Hardenize

Hardenize Policy: Email Infrastructure

PREVIEW 3 (19 May 2021)

Introduction

Hardenize is a network and security configuration monitoring platform designed to help organisations improve their security. Our product is designed to provide deep visibility into our customers' networks using automated discovery and comprehensive analysis. In addition to serving our customers commercially, we're committing to providing our assessment tools to the long tail of small organisations and individuals free of charge on our web site.

Ultimately, we exist to help everyone deploy their systems correctly and securely. Today, after many years of waiting, we have security standards that are fit for purpose, but achieving security turned out to be just as elusive. There are several aspects to this problem and we're working to address the problem at multiple levels.

One big problem is that we now, paradoxically, have too many security standards. Knowing what they are, what purpose they serve, and understanding when they are appropriate to use is all difficult. It's difficult if you have resources to spend on the task, but it's impossible for small organisations.

Providing our assessment tools free of charge was our starting effort, but we now want to go further by providing a set of high-level guides that provide full coverage of a variety of network and security standards. The main goal is to provide—as well as maintain over time—one resource that can be used as guidance when deciding what standards to use, why, and when. Tentatively, we call this resource Hardenize Policy.

We anticipate that the complete first version of Hardenize Policy will consist of four documents, each addressing one well-defined aspect of network and security: 1) Domain name and DNS configuration; 2) Email infrastructure; 3) TLS and PKI; and 4) Web application security.

Email Infrastructure

This document is the first instalment of Hardenize Policy and focuses on email infrastructure. In line with our goals, we provide a concise and high-level overview of how to secure email infrastructure and achieve the following goals:

- Ensure that emails are successfully and securely delivered to you.
- Detect configuration and interoperability issues early so that they can be addressed quickly. Reduce costs of service downtime.
- Protect your email in transit, preventing exposure of sensitive data via pervasive passive monitoring or more advanced active network attacks.

- Prevent malicious third parties from sending email messages under your identity, targeting your customers and business partners. Protect your brand and reputation. Reduce time spent on dealing with email fraud.

We want these documents to be concise and structured in a way to support ease of use. We hope that, at this level of detail, our policies will be an easy read and support quick and efficient knowledge transfer. At the same time, it's clear that we will not be providing here sufficient information to implement these technologies. We plan to provide that separately, via detailed implementation guides. In particular, we're planning detailed configuration guides for SPF, DKIM, and DMARC.

Some topics related to email security are currently outside the scope of this document. We are intentionally not discussing configuration of email receivers or other mechanisms that could be used to protect email, such as PGP/GnuPG and S/MIME. This may change in the future, depending on what feedback we receive.

This document is a preview that we wish to refine in collaboration with a wider community. In addition, because this is the first part of Hardenize Policy, it is also important get the format right, so that we can later replicate the approach in the other documents.

Table of Contents

Configuration Guidance

1. Support STARTTLS

STARTTLS is an SMTP extension that enables encrypted delivery of email messages. Upon connecting to the server, the client issues a request for encryption. If both sides support encryption and successfully perform a TLS handshake, the email is delivered securely.

Why:	STARTTLS is necessary to enable secure (encrypted) delivery of email messages. Without encryption, email messages are delivered as plaintext and exposed to passive monitoring attacks.
Standardisation:	IETF RFC 3207.
Maturity:	Very good. STARTTLS is widely supported by clients and servers.
Effort:	Low.
Risks:	None when STARTTLS is optional. Clients that do not support encryption or have trouble establishing an encrypted channel will fall back to delivering email in plaintext.
Priority:	Very high.

2. Configure TLS appropriately for SMTP

Servers that support STARTTLS use Secure Socket Layers (SSL) and Transport Layer Security (TLS) protocols for encryption. These protocols have a long history and generally provide many features of varying qualities, from old and insecure (e.g., SSL v2), to modern and secure (e.g., TLS 1.3). Just enabling encryption isn't sufficient to guarantee effective security.

Email is traditionally secured using opportunistic encryption, which only provides protection against passive attackers. Because many clients and servers are still willing to send and receive plaintext messages, active network attacks are still possible. An attacker can bypass the security of a very well configured TLS server by intercepting client traffic and pretending to be the server that doesn't support encryption.

There is a separate problem of sometimes having to support weak encryption to interoperate with very old clients. If a server's TLS configuration is strict, clients may fall back to plaintext. Thus, the approach used by many installations is to use minimum security requirements in order to avoid fallback to plaintext.

Advice:

- You should support TLS 1.3.
- You must support TLS 1.2.
- You may support TLS 1.1 and TLS 1.0 for interoperability reasons, but only as fallback.
- SSL 3 should not be supported.
- SSL 2 must not be supported.

Prefer cipher suites and protocols that provide forward security; this is probably the most important aspect of the configuration to get right. It is acceptable to fallback to encryption without forward security, but only if necessary for interoperability.

Use only cipher suites that provide at least 128 bits of encryption. As an exception for interoperability with other servers, you may also use TLS_RSA_WITH_3DES_EDE_CBC_SHA, which provides security of about 112 bits.

Why:	Avoid use of insecure cryptographic primitives, which may be ineffective or could even negatively affect other related systems. Support widely-required primitives to ensure that all or most email is received encrypted. Demonstrate good security hygiene. Adopt modern encryption to prepare for deployment of more advanced features that require it.
Standardisation:	N/A
Maturity:	Very good. TLS 1.3 is relatively new and may not be universally supported, but this protocol version is ultimately not necessary.
Effort:	Moderate. Configuring TLS 1.2 and earlier protocol versions can be daunting due to the sheer number of options, many of which are insecure. However, we maintain a comprehensive, up to date, and easy to use configuration advice.
Risks:	Low. Clients that fail to negotiate an encrypted connection are likely to attempt delivery without encryption.
Priority:	High. Very high if using MTA-STS or DANE.

3. Use valid TLS certificates issued by public CAs

TLS connections are only secure if the connecting side correctly validates the provided certificate during the handshake phase. Email senders traditionally do not verify server certificates, even though this is required by RFC 7817. STARTTLS with any certificate (even if it's invalid) is sufficient to prevent passive attacks (e.g., email interception and recording), but doesn't prevent active network attacks. When validation is not performed, active network attackers can pose as the target server to collect its email.

Deploy your SMTP servers with valid certificates issued by public CAs. A certificate is valid if it's matching the server hostname, hasn't expired, hasn't been revoked, and is accompanied by a valid certificate chain. This is not only good discipline, but is also a necessary step for further security measures such as DANE and MTA-STS.

Why:	Ensure interoperability with clients that check certificate validity. Build a foundation for stronger protection measures using DANE and MTA-STS.
Standardisation:	N/A
Maturity:	Very good.

Effort:	Moderate. Ongoing effort is required to replace certificates before they expire. However, with the current drive to automated certificate issuance, this task is likely to get easier over time.
Risks:	None.
Priority:	Moderate. Very high if using DANE or MTA-STS.

4. Keep detailed SMTP server connection logs

Decisions to improve security are sometimes avoided or postponed because there is often insufficient visibility into how exactly protocols are used. For example, a change to require STARTTLS on your SMTP servers may lead to some emails not being delivered. Keeping comprehensive logs of the activity makes this type of decision easier because you can at any point of time check if any emails are delivered via plaintext.

Why:	Achieve visibility into capabilities of client systems connecting to your servers, which will enable you to detect problems early and make changes confident that they wouldn't cause interoperability issues.
Standardisation:	N/A
Maturity:	Good.
Effort:	Moderate.
Risks:	None.
Priority:	Moderate.

5. Consider requiring STARTTLS

Email servers are typically configured to accept unencrypted email. This is not great, as it increases chances that sensitive information will be intercepted in transit and recorded. When STARTTLS is required, emails are not exposed if secure transport is not possible.

Why:	Prevent any emails from being passively recorded.
Standardisation:	N/A
Maturity:	Good.
Effort:	Low
Risks:	Moderate to Low. As of December 2020, Google reports about 6% of their inbound email and about 15% of their outbound email is not encrypted. If servers are configured to require STARTTLS, some messages may be lost. For best results, use logging over an extended period of time to determine if this is a real risk for you.

Priority:	Moderate.

6. Use DMARC reporting

DMARC reporting provides insight and visibility into who is sending email on your behalf. It is the necessary first step involved with every DMARC deployment, but works just as well on its own.

Why:	Build inventory of systems sending email on your behalf. Achieve visibility into phishing and impersonation attacks. After DMARC is deployed, detect misconfiguration early.
Standardisation:	IETF RFC 7489.
Maturity:	Very good. DMARC reporting can be activated with a simple DNS change. There is a wide range of third-party providers that can be used to receive and analyse the reports.
Effort:	Low.
Risks:	Low. Forensic emails could contain sensitive information; if they are used, care must be taken to restrict who has access to the reports.
Priority:	High. Very High if DMARC is deployed.

7. Use SMTP TLS Reporting

SMTP TLS Reporting (TLS-RPT) is a reporting mechanism that can be used to monitor for failures involved in email delivery. It is designed to detect interoperability issues as well as active network attacks against email delivery.

Why:	Achieve visibility into any problems related to the operation of your SMTP servers, including detection of active network attacks.
Standardisation:	IETF RFC 8460.
Maturity:	Low. SMTP TLS Reporting is a young standard that is not yet widely used.
Effort:	Low.
Risks:	None.
Priority:	Medium. High if using DANE or MTA-STS.

8. Use SPF

Sender Policy Framework (SPF) is a security mechanism that can be used to authenticate systems that send email. The usual approach is to publish the list of IP addresses that are authorised. SPF works best when used with the infrastructure you control directly because it's configured entirely at the DNS level. A referral mechanism is used when working with third parties, although care must be taken to stay under the limit of 10 DNS lookups. SPF is insufficient on its own; for best results, use in combination with DKIM and as part of a DMARC deployment.

Why:	Indicate which systems are allowed to send email on your behalf, reducing impersonation attacks against your domain names.
Standardisation:	IETF RFC 7208.
Maturity:	Very good.
Effort:	Moderate. SPF requires an accurate inventory of servers authorised to send email and an ongoing effort to keep the DNS configuration in sync.
Risks:	Low. Email sent by servers that are not authorised may be rejected. DMARC reporting can help detect such situations.
Priority:	High.

9. Use DKIM

DomainKeys Identified Mail (DKIM) is a security mechanism that can be used to authenticate email messages and ensure they have been sent by the domain owner. This is achieved using public cryptography; email messages are accompanied by digital signatures that receivers are able to verify. Compared to SPF, DKIM requires more effort to configure because each sender must be configured with a separate encryption key. On the other hand, DKIM is less fragile and easier to use with third parties. Additionally, it supports forwarding. DKIM is insufficient on its own; for best results, use in combination with SPF and as part of a DMARC deployment.

Why:	Support authentication of your email messages, reducing impersonation attacks against your domain names.
Standardisation:	IETF RFC 6376.
Maturity:	Very good.
Effort:	High. DKIM requires an accurate inventory of servers authorised to send email. Each server must be configured with a valid and endorsed encryption key. The keys need to be periodically rotated.
Risks:	Low. Email sent by servers that are not authorised may be rejected. DMARC reporting can help detect such situations.
Priority:	High.

10. Use DMARC to quarantine or reject spoofed email

Domain-based Message Authentication, Reporting, and Conformance (DMARC) is a mechanism that enables domain owners to communicate policies and preferences for email message validation. DMARC builds on, improves, and unifies SPF and DKIM standards and provides reporting functionality.

DMARC is best used to protect transactional emails, for example official company communications. Enforcing strong email authentication is in conflict with the widespread practice of email forwarding; email messages are often forwarded from one mailbox to another, or distributed via mailing lists. Even though there is no malice involved, such messages are effectively being spoofed and detected as such by DMARC. In particular, DMARC doesn’t work well for email providers providing mailboxes to the general public. They may choose not to enforce a DMARC policy in avoid email loss due to forwarding. However, in that case they also forego its protection mechanisms. A new standard called ARC is being developed to address these shortcomings in the future.

When the "quarantine" policy is used, messages that fail DMARC validation will be flagged as spam, which means that they can be recovered. Organizations that require very strict security should consider using the "reject" policy, which leads to deletion of failed email messages.

Why:	Prevent phishing and impersonation attacks against your brands.
Standardisation:	IETF RFC 7489.
Maturity:	Very good.
Effort:	Moderate. Enforcement of email message validation requires certain operational proficiency. After DMARC is enabled, every email sent must be authenticated using either SPF or DKIM. The recommended approach is to start with only DMARC reporting and no policy enforcement, then address any discovered issues, after which an appropriate policy can be activated.
Risks:	Moderate. If organization email messages are not correctly authenticated due to misconfiguration, they may be seen as invalid and rejected. Some messages may be lost due to email forwarding.
Priority:	High. Start with only reporting enabled, then gradually advance to quarantine and, optionally, reject.

11. Use DANE if using DNSSEC

DNS-Based Authentication of Named Entities (DANE) is a protocol that builds on DNSSEC to enable strong authentication of internet services. Even when DANE is not deployed, DNSSEC by itself adds a layer of protection because it prevents spoofing of MX DNS records. With DANE, the DNS configuration and TLS certificates are checked in tandem and together ensure that active network attacks are not possible. DANE is not foolproof, given that it works only when both the sending and receiving sides support it.

Why:	Prevent active network attacks against email message delivery.

Standardisation:	IETF RFC 7671.
Maturity:	Moderate to Low. There is strong resistance to DNSSEC, which affects the adoption of DANE. Even though the overall adoption is not great, DANE is required by several European governments.
Effort:	High. DNSSEC and DANE are complex standards. After DANE is deployed, only certificates that match the DANE configuration can be used. Additionally, changes to the certificates must be kept in sync with the DANE configuration, which is a largely manual task given that mature tooling is not available.
Risks:	Moderate. The additional complexity of DNSSEC and DANE increases the likelihood of configuration problems, possibly leading to delayed or lost email.
Priority:	Moderate if using DNSSEC, Low otherwise. DANE will not defeat all active network attacks, but it can close the security gap as part of a comprehensive defense strategy.

12. Use MTA-STS

MTA Strict Transport Security (MTA-STS) a security mechanism that enables domain owners to secure email delivery by protecting the MX records, enforcing strict certificate validation, and modern encryption standards. MTA-STS is conceptually similar to its predecessor and HTTP counterpart, HTTP Strict Transport Security. It was designed for use by organizations who didn't wish to deploy DNSSEC, but wanted to improve email security. MTA-STS is not foolproof, given that it works only when both the sending and receiving sides support it.

Why:	Prevent active network attacks against email message delivery.
Standardisation:	IETF RFC 8461.
Maturity:	Low. MTA-STS is very young; it’s not yet supported by most MTAs. Notably, it may be supported by some large providers, for example Google.
Effort:	Moderate. MTA-STS policies are indicated via DNS and delivered via HTTPS, which means that another endpoint needs to be maintained. Changes to the MX records must be kept with the MTA-STS at all times. SMTP servers must deploy with well-configured TLS and valid certificates.
Risks:	Moderate. The additional complexity of MTA-STS increases the likelihood of configuration problems, possibly leading to delayed or lost email.
Priority:	Low. MTA-STS will not defeat all active network attacks, but it can close the security gap as part of a comprehensive defense strategy.

13. Monitor for look-alike domain names

Protecting the domain names you own from email spoofing attacks is necessary, but not entirely sufficient for a robust defense. There are other ways in which your organization may be attacked. For example, someone could create a look-alike domain name or even a subdomain on an entirely unrelated domain name. Even though the name may not exactly match yours, the similarity may be sufficient enough to confuse email recipients into believing the messages came from you. Monitoring for look-alike domain names must be done via a third-party service provider who has full visibility of global infrastructure and can help identify domain names that are similar to yours.

Why:	Detect third parties that operate domains and subdomains that are similar to yours and may pose a threat to your operations.
Standardisation:	N/A
Maturity:	N/A
Effort:	Low to High, depending on the existence threats. Achieving visibility is straightforward with a good service provider, but determining which threats are real and pose danger may require manual work. Taking down the threats is very involved.
Risks:	None
Priority:	High

14. Be aware of BIMI

Brand Indicator for Message Identification (BIMI) is a very recent initiative that aims to create a verifiable connection between a domain name and a brand. The initial use case for BIMI is display of company logos in email clients. Companies that wish to use BIMI need to have their logo certified (e.g., by obtaining a Verifiable Mark Certificate from a CA that supports them), after which the image can be placed on a web server and announced via a DNS record. BIMI requires a robust DMARC configuration with policy set to "quarantine" or "reject".

BIMI doesn't provide a lot of security value by itself, but it does build on DMARC and it can be used as a lever internally to accelerate DMARC deployment. BIMI's main value is for brand promotion (your messages stand out in the users' mailboxes). Over time, BIMI may assist in helping phishing, if the standard becomes widespread and users learn to use it to distinguish legitimate messages from those that are not.

Why:	Promote your brand and provide assurance to your users. In the long term, if enough organizations adopt BIMI, users may be better-equipped to detect spoofed (phishing) emails.
Standardisation:	In progress.
Maturity:	N/A
Effort:	Low.
Risks:	None.
Priority:	Low.

Acknowledgments

This document is a result of our own experiences and research, as well as a careful study of the technologies, practical considerations, and lessons learned published by others. In other words, we stand on the shoulders of giants. In this early version, the acknowledgements section is largely a placeholder. We're planning to seek wider input from experts in their fields, to further improve the content here.

The format of this document has been greatly influenced by the .trust Technical Policy, which was first released in 2014 by NCC. As they say, less is sometimes more, and we still remember how our first sight of their work led to clarity of many topics. We hope to replicate that experience with our work. Our policy covers essentially the same ground and has the same goal as NCC's, but we approach the problem from a different perspective. The biggest difference is that we are not prescriptive. We don't think that there can be one policy that works for everyone, which is why we prefer to think in terms of frameworks that can be adapted to the particular circumstances. We may explore evolving the policy with maturity levels in the future. Another difference is that we wish to make these documents even simpler to support more efficient decision making. Because we can't magically solve all the complexity issues, we hope to address them in detailed implementation guides.