A Study of SMTP [in]Security

Ian Foster, Jon Larson, and Max Masich

Abstract

The Simple Mail Transfer Protocol (SMTP) and related Extended SMTP (ESMTP) are the primary means of delivering email messages between servers over the internet today. Internet traffic can easily be collected by third-parties, yet encryption of SMTP messages is not universal, and in fact cannot be required by a Mail Exchange (MX) server. This means that it is possible for an inter-domain email to be transferred and/or stored as plain-text at at least one point on its path across the internet. This paper provides an analysis of current email providers and their support for TLS encryption over SMTP. We show that while a majority of SMTP servers do provide support for TLS, almost half of all email users use a provider that does not support TLS encryption according to the standard ESMTP protocol. We further show that of those email providers that ostensibly support TLS, a number of them are configured such that their true security is suspect (e.g. invalid certificates, weak ciphers). Our aim is to raise awareness of the vulnerabilities present in the current SMTP environment on the internet.

1 Introduction

Every day, billions of emails are sent across the internet using the Simple Mail Transfer Protocol (SMTP). When it was originally devised in 1982, SMTP did not make any specifications regarding encryption and security – these details were only added many years later, as part of Extended SMTP (ESMTP). Yet even today, encryption is not required for sending emails, and so an email message may appear in plain-text somewhere in its travels.

We wanted to determine how likely it is that email communication between two users is protected against possible eavesdropping. Recently it has been brought to the publics attention that in addition to many solitary malicious users, there are large organisations performing mass surveillance of online communications. Many users rely on email for online communication, sometimes including for sensitive data such as bank information and passwords. In order to protect user's email communication encryption must be used. We set out to study how well email encryption is implemented and how many users it protects.

2 Background

As the ARPANET grew throughout the 1970s, researchers started designing ways of sending messages to other users on remote hosts, using already standardized protocols such as FTP. By 1979, there were already a myriad of incompatible standards, and researchers began to devise a unified standard for transferring mail (RFC 808). In 1981, the Simple Mail Transfer protocol was first proposed in RFC 788, and later standardized in RFC 821. However, these early versions of SMTP did not include any support for encryption[6]. It wasn't until 1995 that Extended SMTP (ESMTP) was standardized, which allowed for arbitrary extensions to the base SMTP protocol, including the support of TLS for SMTP (RFC 2487). Even with the addition of TLS support, SMTP servers are still not allowed to require encryption so that they can be backward-compatible[7].

Today, the basic process of sending an email is the following: the sending client submits a message to a Mail Submission Agent (MSA), which sends the message to a Mail Transfer Agent (MTA). The MTA issues a DNS query to get the Mail Exchange (MX) records of the destination address, and sends the message to this server. From there, the message is relayed (possibly using multiple MX record queries and forwards) to a Mail Delivery Agent (MDA). Finally, the MDA delivers the message to the destination client. It is important to note that both submission and delivery (Sender to MSA and MDA to Recipient) may use a number of different protocols, including HTTP and SMTP (on port 587) for submission and POP3 and IMAP for delivery. However, the transfer from MTA to MX server is always done using SMTP on port 25.

3 Related Works

From what we could find, there has been no similar work done specifically studying the state of security of the SMTP ecosystem.

On the other hand, there have been studies of a similar vein on the state HTTPS. One in particular that we drew inspiration from was [2] in our analysis of SMTP TLS certificate validity.

4 Methodology

4.1 Collecting User Distributions

In order to determine user distribution of popular email providers, we acquired user leaks from Sony (from 2011, having 20,000 records)[5], Gawker (from 2010, having 500,000 records)[4], and Adobe (from 2013, having 141 million records)[3]. From these leaks we would get a good idea of what mail providers were popular, and the approximate user distributions among them.

It should be noted that it is difficult to accurately assess the distribution of users among all email providers. One concern of ours was that our three data sources may be biased towards a U.S. users distribution. However, due to the large dataset (approximately 140 million records), the presence of five foreign email providers in the top ten email providers, and the relatively recent acquisition of the list of emails, we feel that the data from the Adobe leak provides a reasonably fair portrayal of current user distribution. The results from the Gawker leak probably skew even more to American users and the list is a little more dated, but the large sample size (admittedly much smaller than Adobe) nevertheless made it a useful tool for getting an idea of user distribution. Results from the Sony leak did not prove to be useful, mainly due to its small size.

4.2 Scanning SMTP Servers

Lists of domains were collected from the above mentioned user leaks, The Alexa Top Million Sites Index[1], and some of the DNS root zone files. For each domain we performed a DNS query for its MX records, and then for all of the MX records IP addresses. If no MX records are found for a domain we use the domain as its own MX record.. (For the purposes of this study we only collected IPv4 A records due to the fact that IPv6 AAAA adoption for email is minimal and our testing machine did not have IPv6 connectivity.)

We faced several issues when setting up our scanning service. First, we had to avoid being rate limited by DNS servers as we collected MX records of domains we were examining. We attempted to set up our own DNS server, but were thwarted in our efforts when we realized our machine did not have the necessary RAM. Instead, used a round robin approach using publicly available DNS servers.

We also found that some SMTP servers would not respond to our simple EHLO commands, including some of the top email providers (e.g. gmx.com). We found that in some cases, our initial timeout value of five seconds was too short to establish a socket connection. Raising that timeout to 10 seconds allowed us to connect many more hosts. In other cases, we found that not having a valid identifier for the EHLO command resulted in the host terminating the connection. Changing the identifier to a valid domain resolved this issue.

For every domains mail servers IPs we collected the following information:

- ESMTP Support Supported if the server responds with 220 to an EHLO command
- TLS Support
  - Supported if the server responds with 220 to a STARTTLS commands
- SSL Cipher Used
- SSL Cipher Bits
- The SSL Certificate provided by the server
  - Discovered by opening a secure socket connection to the server

Due to much of the scanning process being slowed down waiting for servers to respond due to network latency, we parallelized the SMTP scanning process. When running with 128 parallel requests we were able collect data for an average of 40 domains per second. All data collected was put into a relational sqlite3 database for analysis. Our SMTP scanning program is open source and freely available on GitHub[8].

4.3 Collecting Email SMTP Headers

After collecting security data on the SMTP servers of one million email providers and domain names, we attempted to determine what security measures were actually implemented when sending inter-domain emails among the top email providers. To this end, we acquired accounts at seven of the top providers (outlook.com, gmail.com, yahoo.com, aol.com, web.de, gmx.de, mail.ru). We had hoped to sign up for more accounts, but met barriers both financial (e.g. needing to pay for internet service for a comcast.net account) and geographical (e.g. needing a Chinese cell phone number to sign up for a qq.com account).

We then sent an email from each account to each of the other accounts and examined the SMTP headers, in particular the Received fields. These fields are added to the header upon receipt at each server, showing the sender and receiver as well as the protocol and security measures used in the transfer. For example, the following field from the header of an email sent from gmx.de to aol.com shows that TLSv1 encryption was used:

```
Received: from mout.gmx.net (mout.
    gmx.net [212.227.15.19])
  (using TLSv1 with cipher DHE-RSA
      -AES128-SHA (128/128 bits))
  (No client certificate requested
      )
  by mtain-dk12.r1000.mx.aol.com (
      Internet Inbound) with ESMTPS
  id 264DF38000098 for <
      username@aol.com>; Tue, 18
      Mar 2014 20:58:36 -0400 (EDT)
```

Based on these headers, we could see which pairs of providers use encryption when sending inter-domain emails.

5 Results

5.1 Top Providers

Analyzing the number of occurrences of each domain name in the Adobe and Gawker email leaks gave us an idea of how many users use each provider. The ordering and the exact numbers of the top domains differ somewhat, but there is some agreement between them. For example, six of the top ten domains are the same for both sources (hotmail.com, gmail.com, yahoo.com, aol.com, comcast.net, and mail.ru). For each of the top providers, we ran our scanner as outline above. The results are summarized in the tables 1 and 2.

Table 1: Adobe Leak

Domain	% Users	ESMTP	TLS
hotmail.com	>21.36	Yes	No
gmail.com	>15.76	Yes	Yes
yahoo.com	>11.69	Yes	Yes
aol.com	2.28	Yes	Yes
comcast.net	0.82	Yes	No
mail.ru	0.82	Yes	No
web.de	0.8	Yes	Yes
qq.com	0.63	Yes	No
gmx.de	0.63	Yes	Yes
naver.com	0.43	Yes	No

Table 2: Gawker Leak

Domain	% Users	ESMTP	TLS
gmail.com	>32.24	Yes	Yes
yahoo.com	>20.46	Yes	Yes
hotmail.com	>15.66	Yes	No
aol.com	3.81	Yes	Yes
comcast.net	1.5	Yes	No
mac.com	1.08	Yes	No
verizon.net	0.47	Yes	No
cox.net	0.41	Yes	No
earthlink.net	0.38	Yes	No
mail.ru	0.3	Yes	No
naver.com	0.43	Yes	No

The two most striking results from these tables are that the majority of users (>60%) get service from just twenty email providers, and that at least half of the top email providers dont provide any support for TLS when receiving emails. The biggest offender appears to be hotmail.com, which also hosts the MX servers for a number of other very large domains (e.g. hotmail.fr, outlook.com, msn.com, live.com).

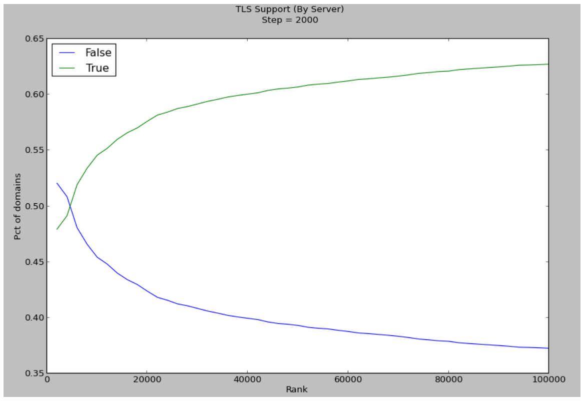


Figure 1: TLS Support by Server

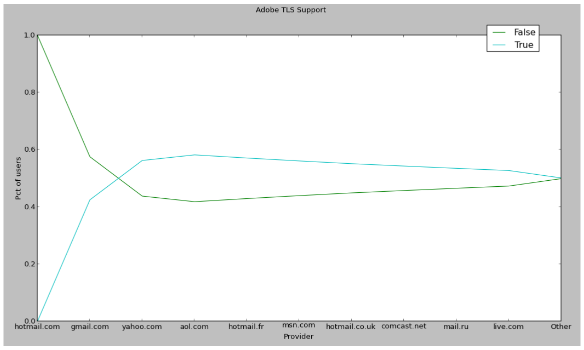


Figure 2: Adobe TLS Support

5.2 TLS Support

We first looked at just the proportion of email providers that support the STARTTLS command, ranked in order of number of users (as found in the Adobe leak). We already know from table 1 that of the top 10, only 50% support TLS. As can be seen in figure 1, of the top 2000 providers, only 47.94% support TLS. We were surprised to see, however, that support for TLS actually increases as we looked at more and more providers, such that of the top 20,000 providers, 57.63% supported TLS, and at 100,000 the number leveled off around 62.75%. Were not entirely sure why we see this result, but one possible explanation is that many smaller, less-used email providers use Google mail servers, which support TLS, instead of setting up their own.

When we look at the proportion of actual users that have TLS support, however, the numbers are not quite as positive. Looking at just the top 10 providers, only 52.72% of users have a provider that supports TLS, and looking at our whole dataset that number goes down to 50.13%. That would suggest that of all the email users (at least, the ones who register with Adobe), only half receive emails with

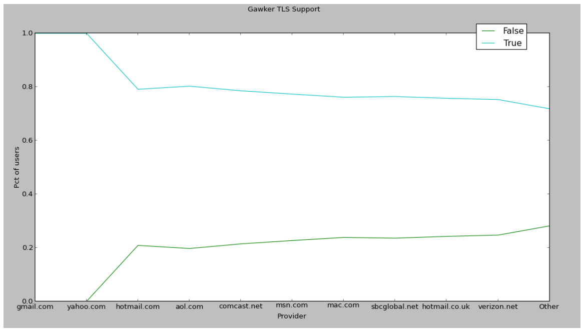


Figure 3: Gawker TLS Support

any encryption. The numbers derived from the Gawker leak are considerably better (75.28% for top 10, 71.84% support for all providers) but again, we consider this source to be less representative of an international user base and more of an American (or at least Anglophone) user base. In either case, this still implies that at most about 72% of email users receive encrypted emails.

5.3 Valid Encryption

Based on the results of the previous section, we know that about 62% of domains (covering between 50% and 72% of users) ostensibly support TLS, but we wanted to determine if those connections were really secure. To do this, we opened secure connections to the SMTP servers and collected security certificates and cipher information. Of the providers that support TLS, we found that only 75.9% of them have valid certificates (i.e. they were signed by a trusted CA). However, when looking at the proportion of users who use those providers, the numbers are much more positive: according to the Adobe leak, 98% of users that use TLS have valid certificates, and according to the Gawker leak, 99.44% use valid certificates.

We also examined the types of ciphers being used by providers who support TLS to see if many were using algorithms with known weaknesses. We found eleven ciphers used by the servers in our study; figure 8. The only ciphers found that have known weaknesses are RC4-MD5 and RC4-SHA, together they are used by less than 0.06% of the servers in this study.

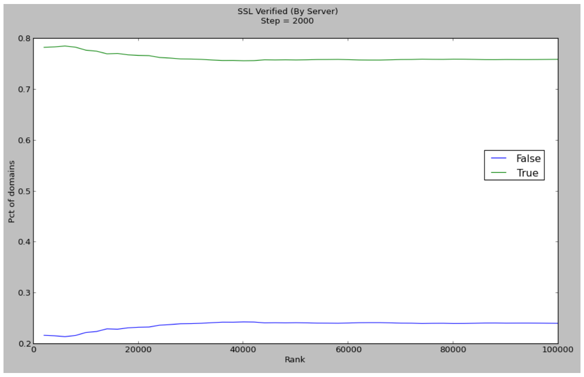


Figure 4: Valid SSL Cert by Server

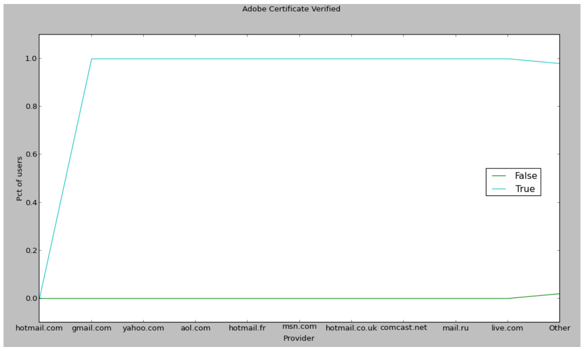


Figure 5: Adobe Valid Cert

5.4 Overall Findings

When looking at the ESMTP, TLS, and certificates use among all of the top million email providers Figure 9 we find that almost all providers support ESMTP with an adoption rate of 99.29%. TLS support is only found on just over half of the providers at 54.58%. Only 42.45% of all providers, or 77.84% of the providers with TLS, have a valid SSL certificate.

5.5 Analysis of SMTP Headers

After determining the state of security for one million email domains, we examined what effect that actually has when sending emails between domains by tracing the Received field of SMTP headers. The results can be seen for seven of

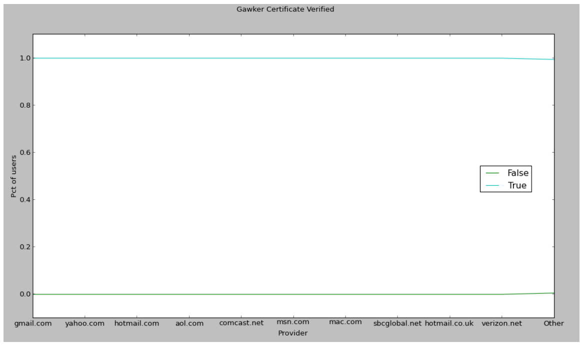


Figure 6: Gawker Valid Cert

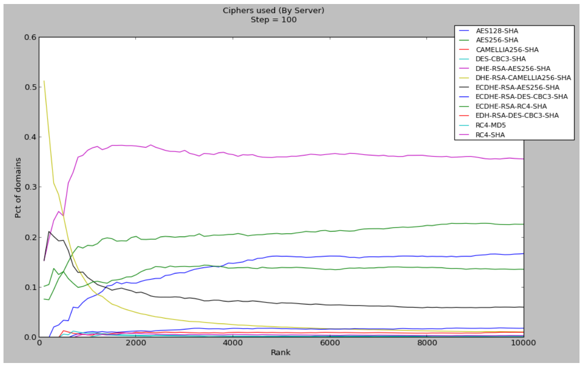


Figure 7: Cipher by Server

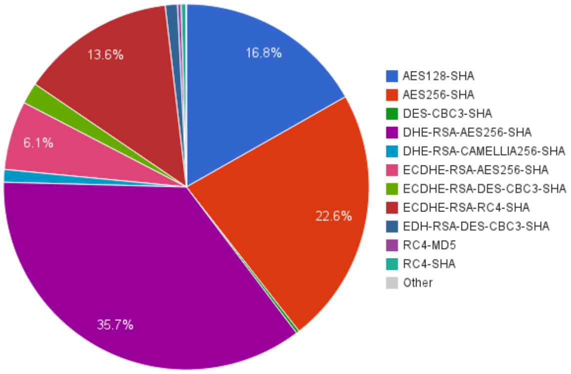


Figure 8: Cipher Pie (Yum!)

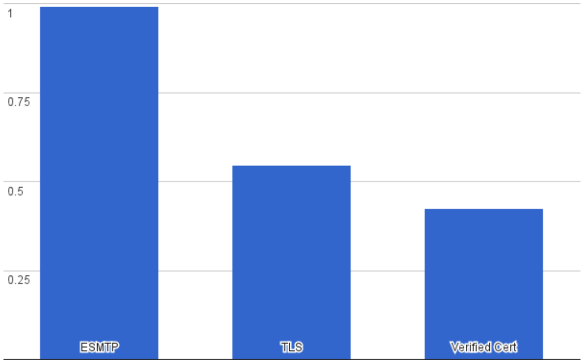


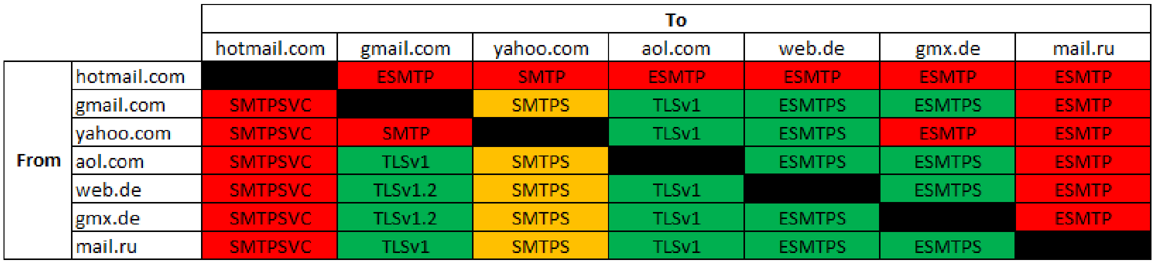
Figure 9: TLS Adoption by Domain

the top ten email providers in Figure 10, which shows the sender and receiver domains along with the protocol/encryption information for the inter-domain hop of the message. Of those seven, all but hotmail.com and mail.ru support TLS.

The most obvious result is that the domains that do not support the STARTTLS command do not receive messages with encryption, which was to be expected. A somewhat surprising result was that hotmail.com, which does not support TLS on incoming emails, also does not use TLS on its outgoing emails. However, mail.ru uses the encryption settings of the destination server, even though it does not use encryption on incoming emails. An even more surprising discovery was that yahoo.com did not use encryption when sending to gmail.com or gmx.de, even though all three domains support TLS.

We were somewhat unsure as to the level of encryption provided on emails bound for yahoo.com. Headers with explicit cipher information were obviously sent with encryption, and IANA defines the ESMTPS parameter as indicating ESMTP with STARTTLS[9]. We assume that SMTPS similarly indicates that STARTTLS (or some other encryption mechanism) was used, but there is no official SMTPS parameter.

The implication of these results is that, while approximately 60% of domains support TLS on incoming messages, not all MTAs actually make use of it when sending to those servers. It is



		To						
		hotmail.com	gmail.com	yahoo.com	aol.com	web.de	gmx.de	mail.ru
From	hotmail.com	■	🟥 ESMTP	🟥 SMTP	🟥 ESMTP	🟥 ESMTP	🟥 ESMTP	🟥 ESMTP
	gmail.com	🟥 SMTPSVC	■	🟨 SMTPS	🟩 TLSv1	🟩 ESMTPS	🟩 ESMTPS	🟥 ESMTP
	yahoo.com	🟥 SMTPSVC	🟥 SMTP	■	🟩 TLSv1	🟩 ESMTPS	🟥 ESMTP	🟥 ESMTP
	aol.com	🟥 SMTPSVC	🟩 TLSv1	🟨 SMTPS	■	🟩 ESMTPS	🟩 ESMTPS	🟥 ESMTP
	web.de	🟥 SMTPSVC	🟩 TLSv1.2	🟨 SMTPS	🟩 TLSv1	■	🟩 ESMTPS	🟥 ESMTP
	gmx.de	🟥 SMTPSVC	🟩 TLSv1.2	🟨 SMTPS	🟩 TLSv1	🟩 ESMTPS	■	🟥 ESMTP
	mail.ru	🟥 SMTPSVC	🟩 TLSv1	🟨 SMTPS	🟩 TLSv1	🟩 ESMTPS	🟩 ESMTPS	■

Figure 10: SMTP Headers

Table 3: Top Mail Servers by Domain

Provider	% of Domains
Google	11.03%
Microsoft	3.45%
GoDaddy	2.06%
MX Logic	1.58%
RackSpace	1.32%

impossible to tell just what percent of email traffic is unencrypted without looking at the actual numbers of emails moving between different domains. However, if only 50% of users are on providers that support TLS on incoming messages and not all domains actually make use of that encryption when sending to those providers, it is reasonable to assume that at most 50% of users are actively using encryption, and the number is probably somewhat lower.

5.6 Outsourced Mail

Many of the top domains actually share a common set of SMTP servers. This is due to providers such as Google and Microsoft who host email for domains owned by other companies. When we looked at who hosted the email of the top million providers we saw that the majority of email was being hosted by the five companies listed in table 3. Here we see that Google hosts the most with 11% of the domains email. The values listed in table 3 represent the lower bound of the amount of domains each provider is in control of.

6 Future Work

Due to the time constraints of this project only a small sample of a few million domains, a larger sample of the entire internet, especially if ranked by region could provide good insight into that state of secure email communication on an country by country basis.

A follow up study collecting the same data would allow us to see how the adoption of TLS for SMTP is changing.

In addition to TLS, SMTP can also make use of Sender Policy Framework (SPF) records for authentication of email origin, and DomainKeys Identified Mail (DKIM) to verify message integrity. Our study could benefit further by checking for the usage of SPF and DKIM use coverage.

Determine reason for Microsoft not supporting TLS. In our preliminary investigation, it was unclear why they would not even support the option for other services to send with TLS. Ideally, we would like to get in touch of the hotmail team to see if we can get an answer to this question directly.

Explore alternatives to SMTP for email communication. An ideal mail protocol would provide end-to-end email encryption. Not only would such a protocol provide stronger data security and integrity, but it could also have the potential for alleviating spam.

Many online services send automated emails that may contain sensitive information such as clear text passwords, password reset links, order information, etc, which may not be sent via the same route as non-automated email from that domain. An analysis of automated email headers from popular providers would give us insight into this issue.

7 Conclusion

We have shown that a significant percentage of email providers today do not support TLS for encrypting incoming email traffic. Unfortunately, some of those providers are among the most popular email providers, resulting in nearly half of email users receiving, and possibly sending, emails across the internet in plaintext. Furthermore, we have shown that not all providers use encryption even when the destination domain supports it, resulting in an even smaller percentage of users whose emails are secured. Our hope is that these results will illuminate the need for wider adoption of the current standards for email encryption, and perhaps someday a rethinking of mail transfer protocols.

References

[1] Alexa Top 1,000,000 Sites

`http://www.alexa.com/topsites/global`

[2] Analysis of the HTTPS Certicate Ecosystem

`http://web.eecs.umich.edu/~mibailey/publications/imc13\_https\_final.pdf`

[3] Adobe Email Leak

`http://www.theguardian.com/technology/2013/nov/07/adobe-password-leak-can-check`

[4] Gawker Email Leak

`http://www.slate.com/articles/technology/technology/2010/12/was\_your\_gawker\_password\_hacked.html`

[5] Sony Email Leak

[6] RFC 821

`http://tools.ietf.org/html/rfc821`

[7] RFC 2487

`http://tools.ietf.org/html/rfc2487`

[8] Project Code

`https://github.com/lanrat/smtp-scanner`

[9] Mail Parameters

`http://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml`