SocketLabs
the science of hitting the inbox

Improving Email Security

with the MTA-STS Standard

By Brian Godiksen

An Email **Best Practices** Whitepaper

CONTENTS

Brian Godiksen

*Brian has been helping organizations optimize email deliverability since joining SocketLabs in 2011. He currently manages a team of deliverability analysts that consult with customers on best infrastructure practices, including email authentication implementation, bounce processing, IP address warm-up, and email marketing list management. Brian leads the fight against spam and email abuse at SocketLabs by managing compliance across the platform. He is an active participant in key industry groups such as M3AAWG and the Email Experience Council. You can read more of Brian's content here on the **SocketLabs** website.*

Executive Overview

MTA-STS has already made massive strides towards global adoption and is poised to become a pervasive security standard.

The Edward Snowden leaks of 2013 opened many peoples' eyes to the fact that mass surveillance was possible by intercepting and spying on email transmissions. Today, compromised systems, database thefts, and technology breaches remain common fixtures in news feeds around the world. As a natural response, the technology industry is rabidly focused on improving the security and encryption of communications across all platforms. Since those early days of enlightenment, industry experts have discussed and attempted a variety of new strategies to combat "pervasive monitoring" of email channels. While pervasive monitoring assaults can take many forms, the most prominent forms of interference were man-in-the-middle (MitM) attacks.

When MitM attacks first became a hot topic back in 2014, the obvious question was, "why isn't everyone encrypting their email traffic?" While it took some time to get the process started, that's exactly what the industry has been doing ever since that time. In the US, organizations large and small have been steadily adopting encryption practices, with compliance-focused industries like financial services and healthcare leading the way. But adoption rates across the many corners of the email world have been slow and varied. In short, everyone was enthusiastic and desired to support encryption, but turning those intentions into a universally accepted practice has been complex – and has taken time.

As of 2019, the email industry has largely completed this transition and much of the world's email traffic is encrypted. However, while this represents a tremendous achievement and has certainly improved the email security levels, it has not been the silver-bullet solution that perfectly protects email transmissions from MitM and other types of monitoring threats. Email traffic remains at risk because the technical process by which encryption was deployed relies only on SMTP. While this protocol is the backbone of all email transmission, it has natural design limitations that prevent fail-safe encryption from being applied. With these limitations email messages are still very susceptible to MitM attacks.

In response to this problem, the Internet Engineering Task Force (IETF) – the premier Internet standards body – along with other industry leaders have developed SMTP MTA Strict Transport Security (MTA-STS) RFC 8461. MTA-STS moves email encryption beyond the "opportunistic", SMTP-only methodology of the last five years, to a more advanced approach that enables greater control and security in the use of encryption technology. With Google's support – as evidenced by their recent announcement – MTA-STS has already made massive strides towards global adoption and is poised to become a pervasive security standard.

This paper explains the evolution of email encryption technology and illustrates the many advantages inherent in the new MTA-STS approach.

Why Does Email Need Encryption in Transit?

Applying the ideal encryption technology in an email communication context is challenging.

Ensuring the privacy of email traffic has long been the objective of both senders and receivers. Among the many reasons this is true, is that pervasive monitoring attacks such as man-in-the-middle (MitM) attacks have become so common. MitM attacks are perpetrated by groups or individuals who capture unsecured email traffic as it traverses the internet. They then syphon off a duplicate copy of the mail. When the original message successfully arrives at the intended destination, everything seems fine to both the sender and the receiver. However, these parties are unaware that a copy of the message is now in the hands of an unintended recipient who can spy on the conversation. Years ago these attacks were extremely easy to execute because neither senders or receivers ever knew someone was spying on them and encryption wasn't commonly deployed.

Encryption, in general, is a way of providing greater information privacy. This security technology uses cryptography to protect electronic information, meaning that information in a message is encoded by the sender using a complex mathematical algorithm, or key. In order for the information to be understandable by anyone else, specifically the intended recipient, the message must be decoded using that same information. This concept is very commonly applied in multiple IT contexts to secure and protect electronic information, both in transit (when being transmitted from one place to another) and at rest (when sitting in a database or application memory). However, applying the ideal encryption technology in an email communication context is challenging because the technology itself is only one component of a complex communication ecosystem.

Why Does Email Need Encryption in Transit?

(continued)

Establishing the industry-wide technology framework to support widespread email encryption requires tremendous levels of agreement and coordination. Specifically, it first requires email senders to have access to encryption technology in the mail transfer agent (MTA) that they're using to sending their outbound mail. As an email service provider (ESP) with our own proprietary Hurricane MTA technology, SocketLabs played a leading role introducing new encryption features. While we had already designed the Hurricane MTA to support basic encryption options for both on-premise and cloud-based services, we were able to quickly address the industry's new demand for more robust encryption features. We transitioned all SocketLabs servers to utilize opportunistic TLS encryption in the message delivery process.

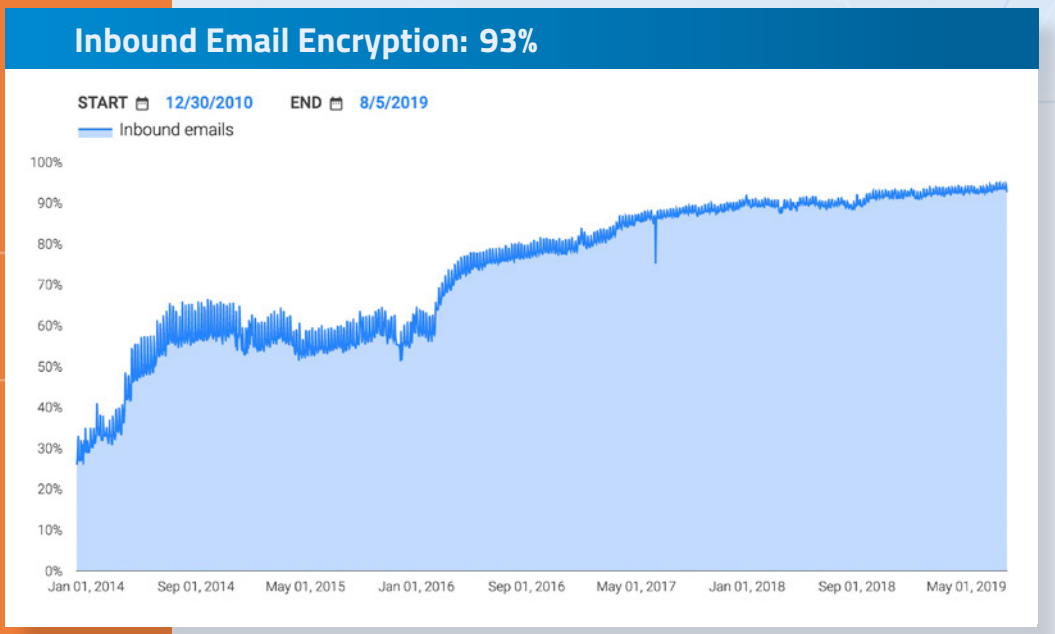
Google, as the largest player in the email industry, plays the most critical role in the adoption of any process or technology. Through their systems they can literally "see" the sending and receiving of most of the world's email traffic. The data they see includes information about which messages are encrypted and which are not. In this position, they are also able to play a key oversight role. Embracing this charge, in 2014 they began publishing a first-of-its-kind transparency report on email encryption in transit. This report measured the level of encryption adoption that Google was seeing across all email traffic. For example, their early statistics showed that 40-50% of mail coming to them was encrypted.

Note: DANE (DNS-Based Authentication of a Named Entities) is an alternate technical approach to requiring the use of encryption in transit for email. However, DANE has not gained the same degree of adoption from mailbox providers as MTA-STS.

Why Does Email Need Encryption in Transit?

(continued)

The Google report also includes specific mention of companies who were not following encryption best practices. By publishing this data for all people to see, they were effectively able to bring attention (and potential embarrassment) to companies who were not supporting encryption. This drove many companies to change their sending policies. Google also introduced user interface features such as the Gmail "lock" icon. The lock icon encouraged the use of encryption by allowing email users to see and know when proper encryption was and wasn't used on the message. Because of these efforts and supporting efforts by other industry players, adoption has moved to 90% of outbound messages and 94% of received. This means that today fewer than 10% of the messages Google sends go to mail systems that don't support encryption.



What does this mean? Essentially, it means that the first step of encryption adoption is now complete because companies can choose to send or receive using encrypted messages if they want to. Most companies now choose to support encryption – especially US companies who have nearly 100% acceptance. But despite this seemingly impressive achievement, pervasive monitoring attacks still happen.

The Problem with "Opportunistic Encryption"

Man-in-the-middle attacks cleverly suppress or 'prevent' encryption from being applied.

Why do man-in-the-middle attacks and other pervasive monitoring email aggressions still occur? The simple answer is that this first wave of encryption that the industry has adopted is "opportunistic encryption" as opposed to "required" or "ensured" encryption. To make the difference between these easier to understand, let's first explain exactly how the current process of opportunistic encryption works…and unfortunately, how it often fails.

Conceptually, the opportunistic encryption process starts with the receiving mail system announcing "We support encryption". The sending system then sees this announcement. It is up to the sending system to try to start the negotiation of an encrypted communication channel. If the receiving system never announces their support for encryption, then the sender will send the message anyway, without trying to encrypt it. So, there's an opportunity for encryption to be applied, but it's not guaranteed.

So, "what's the problem?" you might ask, given that nearly every company has now adopted encryption? First of all, the sender is the one calling the shots. The recipient does not have control, and consequently has no way to ensure that encryption is used. If either party doesn't support encryption, then the entire transaction will be unencrypted by default. Now that adoption is nearly universal, this is less of a problem, but it's still a concern.

However, even if the message sender AND the message receiver support encryption, there's still a huge problem. To illustrate this problem, imagine that a financial services company wanted only to receive encrypted messages. At first, this may seem simple. The sender would see that the receiving system supports encryption and then establish an encrypted connection across which they would send the email message. Even if a would-be spy could intercept the email somehow, they'd be out of luck because they'd have to be able to decrypt the information they'd stolen, right? Wrong. Here's why.

Although it's true that encrypted transmissions can't be read by anyone without the proper key, that's not how man-in-the-middle attacks work. These types of attacks don't "overcome" encryption or "crack the code" to make the message content visible. Rather, they cleverly suppress or "prevent" encryption from ever being applied in the first place. How? Glad you asked …

The Anatomy of a Man-in-the-Middle Attack

To grasp how man-in-the-middle attacks work, you first have to understand the series of invisible communication steps that are involved each time a message is sent. Essentially, you can think of it like a polite intercom conversation between a building guard and a courier who has pressed the doorbell in an attempt to deliver a package. In this analogy, the email sender is like the courier who rings the doorbell and the doorman is like the company receiving the email:

- **Courier (sender):** Rings the bell
- **Doorman (receiver) on the intercom:** "Hello?"
- **Courier (sender):** "Hello, I'm looking for the XYZ Corporation."
- **Doorman (receiver):** "Yes, this is the correct address. Who is this?"
- **Courier (sender):** "I'm a courier. I was sent to this address to deliver a package."

At this point, there are two possible answers the doorman can provide:

Option 1:

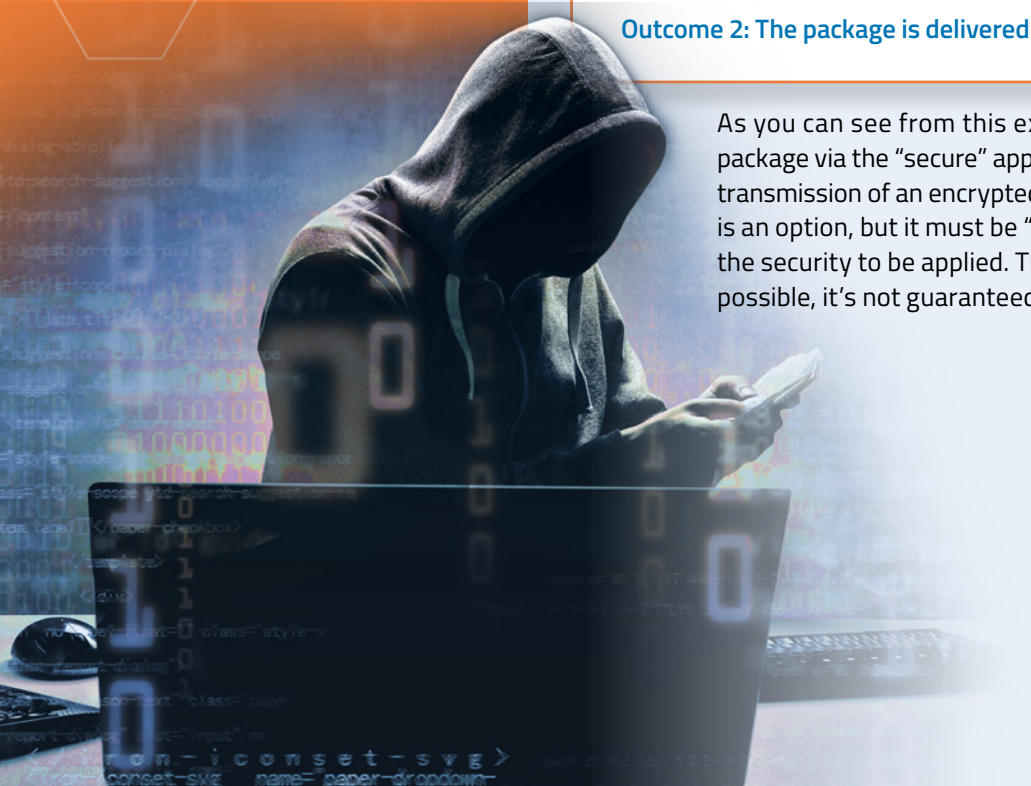
- **Doorman (receiver):** "We support a secure procedure for receiving packages. Just type in the code 123 to open the door, then place the package inside. Thank you."
- **Courier (sender):** "Sure, no problem. I'll follow your instructions and place it in the secure area. Have a nice day!"

Outcome 1: The package is delivered in a secure fashion

Option 2:

- **Doorman (receiver): "**Just place the package outside on the doormat. Someone will get it later."
- **Courier (sender):** "No problem! I'll place it here outside the door. Have a nice day!"

Outcome 2: The package is delivered with no security



As you can see from this exchange, delivering the package via the "secure" approach – which represents the transmission of an encrypted message in an email context – is an option, but it must be "supported" by the receiver for the security to be applied. Therefore, while security is possible, it's not guaranteed.

The Anatomy of a Man-in-the-Middle Attack

(continued)

MTA-STS innovates the communication protocols used to enable encryption, making email safer and more secure.

But there's an even BIGGER shortcoming with this approach: the insecure nature of the initial communication between the sender and the receiver. Specifically, the courier is making the decision on how to deliver the package based on the answer that was given to them over the intercom. But how does the courier really know if this information can be trusted? Was the intercom conversation itself secure? He can't see the person talking to confirm their identity. Nor is there any other means of ensuring that the intercom communication was secure. So, in Option 2, how does the courier really know if the doorman's voice is genuine and if the instruction to just place the package on the doormat is a legitimate reflection of the receiving company's wishes? The bottom line is, there is no way to know. Therefore, because of the method of communication between the sender and the receiver, "the door is left open", so to speak, for nefarious individuals and organizations to intercept that "insecure" communication and CHANGE the instructions that are being fed back to the courier. This opening allows them to fool the courier into leaving the package on the doormat when the company's real preference was to have the package placed into the secure area.

This "open door" is exactly the type of security gap that has been used to steal information from emails through "man-in-the-middle" attacks. They work like this:

1. The sender connects to what they think is the receiving system.
2. The receiver responds back and part of the response includes: "We support encryption."
3. Because this communication is occurring through the very common, but unsecured, email protocol, the bad guys are able to intercept the reply message on its way back to the sender.
4. The bad guys remove the "We support encryption" portion of receiver's response.
5. The sender receives the altered message and now doesn't know that the receiver supports encryption.
6. The sender therefore sends the message to the recipient without encryption.
7. The bad guys now intercept this communication and can see its contents (including any confidential information) because the sender was fooled into NOT encrypting the outbound email.

The man-in-the-middle attacks are successful because they make it appear as though encryption is not supported, when it actually is. To combat this threat, the next major step for email is to innovate the communication protocols used to enable encryption to make it safer and more secure.

The Next Major Step with Email Encryption: MTA-STS

This process applies a layer of security that is not currently common practice.

The first wave of email encryption described above makes use of older technologies and protocols. With industry-wide adoption now basically accomplished, **the industry is embarking on the next big initiative: advancing the underlying technology and communication protocols that enable email encryption.** By adopting new standards for communication, we as industry participants are endeavoring to "change the game" and make man-in-the-middle attacks impossible.

How does MTA-STS improve this? The primary answer lies in the fact that MTA-STS:

1. **Allows a sender to check for encryption support of the receiving system using a different protocol than that of the initial insecure communication channel to start the process of transmitting an email message.**
2. **Allows the receiver to define a clear policy for all senders to follow regarding what to do if they are faced with an email connection that implies encryption is not supported.**

Regarding point number one above, the MTA-STS standard creates a mechanism whereby the sending and receiving parties use the secure HTTPS protocol to converse. It takes advantage of this common, pre-existing system for establishing secure communication between web browsers and web servers. When this technology is used in an email context, the sender and receiver can establish a secure communication channel that is SEPARATE from the channel over which they send the email communications. Here's how it works:

1. The sender queries the recipient's DNS server to find the recipient's policy.
2. The DNS record will indicate that there is a secure website where the security policy can be accessed.
3. The MTA-STS policy is retrieved from the secure website.
4. In the policy, the recipient can define their choices for receiving email such as:
   a. We support email encryption.
   b. Here's instructions on how a sender can validate the encryption method.

In the MTA-STS system, senders and receivers rely on trusted HTTPS technology to ensure that receivers can accept the encryption. This enables a new level of trust that was not previously possible because the initial exchange of information between senders and receivers is safe from interference. Using MTA-STS, the sending and receiving sides can now communicate across the web to understand the supported levels of email communication. By using secure internet connections, this process a) operates outside of email's normal communication bands, and b) applies a layer of security that is not currently common practice. Although the MTA-STS protocol has only been in existence for a short period of time (since November 2018), now that it has been adopted by Google, **this approach is expected to be widely embraced as an accepted**

The Next Major Step with Email Encryption: MTA-STS

(continued)

standard in the email security ecosystem. Google announced in April of 2019 that they were the first major email provider to follow the new SMTP MTA Strict Transport Security (MTA-STS) RFC 8461 and SMTP TLS Reporting RFC 8460 internet standards. This announcement means that they're leading the charge in adoption and are hoping that others will follow. For now, Google has defined a relaxed policy, meaning that they will allow BOTH encrypted and unencrypted traffic. It's essential that they do this in the short term since none of the sending organization can support the protocol – yet.

Given its significant benefits, adoption of the MTA-STS standard is likely to follow a pattern similar to that of opportunistic encryption. As more and more companies define MTA-STS policies, Google and the fraternity of other mailbox providers (who typically follow Google's lead) will likely define and enforce a more-strict MTA-STA policy. This advanced policy will likely require the rejection of any incoming mail that cannot be encrypted. So, while in the short-term MTA-STA will "allow" better security – and will likely be embraced by financial services, healthcare, and other compliance-focused industries – in the future this improved level of security will be a requirement respected and enforced across the industry.

What Steps Should Senders Take to Adopt MTA-STS?

There are several important steps that senders should take to embrace the new MTA-STS encryption standard.

1. Publish a corporate MTA-STS security policy on your website. (SocketLabs has created an MTA-STS policy verificaiton tool that can provide assurance that your policy is set up correctly.)
2. Establish the technical ability to "look up" and "honor" the MTA-STS policies of other organizations. This will allow the sender to be "policy sensitive" and capable of reacting appropriately to the wishes of a receiving mailbox provider.

SocketLabs can assist our customers will both steps listed above. First, for those that need to establish an MTA-STS policy, our team can help you properly choose, establish, and configure your policy setting so that the world can clearly see that encryption is important to you. Second, our cloud-based email we are the first email services provider to implement MTA-STS support into our market-leading MTA product, the Hurricane MTA. As such **we are prepared to help our clients adopt this new standard and begin protecting their partner and customer relationships with encrypted email communications.**

About SocketLabs

SocketLabs is a B2B technology firm that provides flexible SaaS and on-premises solutions for solving a variety of complex email delivery challenges for both transactional and marketing messages. We are a pioneer in the Email Service Provider (ESP) market with a decade-long track record of excellence. Our unique, proprietary mail transfer agent (MTA) technology is trusted by clients around the globe who invigorate their SaaS platforms, mobile apps, and custom applications by "plugging in" to an unmatched email experience. Our founders have been creating cutting-edge email solutions for over 20 years and have built a customer support organization that considers "responsiveness and satisfaction" as our key performance objectives.

Email us!

support@socketlabs.com

Call us!

USA:
800.650.1639
International:
484.418.1285

Chat with us!

www.socketlabs.com/chat