Read more: Boosty

I. Introduction

US20220232015A1 is a patent filed by Ravi Prasenna and assigned to Netskope, Inc. The patent was filed on July 30, 2021, and published on July 21, 2022. The patent describes a system that includes a network security system interposed between clients and cloud applications. This system is configured to generate a synthetic request and inject it into an application session to transmit the synthetic request to a cloud application. The system also includes inline metadata generation logic configured to issue synthetic requests.

II. Main Idea

The main idea of the patent is to provide a network security system that can effectively monitor and control the flow of document files within a corporate network, particularly focusing on identifying and managing potential security threats. The system uses an inline proxy as an intermediary between the cloud and the corporate network, controlling files that come from outside the corporate network. It identifies document files attempting to enter the corporate network using various methods and metadata which identifies the document file origin. The system also categorizes documents as sanctioned (allowed without threat scanning), blacklisted (automatically and permanently blocked), or unknown (evaluated and potentially quarantined for further analysis). The patent emphasizes the use of policy-based rules, threat scanning, and sandboxing for unknown or potentially malicious documents.

The patent US20220232015A1 presents several key points and takeaways:

- **Network Security System**: The patent describes a system for network security that is interposed between clients and cloud applications. This system is designed to enhance security in cloud-based environments
- **Synthetic Request Generation**: The system is configured to generate a synthetic request and inject it into an application session. This synthetic request is then transmitted to a cloud application
- **Inline Metadata Generation Logic**: The system includes inline metadata generation logic. This logic is configured to issue synthetic requests, which can provide additional security measures
- **Separate Synthetic Requests**: The technology disclosed relates to an inline proxy configured with synthetic request injection. It can generate, during the application session, synthetic requests that are separate from the incoming requests
- **Cloud Policy Enforcement**: The synthetic request injection is used to retrieve metadata for cloud policy enforcement. This suggests that the system can be used to enforce security policies in cloud applications

A. Benefits

Benefits as follows are:

- **Enhanced Security**: The system provides a robust mechanism for monitoring and controlling the flow of document files within a corporate network, particularly those shared via cloud-based storage, which enhances security
- **Proactive Threat Detection**: By using synthetic requests to generate metadata, the system can proactively detect and respond to potential security threats before they impact the network
- **Dynamic Policy Enforcement**: The inline metadata generation logic allows for dynamic enforcement of cloud security policies based on real-time metadata, which can adapt to changing threat landscapes
- **Efficiency**: The system can improve data throughput efficiency by automatically blocking known malicious files without the need for deep threat scanning, reducing latency
- **Efficient Metadata Generation**: The inline metadata generation logic issues synthetic requests to provide metadata to the second point of presence, ensuring efficient and timely metadata generation
- **Stability and Consistency**: The system's use of unique file IDs ensures that files can be tracked and managed consistently throughout their lifecycle, even if file names change
- **Synthetic Request Generation**: The system is configured with synthetic request injection, which can generate synthetic requests separate from incoming requests during an application session. This can help in better monitoring and controlling network traffic
- **Protection Against Malicious Attacks**: The system can identify and block documents from known malicious websites, thereby protecting the network from potential threats
- **Flexible and Dynamic**: The system can adapt to different instances (personal or corporate) and can handle documents from various sources like Google Drive, Docs, Sheets, etc

SECURITY

Read more: Boosty

- **Improved Data Throughput Efficiency**: By automatically discarding blacklisted URLs known to include malicious objects or links, the system reduces latency and improves data throughput efficiency

B. *Drawbacks*

Drawbacks as follows are:

- **Complexity**: Implementing and managing the system may add complexity to the network infrastructure, requiring specialized knowledge and potentially increasing the administrative overhead
- **False Positives/Negatives**: The system may incorrectly categorize legitimate documents as threats (false positives) or fail to detect actual threats (false negatives), which could disrupt normal business operations or leave the network vulnerable
- **Maintenance and Updates**: The metadata store and policy rules may need regular updates to keep up with evolving threats, which can be resource-intensive and require continuous attention from security teams
- **User Experience Impact**: The process of blocking and quarantining documents could impact the user experience, especially if legitimate documents are delayed or if users must navigate additional security steps
- **Over-Reliance on Known Threats**: The system's effectiveness against known malicious sites and files might not extend to zero-day threats or sophisticated attacks that have not yet been identified and categorized
- **Performance Impact**: The additional processing required to generate synthetic requests and analyze metadata could potentially impact network performance, especially in high-traffic environments
- **Adaptability**: The system's ability to adapt to new types of cloud services and applications may be limited by its current design and may require further development to handle emerging technologies
- **Privacy Concerns**: The collection and analysis of metadata might raise privacy concerns, depending on the type of data collected and how it is used within the system
- **Cost**: The implementation and operation of such a security system may incur significant costs, including hardware, software, and personnel expenses

III. Network Security System

The "Network Security System" is a system designed to enhance security for communications between clients and cloud applications:

- **Interposition**: The system is interposed between clients and cloud applications, acting as a mediator or proxy to monitor and potentially modify the traffic
- **Synthetic Request Injection**: The system generates synthetic requests that are injected into application sessions. These synthetic requests are used to interact with cloud applications, separate from the actual client requests
- **Inline Metadata Generation**: The system includes logic that generates metadata inline with the traffic flow. This metadata is used to issue synthetic requests, which can be leveraged for various security purposes, such as policy enforcement or security assessment
- **Cloud Policy Enforcement**: The synthetic requests are used to retrieve metadata that is crucial for enforcing cloud security policies. This suggests that the system can dynamically adapt and enforce security measures based on the metadata obtained from the synthetic requests
- **Points of Presence**: The system may include multiple points of presence that intermediate traffic. These points of presence are equipped with the inline metadata generation logic and are capable of issuing synthetic requests
- **Redundancies in Metadata Synchronization**: The system addresses potential redundancies in metadata synchronization between the points of presence, which is important for maintaining consistency and efficiency in security operations

A. *Significance of 'Network Security System'*

As the patent primarily focuses on a network security system that enhances the security of corporate networks, the system is designed to control and monitor files that come from outside the corporate network, particularly those shared via cloud-based storage. It uses an inline proxy as an intermediary between the cloud and the corporate network.

The system identifies document files attempting to enter the corporate network using various methods and metadata, which identifies the document file's origin. The metadata is stored in a metadata store accessible by the inline proxy. The system allows documents originating from sanctioned sources, such as known organizations with a previous history with the corporate network, to enter the network without threat scanning.

The system also identifies and blocks document files received from known malicious websites. These are websites and URLs that have been associated with phishing attacks in the past or in any other way compromise network security. The metadata store tracks, stores, and maintains a database of all known blacklisted sites.

For unknown documents, the system evaluates their ownership and other metadata properties to identify the source. If a document cannot be identified as to its source, it is temporarily blocked from entering the corporate network. This involves policy-based rules and matching techniques. The document is quarantined and initially threat scanned. If it is certain that malicious code may be involved, the document will enter a sandbox for further analysis

IV. Synthetic Request Generation

"Synthetic Request Generation" is a key component of the network security system described in patent. Synthetic request generation is a method used in synthetic monitoring or testing, where artificial or "synthetic" requests are created to mimic real

SHARKSECURITY

Read more: Boosty

user traffic. These requests are used to interact with systems, such as cloud applications, separate from actual client requests. The purpose of synthetic request generation is to test and monitor the performance and functionality of systems, helping to identify potential issues before they affect real users.

- **Definition**: Synthetic request generation involves creating artificial or "synthetic" requests that mimic real user traffic. These requests are used to interact with systems, such as cloud applications, separate from actual client requests
- **Purpose**: Synthetic requests are used to test and monitor the performance and functionality of systems. They can help identify potential issues before they affect real users, ensuring that systems are working properly before they go into production
- **Use in Network Security**: In the context of the patent, synthetic requests are injected into application sessions and transmitted to cloud applications. This allows the system to interact with the cloud applications and retrieve metadata for cloud policy enforcement
- **Generation Process**: Synthetic requests can be generated programmatically, often using scripts or tools designed for synthetic monitoring or testing. These tools can simulate a variety of scenarios, object types, and environment variables
- **Benefits**: Synthetic request generation allows for proactive monitoring of system performance and functionality. It can help identify issues early, before they affect real users, and can provide valuable information on system uptime, response times, and transaction success rates
- **Challenges**: While synthetic request generation can provide valuable insights, it also comes with challenges. For example, it may not fully replicate the diversity and unpredictability of real user behavior. Additionally, it requires careful design and implementation to ensure that the synthetic requests accurately represent the interactions they are intended to mimic

Synthetic request generation can be used in various ways:

- **Load Testing**: Synthetic requests can be used to evaluate how a system behaves under heavy load, helping to identify if a website or application is likely to crash due to a spike in user traffic
- **Transaction Monitoring**: Developers or QA engineers can use synthetic requests to determine how a system handles a specific type of request
- **Component Monitoring**: In distributed systems, such as microservices applications, synthetic requests can be directed at specific components to measure their response
- **API Monitoring**: Synthetic API tests enable engineers to assess whether APIs manage requests as required
- **Data Privacy**: Synthetic data generation can enable the creation of larger datasets, enhance model performance, and protect individual privacy

Potential applications of synthetic request generation are wide-ranging and can be found in various fields:

- **Software Development and Quality Assurance**: Synthetic requests can be used to test virtually any type of user transaction or request, for any purpose. If a real user can initiate a request, that request can also be monitored synthetically
- **Network Security**: Synthetic requests can be used to retrieve metadata for cloud policy enforcement
- **Performance Monitoring**: Companies can leverage synthetic testing to proactively monitor the availability of their services, the response time of their applications, and the functionality of customer transactions
- **User Experience Optimization**: Synthetic monitoring can be used to understand how a real user might experience an app or website, helping to identify optimization opportunities

A. Significance of 'Synthetic Request Generation"

The significance of 'Synthetic Request Generation' lies in its application within a network security system to enhance the monitoring and control of interactions with cloud applications:

- **Proactive Security Measures**: Synthetic request generation is used to proactively test and monitor the performance and functionality of cloud applications, which is crucial for identifying and addressing potential security issues before they impact real users
- **Metadata Retrieval**: The inline metadata generation logic within the network security system issues synthetic requests to provide metadata. This metadata is then used for cloud policy enforcement, allowing the system to dynamically adapt and enforce security measures
- **Cloud Policy Enforcement**: Synthetic requests are used to retrieve metadata that is crucial for enforcing cloud security policies. This suggests that the system can dynamically adapt and enforce security measures based on the metadata obtained from the synthetic requests
- **Enhanced Monitoring**: Synthetic monitoring, which includes synthetic request generation, is a critical component in network performance monitoring and digital experience monitoring. It enables IT development, NetOps, and DevOps teams to improve user experiences and optimize business-critical functions
- **Versatility in Testing**: Synthetic request generation can be used to test virtually any type of user transaction or request, for any purpose, providing a comprehensive approach to system testing and monitoring

V. Inline Metadata Generation Logic

This inline metadata generation logic is part of a broader trend in cybersecurity innovation, where companies are

START SECURITY

investing in research and development to create advanced security solutions that can protect against evolving threats in the cloud and network environments.

The "Inline Metadata Generation Logic" is a key component of the network security system:

- **Definition**: Inline metadata generation logic refers to the system's ability to generate metadata "inline" or in real-time as the traffic flows through the system. Metadata is data about data, providing additional context or information about the data being processed
- **Function**: The inline metadata generation logic is configured to issue synthetic requests. These synthetic requests are used to interact with cloud applications and retrieve metadata for cloud policy enforcement
- **Role in Network Security**: inline metadata generation logic plays a crucial role in enhancing network security. By issuing synthetic requests and retrieving metadata, the system can enforce security policies dynamically based on the metadata obtained from the synthetic requests
- **Points of Presence**: The system includes multiple points of presence that intermediate traffic. Each of these points of presence is equipped with the inline metadata generation logic and is capable of issuing synthetic requests
- **Redundancies in Metadata Synchronization**: The system addresses potential redundancies in metadata synchronization between the points of presence. This is important for maintaining consistency and efficiency in security operations

The "Inline Metadata Generation Logic refers to the system's ability to generate metadata "inline" or in real-time as the traffic flows through the system. Metadata is data about data, providing additional context or information about the data being processed.

The purpose of inline metadata generation logic is to issue synthetic requests. These synthetic requests are used to interact with cloud applications and retrieve metadata for cloud policy enforcement. By issuing synthetic requests and retrieving metadata, the system can enforce security policies dynamically based on the metadata obtained from the synthetic requests.

The inline metadata generation logic works by monitoring the traffic flow between clients and cloud applications. As the traffic flows through the system, the logic generates metadata in real-time. This metadata is then used to issue synthetic requests, which are injected into application sessions and transmitted to cloud applications.

The key points regarding "Inline Metadata Generation Logic" are:

- **Real-Time Metadata Creation**: The logic is designed to generate metadata in real-time as network traffic passes through the system
- **Synthetic Request Issuance**: It is configured to issue synthetic requests, which are separate from actual client requests, to interact with cloud applications and retrieve necessary metadata
- **Cloud Policy Enforcement**: The metadata generated by the inline logic is used for enforcing cloud security policies, allowing the system to dynamically adapt and enforce security measures
- **Operational Efficiency**: Inline metadata generation helps maintain operational efficiency by ensuring that metadata is generated and applied to traffic without significant delay
- **Redundancy Management**: The system may include multiple points of presence with inline metadata generation logic, and it addresses potential redundancies in metadata synchronization between these points
- **Enhanced Security**: By generating metadata inline, the system can proactively respond to security threats and enforce policies more effectively

Potential applications of inline metadata generation logic are wide-ranging and can be found in various fields:

- **Network Security**: Inline metadata generation logic can be used to enhance network security by enforcing security policies dynamically based on the metadata obtained from synthetic requests
- **Software Development and Quality Assurance**: Inline metadata generation logic can be used in software development and testing to monitor and analyze the behavior of applications in real-time
- **Performance Monitoring**: Inline metadata generation logic can be used to monitor the performance of systems and applications in real-time, helping to identify potential issues before they affect real users
- **Data Management**: Inline metadata generation logic can be used in data management systems to keep track of changes to data and maintain consistency and efficiency in operations
- **API Development**: Inline metadata generation logic can be used in API development to provide additional context or information about the data being processed, enhancing the functionality and usability of APIs
- **Research and Development**: Supporting reproducible computational research by providing metadata that documents the computational processes and data lineage
- **Compliance and Governance**: Ensuring that data handling complies with relevant regulations and governance policies

A. Significance of Inline Metadata Generation Logic

The significance 'Inline Metadata Generation Logic' is that enhances the system's ability to enforce cloud security policies and improve the overall security of corporate networks:

SOLARY SECURITY

- **Metadata Generation**: The inline metadata generation logic is designed to issue synthetic requests to provide metadata. This metadata is essential for the operation of the network security system, particularly for enforcing cloud security policies
- **Cloud Policy Enforcement**: The metadata generated by the inline metadata generation logic is used to enforce cloud security policies. This suggests that the system can dynamically adapt and enforce security measures based on the metadata obtained from the synthetic requests
- **Network Security Enhancement**: The inline metadata generation logic is a critical component of the network security system. By generating and utilizing metadata, the system can better monitor and control interactions with cloud applications, thereby enhancing the overall security of the corporate network
- **Efficiency and Accuracy**: Centralizing business logic into a metadata layer, as done by the inline metadata generation logic, can help eliminate errors and improve efficiency. This is particularly beneficial in complex network environments where accurate and efficient operation is crucial

VI. Separate Synthetic Requests

The term "Separate Synthetic Requests" refers to synthetic requests that are generated and issued separately from the incoming requests during an application session. These synthetic requests are not responses to client requests but are independently generated by the system.

The system described in the patent includes an inline proxy configured with synthetic request injection. This proxy can generate synthetic requests that are separate from the incoming requests during the application session. These separate synthetic requests are used to interact with cloud applications and retrieve metadata for cloud policy enforcement.

The generation of separate synthetic requests allows the system to interact with the cloud applications independently of the client's actions. This can provide additional security measures, as the system can retrieve metadata and enforce security policies dynamically based on the metadata obtained from the synthetic requests.

Here are the key points:

- **Separate from Client Requests**: These synthetic requests are not responses to client requests but are independently generated by the system
- **Interaction with Cloud Applications**: The separate synthetic requests are used to interact with cloud applications and retrieve metadata for cloud policy enforcement
- **Real-Time Metadata Retrieval**: The generation of separate synthetic requests allows the system to interact with the cloud applications independently of the client's actions. This can provide additional security measures, as the system can retrieve metadata and enforce security policies dynamically based on the metadata obtained from the synthetic requests
- **Enhanced Security**: The use of separate synthetic requests can enhance the security of cloud applications by allowing the system to proactively retrieve metadata and enforce security policies
- **Potential Applications**: Separate synthetic requests can be used in various fields, including network security, performance monitoring, software development and quality assurance, data management, and API development

Potential applications of separate synthetic requests could include:

- **Network Security**: Separate synthetic requests can be used to enhance network security by enforcing security policies dynamically based on the metadata obtained from the synthetic requests
- **Performance Monitoring**: Separate synthetic requests can be used to monitor the performance of systems and applications in real-time, helping to identify potential issues before they affect real users
- **Software Development and Quality Assurance**: Separate synthetic requests can be used in software development and testing to monitor and analyze the behavior of applications in real-time
- **Data Management**: Separate synthetic requests can be used in data management systems to keep track of changes to data and maintain consistency and efficiency in operations
- **API Development**: Separate synthetic requests can be used in API development to provide additional context or information about the data being processed, enhancing the functionality and usability of APIs

A. Significance of Separate Synthetic Requests

The use of separate synthetic requests in the network security system enhances the system's ability to enforce cloud security policies, proactively identify potential security issues, and improve the overall security of corporate networks:

- **Metadata Generation**: Separate synthetic requests are used by the inline metadata generation logic to generate metadata. This metadata is crucial for enforcing cloud security policies and for the operation of the network security system
- **Proactive Security Measures**: Separate synthetic requests allow for proactive testing and monitoring of the system's interactions with cloud applications. This can help identify and address potential security issues before they impact real users
- **Cloud Policy Enforcement**: The metadata obtained from separate synthetic requests is used to enforce cloud security policies. This allows the system to dynamically adapt and enforce security measures based on the metadata

SHADY SECURITY

Read more: Boosty

- **Efficiency and Accuracy**: The use of separate synthetic requests can improve the efficiency and accuracy of the network security system. By generating and utilizing metadata from these requests, the system can better monitor and control interactions with cloud applications

VII. Cloud Policy Enforcement

"Cloud Policy Enforcement" refers to the application of security policies in cloud environments based on the metadata retrieved from synthetic requests

The network security system described in the patent is configured to generate synthetic requests and inject them into an application session. These synthetic requests are transmitted to a cloud application, and the responses provide the metadata. The system then applies the policy to the incoming request based on this metadata.

Cloud policy enforcement is crucial for maintaining security in cloud environments. Policies can include rules about access control, data protection, network security, and more. By enforcing these policies, the system can prevent unauthorized access, protect sensitive data, and maintain the integrity of the network.

The system described in the patent enhances cloud policy enforcement by using synthetic requests to retrieve metadata. This allows the system to enforce security policies dynamically based on the metadata obtained from the synthetic requests, providing a more proactive and adaptive approach to cloud security.

Here are the key points:

- **Dynamic Policy Enforcement**: The system dynamically enforces security policies based on the metadata obtained from synthetic requests. This suggests that the system can adapt and enforce security measures in real-time
- **Metadata-Based**: The enforcement of cloud policies is based on the metadata retrieved from synthetic requests. This metadata provides the necessary context for the system to decide which policies to enforce
- **Security Enhancement**: Cloud policy enforcement is a crucial aspect of maintaining security in cloud environments. Policies can include rules about access control, data protection, network security, and more
- **Proactive Security**: The system described in the patent enhances cloud policy enforcement by using synthetic requests to retrieve metadata. This allows the system to enforce security policies proactively, providing a more adaptive approach to cloud security
- **Potential Applications**: Cloud policy enforcement can be used in various fields, including cloud security, access control, data protection, compliance, and risk management

Potential applications of cloud policy enforcement include:

- **Cloud Security**: Cloud policy enforcement is a fundamental aspect of cloud security, helping to protect data, applications, and infrastructure in the cloud
- **Access Control**: Policies can be used to control who has access to certain resources in the cloud, preventing unauthorized access
- **Data Protection**: Policies can be used to protect sensitive data in the cloud, such as encrypting data at rest and in transit
- **Compliance**: Cloud policy enforcement can help organizations comply with regulations and standards related to data protection and privacy
- **Risk Management**: By enforcing policies in the cloud, organizations can manage risks related to security, privacy, and compliance

A. Significance of Separate Synthetic Requests

'Cloud Policy Enforcement' is significant as it pertains to the enforcement of security policies in a cloud environment. This involves the use of synthetic requests and inline metadata generation logic to ensure that data traffic between clients and cloud applications adheres to established security policies. This can help prevent unauthorized access and protect sensitive data, thereby enhancing the overall security of the cloud environment.

VIII. Google Chrome Profile Support'

The "Google Chrome Profile Support" refers to the system's ability to handle and interpret user-session information, such as authentication IDs and session IDs (cookies), associated with a user's Google Chrome profile:

- **User-Session Information**: When a user opens a file (e.g., Google Drive file, Docs, Sheets, etc.) from their corporate login account, the opened file will have the already logged-in user-session information like auth_id and SID (cookies)
- **File Identification**: With the current approach, this file will be identified as already logged in user. This means that the system can recognize and associate the activity with the correct user profile
- **Example Scenario**: For instance, if a user logs onto Gmail with an ID “abc@kkrlog.com” and gets a document from an external user “xyz@gmail.com”. When the user opens the file, it will show that "abc@kkrlog.com" is the user performing the activity and the instance of the file is “kkrlog.com” but "gmail.com” is the actual instance of the file
- **Google Chrome Profile Management**: Google Chrome allows users to create and manage multiple profiles. Each profile has its own set of bookmarks, extensions, and settings. This feature can be used to keep personal and work-related browsing activities separate, ensuring privacy and preventing data leakage
- **Potential Applications**: The ability to handle and interpret user-session information associated with Google Chrome profiles can be used in various fields, including network security, data management, and user experience optimization

SAMPLE

Read more: Boosty

IX. Policies Regarding Attachments

Policies Regarding Attachments outlines the approach to handling file attachments within a corporate network, particularly in relation to cloud applications and services. These policies and mechanisms are designed to enhance security and compliance within corporate environments, particularly when dealing with cloud-based file sharing and collaboration tools:

- **Two Fundamental Policies**: The system distinguishes between two primary policies for users regarding file attachments: "allowed corporate instance" and "blocking personal instance"
- **Corporate Instance Definition**: A corporate instance is defined as a company-sanctioned instance of a cloud application. Even if the owner of a shared file is an external user, if the instance of the file is considered corporate, the "allowed corporate instance" policy will activate, permitting the user to perform activities on externally shared files
- **Identification of File Owner**: The system needs to identify the owner of the created file. To prevent phishing attacks and unauthorized access, external files are not allowed to access the corporate network or perform any activity
- **Traffic Analysis**: When a user receives a document from Google Drive, Docs, Sheets, Slides, etc., via email or shared link, the response transaction data includes the owner of the file. The system uses patterns in the data to determine if the document is created by a personal account or a specific corporate instance
- **Instance Extraction**: The system extracts the instance for file view activity and populates the instance as the owner of the file. For other activities (download/edit), the owner might not be known in the traffic, but the file_id is unique at least across the instance
- **Blocking Personal Documents**: The system helps corporate users block personally created documents from being viewed and allows only corporate documents. However, this may block customers who are accessing personally created documents from their personal instance
- **Instance Determination**: When users view documents from Google Drive, Docs, Sheets, etc., the response data has the instance details. If a user logs in to a personal account, the instance will be gmail.com, and if the user logs in with a corporate account, the instance will be a corporate instance

X. Process Flow

Process Flow describes as the procedure for evaluating document files shared within a corporate network, particularly in relation to potential security threats.

- Malicious Document
- Inline Proxy
- Document Identification
- Sanctioned Documents
- Blacklisted Sites
- Unknown Documents

A malicious document, originating from a malicious website, is shared into a cloud-based store accessible to a corporate network. The goal of a malicious attacker is to make the document enticing so that it would be accessed by multiple users in a corporate network or using remote corporate devices.

The inline proxy, which is part of the network security system, acts as an intermediary between the cloud and the corporate network, controlling files that come from outside the corporate network.

Document files attempting to enter the corporate network are identified by the methods described in the patent and other metadata which identifies the document file origin. The metadata is stored in a metadata store accessible by the inline proxy.

Internal corporate documents are always sanctioned. Documents originating outside the corporate network, if sanctioned, are always allowed into the corporate network without threat scanning. These are documents from known sources, including large organizations and organizations which have a previous history with the corporate network.

Document files received from known malicious websites are identified by the inline proxy as blacklisted sites. These are websites and URLs that have been associated with phishing attacks in the past or in any other way compromises network security. The metadata store tracks, stores, and maintains in a database all known blacklisted sites. Documents received in this category are automatically and permanently blocked.

Unknown documents are evaluated as to their ownership and other metadata properties, which will identify the source of the unknown document. If a document cannot be identified as to its source, it is temporarily blocked from entering the corporate network. This involves policy-based rules including matching techniques. The document is quarantined, and initially threat scanned. Much of this work requires the involvement of a network security administrator. If it is a certainty that malicious code may be involved, the document will enter the sandbox for further analysis.

SRC Security