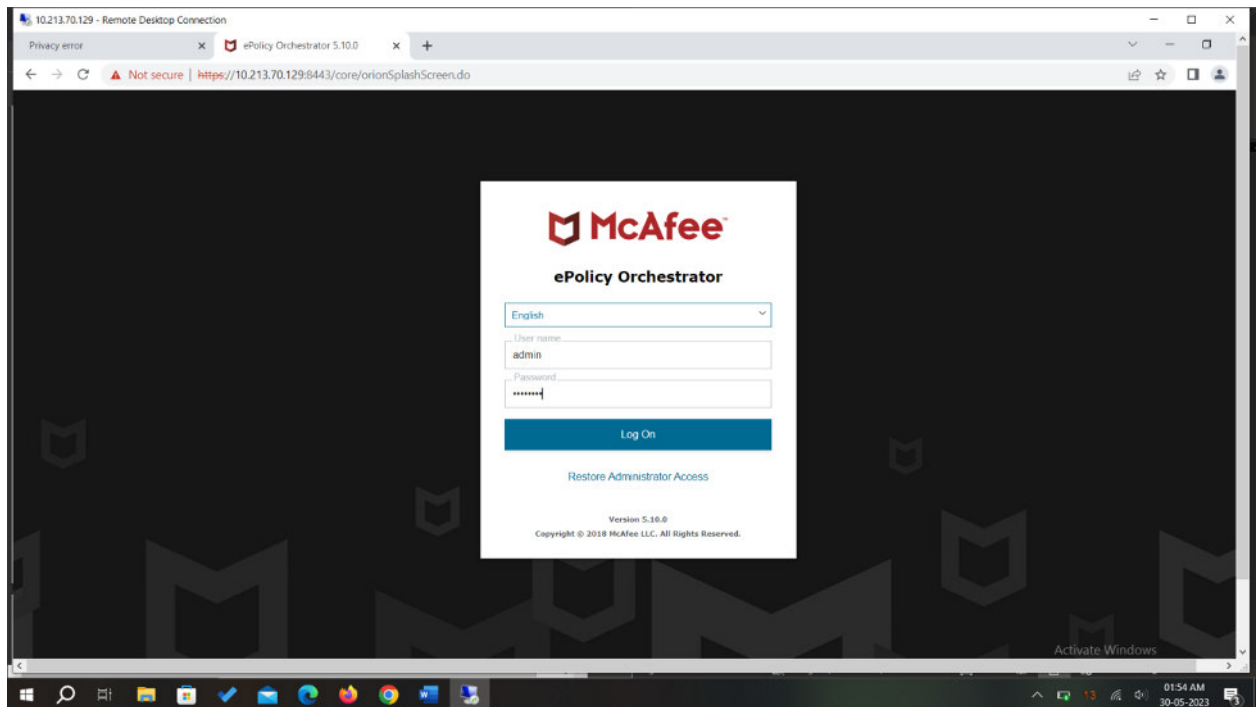


Fig.8. Trellix ePolicy Orchestrator (formerly, McAfee ePO)

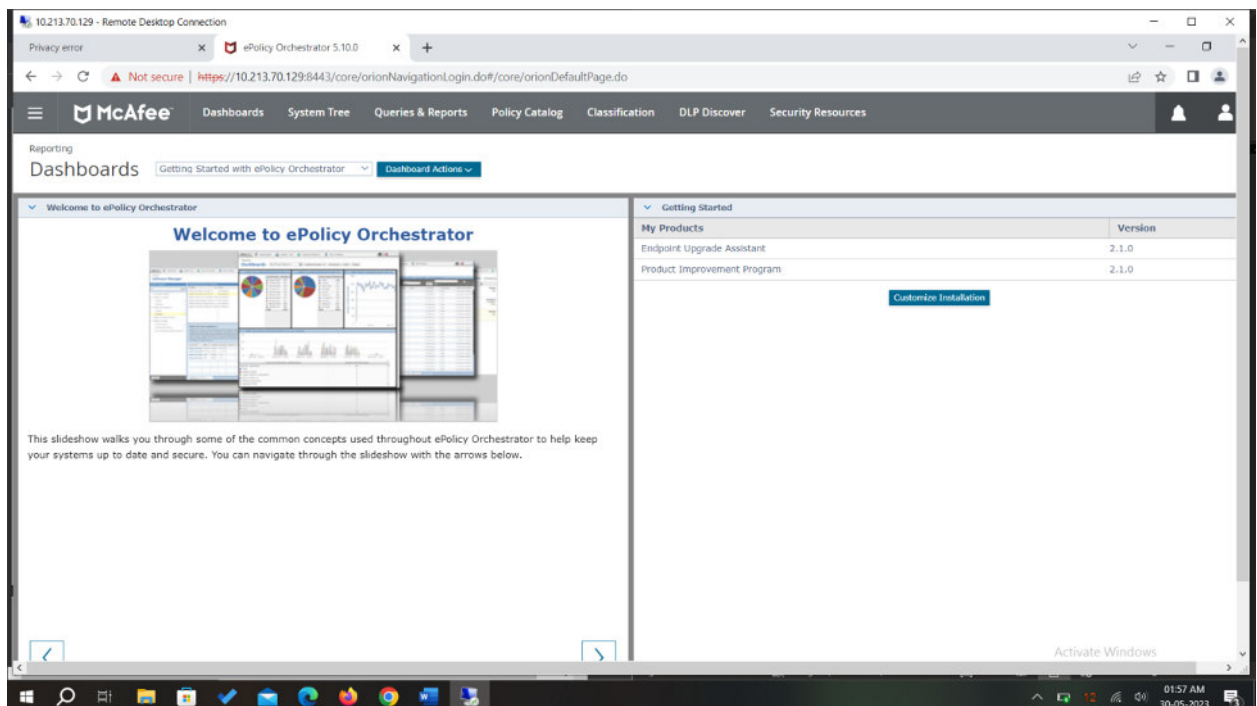


Fig.9. ePO Dashboard

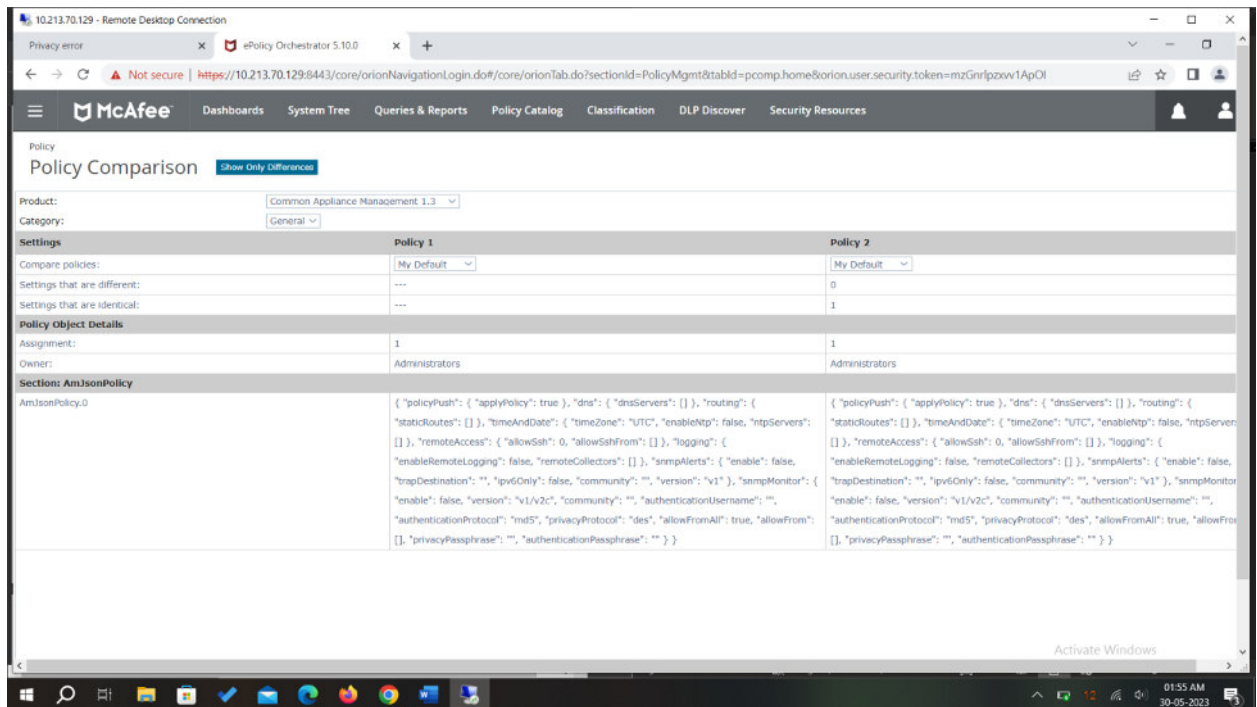


Fig.10. ePO Before (before the creation of Policy Comparison Mechanism)

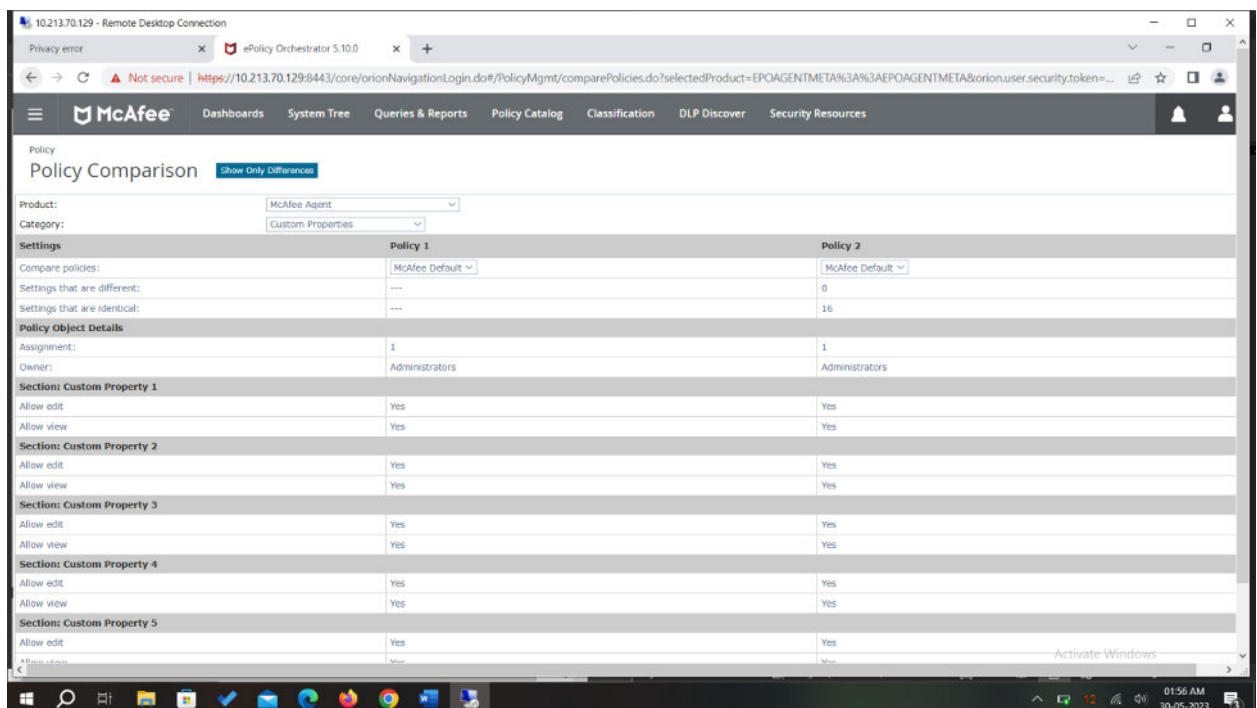


Fig.11. ePO After (After the creation of Policy Comparison Mechanism)

4.1.2. Best Practices

When building a system based on microservices architecture for a mammoth like ePolicy Orchestrator, there are several important principles and practices that should be followed for optimal results.

a. Service Design and Boundaries: It is crucial to carefully design the services and establish clear boundaries based on the business capabilities they represent. Each microservice should have a well-defined responsibility, focusing on a specific function or feature. This approach allows for better organization and maintainability of the system. Adhering to the Single Responsibility Principle ensures that each microservice has a single purpose and is responsible for a specific task or functionality. By clearly defining the service boundaries and responsibilities, the system becomes more modular, scalable, and easier to manage.

b. Communication and APIs: Effective communication between microservices is essential for the system's overall performance and flexibility. To achieve this, it is important to design well-defined and versioned APIs (Application Programming Interfaces). APIs act as the interface through which microservices interact with each other. It is crucial to establish consistent communication protocols, such as using lightweight formats like JSON or employing RESTful APIs. Versioning APIs helps ensure backward compatibility and facilitates seamless transitions when introducing changes or adding new features to the system. Well-designed communication channels and APIs enhance the reliability and maintainability of the microservices ecosystem.

c. Infrastructure and Operations: Proper infrastructure management and efficient operations are vital for the successful implementation of microservices. Adopting containerization technologies like Docker provides a lightweight and portable environment for running microservices. Containers encapsulate the necessary dependencies and configurations, enabling consistent deployment across different environments. Orchestration tools like Kubernetes offer powerful features for managing the deployment, scaling, and monitoring of microservices. They automate tasks such as load balancing, service discovery, and scaling based on demand. By leveraging containerization and orchestration technologies, the system gains flexibility, scalability, and ease of management.

Investing in monitoring, logging, and observability tools is also crucial for understanding and improving the system's behavior. Monitoring tools provide real-time insights into the performance, health, and availability of microservices. Logging helps capture and analyze events and activities within the system, aiding in debugging and troubleshooting. Observability tools allow developers and operations teams to gather information about the internal state of the microservices and identify potential issues or bottlenecks. These tools combined enable proactive measures to maintain system stability, optimize performance, and ensure efficient operations.

d. Testing and Automation: Ensuring the integrity of the overall system requires a comprehensive testing approach. This involves implementing various types of tests throughout the development and deployment process. Unit tests focus on validating the functionality and behavior of individual microservices. Integration tests verify the correct interaction and communication between different microservices, ensuring they work seamlessly together. Contract tests ensure that the APIs conform to the agreed specifications and prevent any breaking changes. By implementing these testing strategies, developers can detect issues early on and ensure that the system functions as expected.

Automation plays a critical role in streamlining operations and reducing manual effort. Automating deployment processes allows for faster and more reliable releases. Scaling operations can be automated to adjust resources based on demand, ensuring optimal performance. Operational tasks such as logging, monitoring, and backups can also be automated, freeing up resources and reducing the potential for human error. Continuous integration and continuous deployment (CI/CD) pipelines enable developers to automate the build, test, and deployment processes, resulting in faster and more efficient development cycles.

By following these principles and practices, the design, implementation, and maintenance of a microservices-based system can be optimized, resulting in a scalable, reliable, and efficient architecture.

4.1.3. Summary

The ePolicy comparison project showcases the synergy between Maven, Spring Boot, and Angular. Maven simplifies project management, dependency resolution, and ensures consistent builds. Spring Boot provides a powerful and efficient back-end framework, enabling the creation of RESTful APIs and seamless integration with databases and external services. Angular, with its component-based architecture and powerful data binding, enables the creation of an intuitive and responsive user interface. Together, these technologies form a robust and scalable solution for comparing ePolicies in DLP products.

4.2. Incidents, events, and cases

4.2.1. Incidents and operational events

There are different tools available to view incidents and operational events.

- **Incidents** — The DLP Incident Manager module displays incidents generated from rules. Trellix DLP Endpoint, Trellix Device Control, Trellix DLP Prevent, Trellix DLP Monitor, Cloud DLP, and Skyhigh Security Cloud enforce rules and send incidents to DLP Incident Manager.

- Operational events — The DLP Operations module displays errors and administrative information. Trellix DLP Discover, Trellix DLP Endpoint, and Trellix DLP Prevent send events to DLP Operations.
- Cases — The DLP Case Management module contains cases that have been created to group and manage related incidents.

When multiple Trellix DLP products are installed, the consoles display incidents and events from all products.

The display for both DLP Incident Manager and DLP Operations can include information about the computer and logged-on user generating the incident/event, client version, operating system, and other information.

We can define custom status and resolution definitions. The definition consists of a custom name and color code and can have the status of enabled or disabled. Custom definitions must be added and enabled in DLP Settings on the Incident Manager, Operations Center, or Case Management page before they can be used.

4.2.1.1. Logging events with Syslog

- We can send certain events using the Syslog protocol to a Syslog server. Configure the Syslog server in the Windows Client configuration on the Debugging and Logging page. The events are sent whether rules are configured to trigger the events or not. The following actions are sent automatically when Send DLP Syslog events to Syslog server is enabled:
 - Printing
 - Copy to removable storage
 - Uploading a file to the web
 - Uploading a file to the cloud
 - Sending email
 - Connect or disconnect a plug and play device
 - Connect or disconnect a removable storage device

4.2.1.2. Stakeholders

A stakeholder is anyone with an interest in a particular incident, event, or case. Typical stakeholders are DLP administrators, case reviewers, managers, or users with incidents. Trellix DLP sends automatic emails to stakeholders when an incident, event, or case is created or changed. It can also automatically add stakeholders to the list, for example, when a reviewer is assigned to a case. The administrator also can manually add stakeholders to specific incidents, events, or cases.

Automatic email details are set in DLP Settings. Options on the Incident Manager, Operations Center, and Case Management pages determine whether automatic emails are sent, and who is automatically added to the stakeholders list. The administrator can add stakeholders manually from the DLP Incident Manager, DLP Operations, or DLP Case Management modules.

4.2.2. Monitoring and reporting events

Trellix DLP products divide events into two classes: incidents (that is, policy violations) and administrative events. These events are viewed in the two consoles, DLP Incident Manager and DLP Operations.

When a Trellix DLP product determines a policy violation has occurred, it generates an event and sends it to the Trellix ePO - On-prem Event Parser. These events are viewed, filtered, and sorted in the DLP Incident Manager console, allowing security officers or administrators to view events and respond quickly. If applicable, suspicious content is attached as evidence to the event.

As Trellix DLP products take a major role in an enterprise's effort to comply with all regulation and privacy laws, the DLP Incident Manager presents information about the transmission of sensitive data in an accurate and flexible way. Auditors, signing officers, privacy officials and other key workers can use the DLP Incident Manager to observe suspicious or unauthorized activities and act in accordance with enterprise privacy policy, relevant regulations or other laws.

The system administrator or the security officer can follow administrative events regarding agents and policy distribution status.

Based on which Trellix DLP products you use, the DLP Operations console can display errors, policy changes, agent overrides, and other administrative events.

We can configure an email notification to be sent to specified addresses whenever updates are made to incidents, cases, and operational events.

4.2.3. DLP Incident Manager/DLP Operations

Use the DLP Incident Manager module in Trellix ePO - On-prem to view the security events from policy violations. Use DLP Operations to view administrative information, such as information about client deployment.

DLP Incident Manager has three tabbed pages. On each page the Present drop-down list determines the data set displayed: Data-in-use/motion, Data-at-rest (Endpoint), or Data-at-rest (Network).

- Analytics — A display of six charts that summarize the incident list. Each chart has a filter to adjust the display. The charts display:
 - Top 10 RuleSets
 - Incidents per Type
 - Top 10 Users with Violations
 - Number of Incidents Per Day
 - Top 10 Destinations
 - Top 10 Classifications
- Incident List — The current list of policy violation events.
- Incident Tasks — A list of actions you can take on the list or selected parts of it. They include assigning reviewers to incidents, setting automatic email notifications, and purging all or part of the list.
- Incident History — A list with all historic incidents. Purging the incident list does not affect the history.

DLP Operations has four tabbed pages:

- Operational Event List — The current list of administrative events.
- Operational Event Tasks — A list of actions you can take on the list or selected parts of it, similar to the incident tasks.
- Operational Event History — A list with all historic events.
- User Information — Displays data from the user information table.

Detailed information can be viewed by drilling down (selecting) a specific incident or event.

4.2.3.1. User Information

The User Information page displays data from the user information table. The table is populated automatically from user information in incidents and operational events. You can add more detailed information by importing from a CSV file.

Information displayed typically includes user principal name (username@xyz), user log on name, user operational unit, first name, last name, user primary email, user manager, department, and business unit. The complete list of available fields can be viewed from the Edit command for the View option.

4.2.3.2. How the Incident Manager works

The Incident List tab of the DLP Incident Manager has all the functionality required for reviewing policy violation incidents. Event details are viewed by clicking a specific event. You can create and save filters to change the view or use the predefined filters in the left pane. You can also change the view by selecting and ordering columns. Color-coded icons and numeric ratings for severity facilitate quick visual scanning of events.

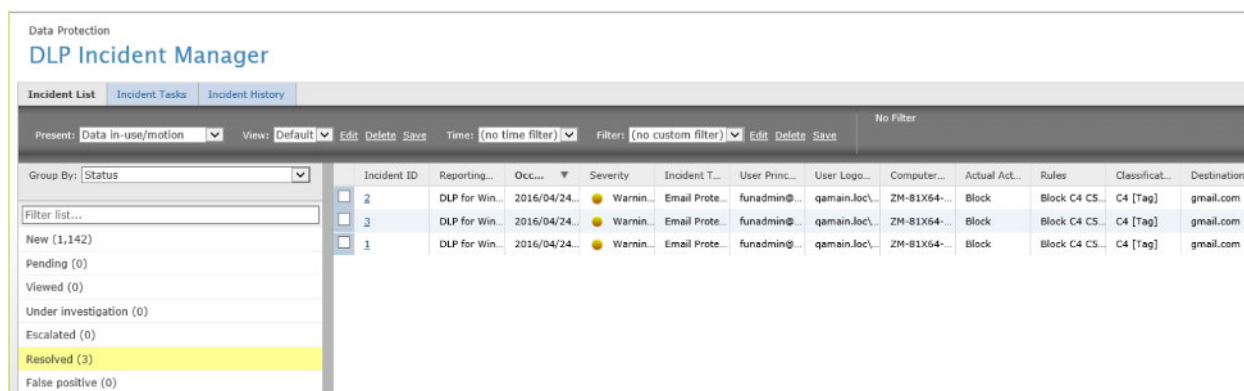
The Incident List tab works with Trellix ePO - On-prem Queries & Reports to create Trellix DLP Endpoint and Trellix DLP appliance reports, and display data on Trellix ePO - On-prem dashboards.

Operations you can perform on events include:

- Case management — Create cases and add selected incidents to a case
- Comments — Add comments to selected incidents
- Email events — Send selected events
- Export device parameters — Export device parameters to a CSV file (Data in-use/motion list only)
- Labels — Set a label for filtering by label
- Release redaction — Remove redaction to view protected fields (requires correct permission)
- Set properties — Edit the severity, status, or resolution; assign a user or group for incident review

4.2.3.2.1. DLP Incident Manager

The DLP Operations page works in an identical manner with administrative events. The events contain information such as why the event was generated and which Trellix DLP product reported the event. It can also include user information connected with the event, such as user logon name, user principal name (username@xyz), or user manager, department, or business unit. Operational events can be filtered by any of these, or by other parameters such as severity, status, client version, policy name, and more.



Incident ID	Reporting...	Occ...	Severity	Incident T...	User Princ...	User Logo...	Computer...	Actual Act...	Rules	Classificat...	Destination
2	DLP for Win...	2016/04/24...	Warnin...	Email Prote...	funadmin@...	qamain.loc...	ZM-81X64...	Block	Block C4 C5...	C4 [Tag]	gmail.com
3	DLP for Win...	2016/04/24...	Warnin...	Email Prote...	funadmin@...	qamain.loc...	ZM-81X64...	Block	Block C4 C5...	C4 [Tag]	gmail.com
1	DLP for Win...	2016/04/24...	Warnin...	Email Prote...	funadmin@...	qamain.loc...	ZM-81X64...	Block	Block C4 C5...	C4 [Tag]	gmail.com

Fig.12. The DLP Incident Manager

Data Protection										
DLP Operations										
Operational Event List Operational Event Tasks Operational Event History User Information										
View: Name Edit Delete Save Time: Last year Filter: (no custom filter) Edit Delete Save Occurred (UTC) is within the last 1 Years										
Group By: User Logon Name (domain\username) Filter list...										
<div> qamain.loc\funadmin (338) DLP\sumit (4) Local Admin (3) QAHAIN\zivtest1 (3) SYSTEM (2) T-W7X86\admin (2) ziv (2) qamain.loc\dipadmin (2) QAHAIN\dipadmin (1) </div>										
Event ID	Occurred (UTC)	Severity	Event Type	User Principal...	User Logon Na...	Computer Name	Reviewer	Status	Error Type	
351	June 15, 2016 1...	Info (0)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	FUN-10X64-4	Unassigned	New	None	
348	June 13, 2016 8...	Critical (4)	DLP Client Excee...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Group1 (Group)	Escalated	None	
347	June 13, 2016 8...	Critical (4)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Group1 (Group)	Escalated	None	
346	June 13, 2016 8...	Critical (4)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Group1 (Group)	Escalated	None	
345	June 13, 2016 8...	Major (3)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Unassigned	New	None	
344	June 13, 2016 8...	Major (3)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Unassigned	New	None	
343	June 13, 2016 8...	Major (3)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Unassigned	New	None	
342	June 13, 2016 8...	Major (3)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Unassigned	New	None	
341	June 13, 2016 8...	Major (3)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Unassigned	New	None	
340	June 13, 2016 8...	Major (3)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Unassigned	New	None	
339	June 13, 2016 8...	Major (3)	DLP Client Install...	funadmin@qama...	qamain.loc\fun...	ZM-81X64-02	Unassigned	New	None	

Fig.13. The DLP Operations

4.2.3.3. Incident tasks/Operational Event tasks

Use the Incident Tasks or Operational Event Tasks tab to set criteria for scheduled tasks. Tasks set up on the pages work with the Trellix ePO - On-prem Server Tasks feature to schedule tasks.

Both tasks tabs are organized by the task type (left pane). The Incident Tasks tab is also organized by incident type, so that it is actually a 4 x 3 matrix, the information displayed depending on which two parameters you select.

	Data in-use/motion	Data at-rest (Endpoint)	Data at-rest (Network)	Data in-use/motion (History)
Set Reviewer	X	X	X	
Automatic mail notification	X	X	X	
Purge events	X	X	X	X
Purge evidence files	X	X	X	X

Table.2. Operational Event (Incident) tasks

Chapter 5.

INDUSTRIAL EXPERIENCE

5.1. Prior Skillset

Prior to starting my internship at Trellix, I had the advantage of acquiring foundational knowledge in development and problem-solving through various courses in my curriculum at Vellore Institute of Technology, Vellore. Courses such as Problem Solving and Programming (CSE1001), Data Structures and Algorithms (CSE2003), Operating Systems (CSE2005), and Software Engineering (CSE3001) provided me with valuable insights and skills that proved to be extremely beneficial during my internship.

These courses equipped me with the necessary tools to approach problems systematically and select optimal solutions. Through projects and assignments completed during these courses, I gained practical experience in applying programming concepts and familiarized myself with different programming languages, their syntax, and their functionality. This knowledge played a crucial role in solving programming problems efficiently.

One of the advantages of the curriculum at Vellore Institute of Technology is the comprehensive blend of theoretical and practical knowledge provided through Classroom Assisted Learning (CAL) courses. Theoretical classes helped strengthen my foundational understanding, while the laboratory and project components allowed me to apply that knowledge in real-world scenarios. This holistic approach ensured that I not only had a theoretical understanding of concepts but also practical experience in their implementation. The skills acquired through these courses were invaluable and proved to be immensely helpful during my internship.

During my internship at Trellix, my existing skills, which I acquired through my curriculum and self-learning, played a significant role in enabling me to grasp the concepts of platform engineering and make meaningful contributions to my data platform team. Some of the skills that proved particularly useful during my internship include:

1. Intermediate knowledge of the Python programming language: This proficiency allowed me to efficiently write code and develop software solutions using Python.
2. Knowledge of shell scripting: Understanding shell scripting languages allowed me to automate tasks and streamline processes, improving efficiency and productivity.
3. Problem assessment: The ability to assess and analyze problems helped me understand the given scenarios and identify the key issues to be addressed.
4. Application of learned concepts: The aptitude to apply the concepts and principles learned in my coursework enabled me to devise effective solutions to complex problems encountered during my internship.

5. Basic problem-solving approach: The understanding of fundamental problem-solving techniques provided a solid foundation for tackling challenges and devising appropriate solutions.

With these skills as a foundation, I was able to successfully complete my internship at Trellix as a software engineer. The knowledge and experience gained from my curriculum and self-learning journey empowered me to contribute effectively to the team and make a positive impact during my time at the company.

5.2. Self Evaluation

During my internship at Trellix, I had the opportunity to apply the theoretical skills I had acquired through my curriculum in practical, real-world applications. The curriculum had primarily focused on imparting theoretical knowledge, but the internship provided me with a platform to experience the complete journey that a product goes through, from being an idea to becoming a fully functional production application. This hands-on experience allowed me to understand the intricacies involved in the development process and significantly contributed to my personal and professional growth.

In addition to technical skills, my internship at Trellix also helped me develop crucial interpersonal skills. One of the most valuable lessons I learned was effective teamwork. Collaborating with my colleagues and actively participating in Agile software development lifecycle taught me the importance of working efficiently as part of a team. I learned how to take ownership of my tasks, prioritize my work, and provide accurate effort estimations. This sense of accountability and effective task management enabled me to contribute more effectively to the team's goals.

Throughout the internship, I gained extensive knowledge and practical experience in various technologies such as Java, Spring, Spring Boot, Maven, and Angular. Working with these technologies exposed me to industry-standard tools and frameworks used in software development. I acquired a deeper understanding of Java programming and its application in building robust and scalable solutions. The experience with Spring and Spring Boot frameworks allowed me to develop efficient and modular applications, leveraging the power of dependency injection and inversion of control. Additionally, working with Maven and Angular enhanced my skills in project management and front-end development, respectively.

The internship at Trellix not only provided me with technical exposure but also taught me valuable lessons in problem-solving and optimizing workflows. I learned how to analyze problems effectively, identify potential bottlenecks, and devise optimal solutions within given time constraints. This experience helped me refine my problem-solving skills and improve my ability to deliver satisfactory results efficiently.

Moreover, being part of a team at Trellox greatly enhanced my communication skills. Collaborating with colleagues, participating in team discussions, and presenting my ideas and solutions to others allowed me to hone my verbal and written communication abilities. Clear and effective communication played a vital role in ensuring seamless collaboration and successful project outcomes.

Overall, my internship experience at Trellox provided me with a holistic understanding of software development processes, enhanced my technical skills, and instilled in me valuable interpersonal and problem-solving abilities. The knowledge gained from working with technologies like Java, Spring, Spring Boot, Maven, and Angular, combined with the practical experience of working on real-life projects, has significantly enriched my professional development.

Chapter 6.

CONCLUSION & FUTURE WORK

6.1. Diagnostics

6.1.1. Diagnostic Tool

The Diagnostic Tool is designed to aid troubleshooting Trellix DLP Endpoint problems on Microsoft Windows endpoint computers. It is not supported on OS X computers.

The Diagnostic Tool gathers information on the performance of client software. The IT team uses this information to troubleshoot problems and tune policies. When severe problems exist, it can be used to collect data for analysis by the Trellix DLP development team.

The tool is installed with the Trellix DLP Endpoint client software package. Look for `hdlpDiag.exe` in the `C:\Program Files\McAfee\DLP\Agent\Tools` folder. Double-click `hdlpDiag.exe` and enter the validation code to open the diagnostic tool utility. It consists of seven tabbed pages, each devoted to a different aspect of Trellix DLP Endpoint software operation.

General information	Collects data such as whether the agent processes and drivers are running and general policy, agent, and logging information. Where an error is detected, information about the error is presented.
DLPE Modules	Displays the agent configuration (as shown in the Trellix DLP Endpoint policy console as the Agent Configuration → Miscellaneous page). It shows the configuration setting and status of each module, add-in, and handler. Selecting a module displays details that can help you determine problems.
Data Flow	Displays the number of events and the memory used by the Trellix DLP Endpoint client, and displays event details when a specific event is selected.
Tools	Allows you to perform several tests and displays the results. When necessary, a data dump is performed for further analysis.

Process list	Displays all processes currently running on the computer. Selecting a process displays details and related window titles and application definitions.
Devices	Displays all Plug and Play and removable devices currently connected to the computer. Selecting a device displays details of the device and related device definitions. Displays all active device control rules and relevant definitions from the device definitions.
Active policy	Displays all rules contained in the active policy, and the relevant policy definitions. Selecting a rule or definition displays the details.

Table.3. *Diagnostic Tools in ePO Policy Comparison Module*

6.1.2. Tuning Policies

The Diagnostic Tool is used to troubleshoot or tune policies.

6.1.2.1. Use case 1: High CPU usage

Users are sometimes plagued by slow performance when a new policy is enforced. One cause might be high CPU usage. To determine this, go to the Process List tab. If you see an unusually large number of events for a process, this could be the problem. For example, a recent check found that taskmgr.exe was classified as an Editor, and had the second highest number of total events. It is quite unlikely that this application is leaking data, and the Trellix DLP Endpoint client does not need to monitor it that closely.

To test the theory, create an application template. In the Policy Catalog, go to DLP Policy → Settings and set an override to Trusted. Apply the policy, and test to see if performance has improved.

6.1.2.2. Use case 2: Creating effective content classification and content fingerprinting criteria

Tagging sensitive data lies at the heart of a data protection policy. Diagnostic Tool displays information that helps you design effective content classification and content fingerprinting

criteria. Tags can be too tight, missing data that should be tagged, or too loose, creating false positives.

The Active Policy page lists classifications and their content classification and content fingerprinting criteria. The Data Flow page lists all tags applied by the policy, and the count for each. When counts are higher than expected, false positives are suspected.

In one case, an extremely high count led to the discovery that the classification was triggered by Disclaimer text. Adding the Disclaimer to the ignored list removed the false positives. By the same token, lower than expected counts suggest a classification that is too strict.

If a new file is tagged while the Diagnostic Tool is running, the file path is displayed in the details pane. Use this information to locate files for testing.

6.2. Conclusion

In conclusion, my internship experience at Trellix was a transformative journey that allowed me to bridge the gap between theoretical knowledge gained from my curriculum and practical application in real-world scenarios. The opportunity to work on projects from conceptualization to production helped me grasp the intricacies involved in software development, project management, and effective teamwork. Through the utilization of technologies such as Java, Spring, Spring Boot, Maven, Angular, and Pearl, I gained valuable hands-on experience and enhanced my technical skills.

One of the significant accomplishments during my internship was the development of a policy comparator tool for the ePolicy Orchestrator (ePO) system. This tool aimed to streamline and automate the comparison process of policies across Trellix's suite of products. By leveraging Java, Spring Boot, Maven, Angular, and Pearl, I was able to create a robust and efficient tool that addressed the needs of the organization.

The policy comparator tool facilitated the comparison of policies by extracting relevant information from the ePO system and presenting it in a user-friendly manner. The Java programming language provided the foundation for developing the backend functionalities, while the Spring Boot framework allowed for rapid development and easy integration with other components. Maven served as a reliable build tool, managing dependencies and ensuring smooth project compilation. On the frontend, Angular enabled the creation of a responsive and intuitive user interface that enhanced the tool's usability.

Throughout the development process, I encountered various challenges that required problem-solving and optimization. With the use of Pearl, a powerful scripting language, I could automate complex tasks and optimize the tool's performance. Pearl's versatility and ease of use were instrumental in ensuring the efficiency and effectiveness of the policy comparator tool.

The successful completion of the policy comparator tool showcased my ability to apply the knowledge gained from my curriculum and adapt it to real-world scenarios. It also allowed me to demonstrate my proficiency in using industry-standard technologies and frameworks to deliver a high-quality software solution. The tool not only provided immediate value to Trellix by streamlining policy comparison processes but also contributed to my personal and professional growth as a software engineer.

6.3. Scope for Future Work

The policy comparator tool developed during my internship lays a strong foundation for further enhancements and future work. Some areas for potential expansion and improvement include:

1. **Enhanced Reporting Capabilities:** Incorporating advanced reporting features within the tool would provide users with comprehensive insights into policy differences and aid in decision-making. Generating visualizations and customizable reports can assist administrators in analyzing policy changes more effectively.
2. **Integration with Continuous Integration/Continuous Deployment (CI/CD) Pipelines:** Integrating the policy comparator tool into the organization's CI/CD pipelines would allow for automated policy comparison as part of the deployment process. This would further streamline operations and ensure policy consistency across different environments.
3. **Expansion to Support Additional Products:** While the current implementation focuses on comparing policies across Trellix's suite of products, future work could involve expanding the tool's capabilities to encompass other software solutions commonly used in the industry. This would broaden the tool's applicability and increase its value to a wider range of organizations.
4. **Security and Compliance Enhancements:** Strengthening the tool's security features to handle sensitive policy information and ensuring compliance with industry standards and regulations would be crucial for its broader adoption. Implementing encryption, access controls, and auditing mechanisms would bolster the tool's security posture.
5. **Integration with ePO Orchestrator:** Integrating the policy comparator tool directly into the ePolicy Orchestrator system would provide a seamless user experience and allow administrators to perform policy comparisons within the familiar ePO interface. This integration could leverage existing APIs and communication protocols to establish a direct connection between the tool and the ePO system.
6. **Usability and User Experience Improvements:** Enhance the user interface to provide a seamless and intuitive user experience. Incorporate user feedback to improve the tool's

usability, streamline workflows, and introduce features such as search functionality, filters, and customizable views.

In summary, the policy comparator tool developed for the ePolicy Orchestrator at Trellix has laid a strong foundation for policy comparison across different suites of products. However, there are several opportunities for future work, including advanced comparison features, integration with additional products, enhanced reporting and visualization, performance optimization, user interface improvements, and security considerations. Pursuing these avenues will further enhance the tool's functionality, extend its reach, and provide users with a comprehensive and efficient policy management solution.

APPENDICES

Glossary

Term	Definition
Action	What a rule does when content matches the definition in the rule. Common examples of actions are block, encrypt, or quarantine.
Crawling	Retrieving files and information from repositories, file systems, and email. Applicable to Trellix DLP Endpoint (Discovery)
Classification	Used to identify and track sensitive content and files. Can include content classifications, content fingerprints, registered documents, and ignored text.
Content classification	A mechanism for identifying sensitive content using data conditions such as text patterns and dictionaries, and file conditions such as document properties or file extensions.
Content fingerprinting	A mechanism for classifying and tracking sensitive content. Content fingerprinting criteria specify applications or locations, and can include data and file conditions. The fingerprint signatures remain with sensitive content when it is copied or moved.
Data vector	A definition of content status or usage. Trellix DLP protects sensitive data when it is stored (data at rest), as it is used (data in use), and when it is transferred (data in motion).
Definition	A configuration component that makes up a classification.
Discover server	The Windows Server where the Trellix DLP Discover software is installed. You can install multiple Discover servers in your network.
DLP server	A Trellix DLP Discover server that has the server role set to DLP Server. DLP Servers are used to store the registered document database. You can also configure DLP

	Server as a proxy server to copy evidence files to the evidence file share in scenarios where the Trellix DLP appliance doesn't have direct access to the evidence file share.
Device class	A collection of devices that have similar characteristics and can be managed in a similar manner. Device classes apply to Windows computers only, and can have the status Managed, Unmanaged, or Excluded.
File information	A definition that can include the file name, owner, size, extension, and date created, changed, or accessed. Use file information definitions in filters to include or exclude files to scan.
Fingerprinting	A text extraction procedure that uses an algorithm to map a document to signatures. Used to create registered documents and for content fingerprinting.
FIPS compliancy	Cryptographic software is configured and used in a way that is compliant with Federal Information Processing Standard 140-2.
Managed devices	A device class status indicating that Trellix Device Control manages the devices in that class.
Match string	The found content that matches a rule.
MTA	Message Transfer Agent or Mail Transfer Agent Software that transfers electronic mail messages from one computer to another using a client-server application architecture.
Path	A UNC name, IP address, or web address. Trellix DLP Endpoint (Discovery)
Policy	A set of definitions, classifications, and rules that define how the Trellix DLP software protects data.
Redaction reviewer	Allows confidential information in the DLP Incident Manager and DLP Operations consoles to be redacted to prevent unauthorized viewing.
RegDoc package	A package of fingerprint data produced by a Trellix DLP Discover registration scan. RegDoc packages are stored in a registration server (DLP Server) database and can be

	called by Trellix DLP Discover scans or Trellix DLP Monitor and Trellix DLP Prevent policies using REST API calls.
Registered documents	Manual registration — Signatures of the files are uploaded to Trellix ePO - On-prem from Trellix DLP when you manually upload files and create a package. These signatures are made available to and downloaded by the endpoints and appliances from the shared location, which are used to track and classify content.
Repository	A folder, server, or account containing shared files. The repository definition includes the paths and credentials for scanning the data. Trellix DLP Endpoint discovery.
Rule	Defines the action taken when an attempt is made to transfer or transmit sensitive data.
Rule set	A combination of rules.
Scheduler	A definition that specifies scan details and the schedule type, such as daily, weekly, monthly, once, or immediately. Applicable to Trellix DLP Endpoint (Discovery).
Strategy	Trellix DLP Endpoint divides applications into four categories called strategies that affect how the software works with different applications. In order of decreasing security, the strategies are Editor, Explorer, Trusted, and Archiver.
Unmanaged devices	A device class status indicating that the devices in that class are not managed by Trellix Device Control. Some endpoint computers use devices that have compatibility issues with the Trellix DLP Endpoint device drivers. To prevent operational problems, these devices are set to Unmanaged.
Excluded devices	A device class status indicating that Trellix Device Control does not try to control the devices in that class. Examples are battery devices and processors.

Table.4. Explained in layman terms, all the glossaries used in the report.

REFERENCES

1. "Automating Cyber Threat Intelligence Analysis with ePolicy Orchestrator" by A. Balamurugan and R. Sridharan.
2. "Continuous Integration and Delivery Pipelines: A Systematic Review" by A. W. Mohamed et al.
3. "Integrating Angular and Spring Boot for Modern Web Applications" by J. Smith and M. Jones.
4. "Integration Testing in Agile Development" by A. Beizer.
5. ePolicy Orchestrator Documentation: <https://docs.mcafee.com/bundle/epolicy-orchestrator-5.10.0-product-guide/page/GUID-69FF8558-F87D-4F2B-86DF-92A72B4C2DEB.html>
6. TeamCity Documentation: <https://www.jetbrains.com/teamcity/documentation/>
7. Angular Documentation: <https://angular.io/docs>
8. Spring Boot Documentation: <https://spring.io/projects/spring-boot#documentation>
9. "Getting Started with ePolicy Orchestrator" by McAfee:
<https://www.mcafee.com/enterprise/en-us/assets/faqs/foundstone/getting-started-with-epo.html>
10. "CI/CD Pipelines: What, Why, and How" by Atlassian:
<https://www.atlassian.com/continuous-delivery/principles/ci-cd-pipeline>
11. "Introduction to TeamCity" by JetBrains: <https://www.jetbrains.com/teamcity/guide/>
12. "The Practice of Network Security Monitoring: Understanding Incident Detection and Response" by R. Bejtlich.
13. "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by J. Humble and D. Farley.
14. "Angular Development with TypeScript" by M. Kutanovski.
15. "Spring Boot in Action" by C. Walls.
16. "ePolicy Orchestrator: Key Capabilities and Use Cases" by McAfee Blogs:
<https://www.mcafee.com/blogs/other-blogs/mcafee-labs/epo-key-capabilities-use-cases/>
17. "CI/CD Pipelines Explained: A Practical Guide for DevOps" by Semaphore:
<https://semaphoreci.com/blog/cicd-pipeline>
18. "Getting Started with Angular and Spring Boot" by Baeldung:
<https://www.baeldung.com/spring-boot-angular-web>

19. "Integration Testing: What It Is and How to Get Started" by TestCraft:
<https://www.testcraft.io/integration-testing-get-started/>
20. "Microsoft SQL Server Documentation" by Microsoft: <https://docs.microsoft.com/en-us/sql/sql-server/>