EGRESS SOLUTION GUIDE

Egress Defend architecture overview and deployment model

egress

Contents

Introduction

The email security landscape is going through its biggest revolution. The old world of email servers protected by secure email gateways (SEGs) is being rapidly replaced with cloud native email environments, predominantly Microsoft 365 (M365).

This tidal wave of cloud adoption and migration to M365 has created a new set of challenges and risks for every organization.

A decade ago, spam and malware infected emails were perceived as the greatest threat. These are now accepted as standard risks and good levels of protection against them are incorporated into all core email platforms, including M365.

Cybercriminals have not stood still. The threat landscape continues to evolve, with attackers becoming more sophisticated and increasingly automated, and the types of attack seen continually bypassing traditional technologies.

This has led to the rise of a new category of anti-phishing technology that Gartner calls 'integrated cloud email security (ICES)'.

Egress Defend is an ICES solution that deploys seamlessly within M365 to detect and neutralize advanced phishing threats.

In this solutions guide, we explore the different ways ICES solutions are commonly architected and explain the architectural and technological approaches we adopted when developing Defend.

Email integration

When integrating new email solutions into their technology stack, customers routinely have the same set of requirements, focused on:

- Simplicity of integration with their existing stack
- A seamless deployment process
- Minimal or zero implementation time
- Service scalability and resilience, which must be watertight as email remains a business-critical application
- 100% uptime

This must all be complemented with sophisticated detection capabilities and streamlined remediation functionality in the event of a potential breach.

How a solution integrates with your architecture will impact these requirements, and there are two different core methods available: API and SMTP (inline).

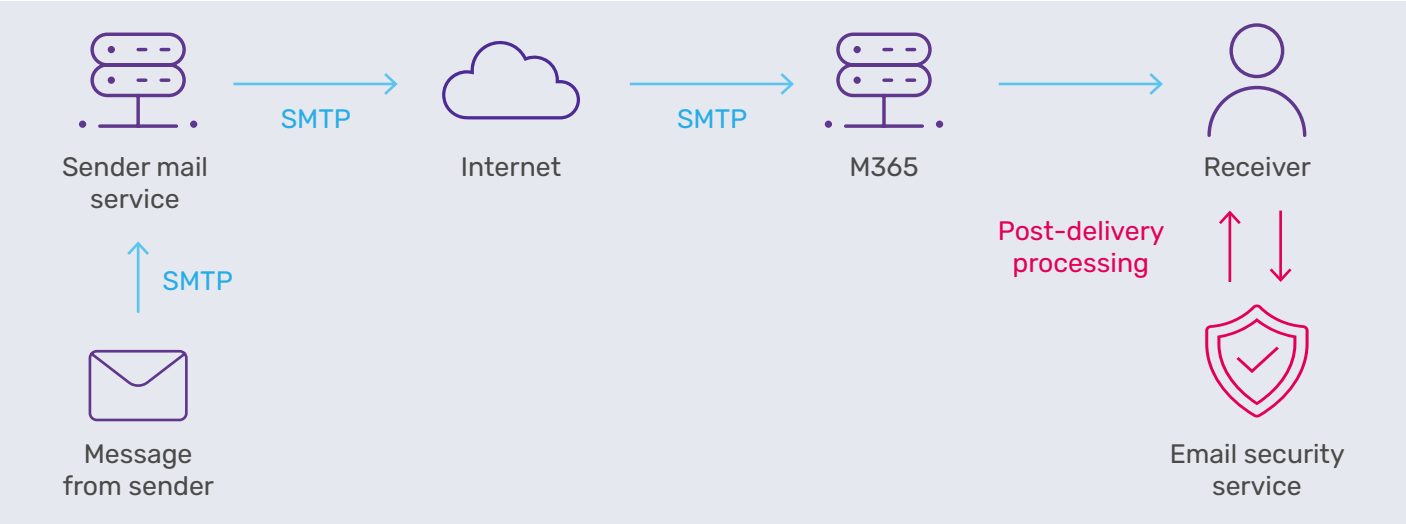
API vs SMTP

Egress has spent many years researching and implementing both API and SMTP email processing. This document summarizes the benefits and weaknesses of both techniques and details the approach Egress has adopted based on this deep analysis.

What is API integration?

APIs are easy to interface with and designed to integrate with a range of systems and services, not just email. A number of new start-ups now offer ICES solutions based only on API.

When a vendor refers to an API–based ICES solution, they are generally referring to interfacing with Microsoft Graph API or Google Mail API, which provide REST APIs for interfacing with core email platforms and other services.

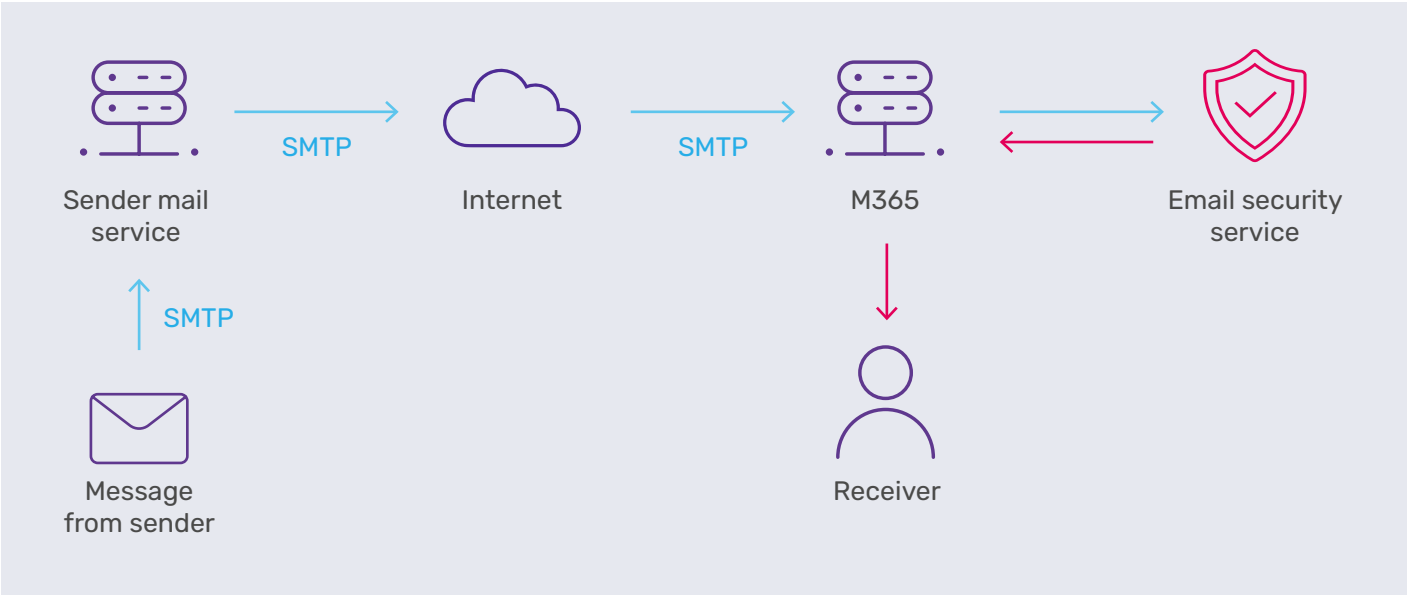


How API processing works for inbound threat detection

What is SMTP (inline) integration?

The core transport mechanism for email, SMTP has been established and refined over the last 20+ years. When discussing SMTP or inline integration, vendors are talking about connectors/transport rules directly applied to their email infrastructure, which are used to forward email for processing.

SMTP configuration can require MX changes when referring to traditional SEG providers or internal rules within the email tenant (Microsoft Exchange, M365, etc.).



How SMTP (inline) processing works for threat detection

The advantages and disadvantages of API-based architecture

Advantages of the API-based approach

The overwhelming benefit of API architecture is touted as simplicity of integration. Deployment can be performed by granting rights via OAuth to one or more service accounts and subscribing to message notifications at the API layer. It is very quick and easy to perform this configuration, as it does not require any changes to email flow. An API-based approach also provides significant benefits for retrospective remediation or modification of messages after an event or trigger.

APIs as an overall approach also give the opportunity to integrate with other data sources and third party systems to improve defenses, including security feeds, security compliance tools, and awareness training.

Disadvantages of the API-based approach

In its current state, API-based ICES architecture comes with some significant disadvantages both in performance and resultant capabilities. These disadvantages all stem from the fact that APIs are designed for non-time critical bulk actions, and not the time-dependent, mission-critical response you need to defend against the most sophisticated cyberattacks.

The biggest challenge with API integration is linked to performance and the well-documented restrictions that are in place. The average time for a phishing email to be acted on by the recipient is 82 seconds.[1] Therefore, it is imperative that messages are processed in real time before a user has the chance to engage.

Microsoft Graph API has poorly documented limits on per message, per mailbox, and tenancy requests that will incur throttling at certain levels. Egress has carried out extensive testing across a range of mailboxes and tenants and witnessed significant and unpredictable throttling, particularly linked to the size of the message, geographic location, and mailbox.

When processing messages in real time, vendors subscribe to notifications for all mailboxes in a tenant. When a message is received, the notification is delivered; the service responds and processes the message, and then forms an opinion on the message. Finally, the system responds accordingly, including to modify, move, or delete the message if it's found to be dangerous.

However, due to the unreliability and inevitable latency of APIs, many vendors put a 'hack' in place.

They create a rule on every user's mailbox to deliver all messages to a separate, hidden folder to avoid the user clicking on a dangerous message in the inbox before it has been processed. If a message is found to be safe, it is moved from the hidden processing folder to the inbox.

This method presents further challenges, as the forwarding rule needs to be continually monitored and reinstated, as it can be deleted by the user. Additionally, if the user has their own mailbox rules, these can interfere with the vendor's enforced rule. Finally, while moving messages is designed to protect the user, if the ICES service is unavailable for any reason, the system admin must decide the best course of action to balance business continuity with risk. Users either need to be granted access to the hidden folder with potential phishing emails present or wait for the service to resume and for emails to be redirected to the users' inboxes.

Other limitations linked to the performance of APIs revolve around their core functionality. To gain the maximum level of protection, it is desirable to modify a message to protect the user. This could be the addition of a warning banner on phishing emails, rewriting dangerous links, or completely neutralizing the payload of a message. While technically possible with API integration, in practice it is near impossible to perform these tasks at scale and in a timely manner. Even if the performance of the core APIs does improve in the future, you'd still have the risk of email clients not synchronizing these changes in time to neutralize a phishing threat, particularly when dealing with geo latency and mobile devices, where that risk may be higher.

It is far easier to move a message than rewrite it, so most vendors opt for this approach and, therefore, must rely on near 100% efficacy to be fully effective.

Moreover, Microsoft and Google regularly change the way their APIs work, which adds a layer of risk to service continuity. Commercially there are also concerns about the viability of MS Graph API as a service. Microsoft has indicated that they are looking to monetize Graph processing and charge per request, changes that have already been introduced to Microsoft Teams.[2] This currently undefined financial overhead would mean reliance on pure Graph API processing could become very costly for the customer and require a Microsoft E5 license.

1 www.wired.com/2015/04/email-phishing-attacks-take-just-minutes-hook-recipients

2 https://devblogs.microsoft.com/microsoft365dev/upcoming-billing-changes-for-microsoft-graph-apis-for-teams-messages/

The advantages and disadvantages of SMTP architecture

Advantages of the SMTP-based approach

The biggest advantage of SMTP results from its longevity and resilience. SMTP is a known entity both from an engineering perspective and for day-to-day management.

As previously stated, timing is critical when providing protection against advanced email-borne attacks. By utilizing SMTP, it is possible to process messages before they arrive in a user's mailbox, thus completely eliminating the chance of a user acting on a message before it has been checked.

SMTP also provides many more options to protect a user from receiving a dangerous email. With SMTP, there are no limitations to how a message can be processed: it can be quarantined, rewritten to include visual notifications, links can be rewritten or neutralized, attachments removed, or the message can be deleted altogether. Essentially, it is possible to classify every message and provide complete management of all malicious emails.

There are effectively no performance limitations aligned to SMTP due the infinite scalability options available. There are no throttling or throughput limitations to contend with and integration can work alongside existing applications and services, including for organizations that want to retain their SEG.

The disadvantages of the SMTP-based approach

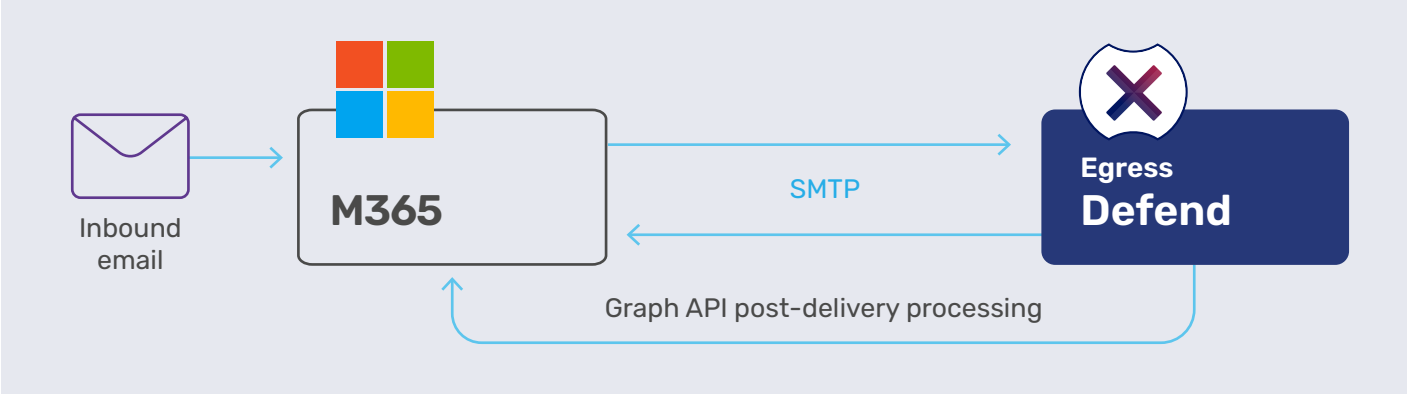
One of the biggest downsides of SMTP is dealing with service interruptions. This is a particularly difficult challenge when MX record changes are required, such as those needed by SEGs. However, there are modern implementation techniques available to provide inline protection without MX record changes.

In the unlikely event of an SMTP service being unavailable, there also needs to be appropriate disaster recovery (DR) and fault tolerance to ensure 100% continuity and no service interruptions. Unlike API-based protection, there is no acceptance of any service outage as this can stop or delay email flow.

Egress Defend: Leveraging both API and SMTP architecture

There are advantages and disadvantages to both methods of operation. SMTP provides the reliable and predictable solution, while API provides greater flexibility and after-the-event protection.

This is why Egress selected a hybrid approach to deliver advanced email security by using the best of both technologies to provide a highly effective and resilient architecture.



Egress Defend architectural overview, which utilizes both SMTP and API-based approaches for efficacy of detection, scalability, and resilience

Message processing

Egress selected SMTP (inline) architecture for message processing as this is currently the only way to guarantee the best combination of protection and service reliability. The average time a user takes to interact with an email is 82 seconds, so SMTP is the only way to guarantee a dangerous email is neutralized before a user can act. Egress has made implementation seamless and addressed resiliency concerns by providing automated disaster recovery (DR) and failover in the unlikely event of system outage.

Most importantly with SMTP processing, it is possible to perform actions on every single email to provide the highest level of protection and user engagement, without the need for the workarounds or hacks required by solutions that only use API integration.

Customizable banners can be added to every message showing the risk level of the email, and full link rewriting can be performed on every message, rather than the selective rewriting offered by API-only products due to the processing limitations. It is also possible to create custom rules to perform different actions on messages based on classification and risk, including deletion, moderation, and quarantine.

Remediation and retrospective action

While API integration is not currently deemed appropriate for real-time message processing, it does provide immense value for remediation and post-delivery actions.

Egress uses Graph API to perform instant remediation on dangerous emails and additional retrospective actions, including message modification. Egress continues to innovate and has imminent plans to add new features that will provide even more dynamic security.

By using SMTP for the heavy lifting, it is also possible to change the classification of a message retrospectively using Graph API. If, for example, a message is later found to be unsafe or maybe the payload has been dynamically updated to become malicious, it is possible for Egress Defend to retrospectively update the message threat.

Efficacy

While many vendors may claim 100% efficacy, in a world of sophisticated cyberattacks this realistically isn't possible. For this reason, there are often gray areas where an email is suspicious and has some of the characteristics of a dangerous email but is not confirmed as a phishing attack.

Egress Defend utilizes informational banners applied to every email. When an email is deemed suspicious, an amber banner is applied to warn the user to take care and report the email if they believe it to be dangerous. Most Graph API-based products don't benefit from this approach, as they move all suspicious messages to Junk or the provisioned hidden folder. If a message is incorrectly classified, the end user either receives a dangerous email to their inbox or a genuine email ends up in the Junk or hidden folder. This either presents a risk to security or creates end user frustration and lost confidence in the solution.

Implementation

The implementation of Egress Defend is simplified into a single click installer that configures all mail processing rules and remediation requirements. Implementation into Microsoft 365 can be performed in minutes, providing immediate protection.

Conclusion

While there is no one right way to setup your email security infrastructure, when evaluating vendors it is important to consider the pros and cons of each deployment method and how they might impact your users, IT and security teams, and overall business objectives. By developing a hybrid model that incorporates both SMTP and API-based architecture, Egress has been able to harness the best of both worlds, providing our customers with the flexibility, scalability, and reliability that is needed to meet the demands of today's threat landscape.

Our partnership and integration with Microsoft 365 create a seamless experience for our customers, across all interfaces, and has even afforded them the opportunity to consolidate their security vendors to reduce overhead.

To learn more about Egress' innovation in the ICES space and how it can help you protect your organization against advanced phishing attacks, schedule your personalized demo today.

About Egress

Egress makes digital communication safer for everyone. As advanced and persistent cybersecurity threats continue to evolve, we recognize that people get hacked, make mistakes, and break the rules. Egress's Intelligent Cloud Email Security suite uses patented self-learning technology to detect sophisticated inbound and outbound threats that protect against data loss, resulting in the reduction of human activated risk. Used by the world's biggest brands, Egress is private equity backed and has offices in London, New York, and Boston.

www.egress.com