draft-ietf-uta-mta-sts-14 Discussion

IETF 101

Major Issues (1)

(Alexey) 3.2. MTA-STS Policies

> The [RFC7231] "Content-Type" media type for this resource MUST be "text/plain". When fetching a policy, senders SHOULD validate that the media type is "text/plain" to guard against cases where webservers allow untrusted users to host non-text content (typically, HTML or images) at a user-defined path. Additional "Content-Type" parameters are ignored.

I find this requirement to be problematic, because if somebody decided to use a file in ShiftJIS charset or one of Unicode-16 variants, it would not be parseable at all.

My recommendations: update this requirement to say that all parameters other than charset are ignored. Additionally, require use of charset=utf-8 or charset=us-ascii

Major Issues (2)

(Alexey) Section 3.2:

```
sts-policy-record        = *WSP sts-policy-field *WSP
                           *(CRLF *WSP sts-policy-field *WSP)
```

I thought the intent of this syntax was to be able to use a generic RFC 5322 header field parser. Unfortunately what you have above is not going to work, as leading "*WSP" are 1) not valid according to RFC 5322 and 2) going to trigger "line continuation rule" (FWS) from RFC 5322.

I suggest you disallow leading "*WSP". If you agree, then the ABNF will become:

```
sts-policy-record        = sts-policy-field *WSP
                           *(CRLF sts-policy-field *WSP)
```

(Note that trailing *WSP is fine).

If you really really want leading *WSP, then you need to add a sentence to the document stating that this format can't be parsed by RFC 5322 parser, without stripping leading WSP on each line first.

Minor Issues (1)

(Alexey) 1.1. Terminology

> The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

You should consider switching to the new template and replace the above with:

> The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. These words may also appear in this document in lowercase, absent their normative meanings.

We also define the following terms for further use in this document:

> o MTA-STS Policy: A commitment by the Policy Domain to support PKIX authenticated TLS for the specified MX hosts.

PKIX needs a Normative Reference to RFC 5280.

Minor Issues (2)

(Alexey) In Section 3.1

Just to double check - I assume the order of different fields is not significant, other than the sts-version field being first? If that is correct, you should add a statement about this in an ABNF comment.

The reason why I ask is I want to know, whether the following 2 examples are the same (assume "ex1" is an extension field):

v=STSv1; id=foo; ex1=bar

and

v=STSv1; ex1=bar; id=foo;

Minor Issues (3)

(Alexey) 3.2. MTA-STS Policies

```
    The [RFC7231] "Content-Type" media type for this
resource MUST be "text/plain".
```

I think this requirement is a bit strong, because we should really register a new text/... media type for the policy format documented in the draft. I appreciate that this is not necessarily what you want to do in the document.

Minor Issues (4)

(Alexey) Section 3.2.

```
This resource contains the following newline-separated key/value
```

I think "CRLF-separated" would be better here, as there are multiple ways of signalling new lines in different protocols.

pairs:

- `"mode": (plain-text). One of "enforce", "testing", or "none", indicating the expected behavior of a sending MTA in the case of a policy validation failure.`

Please add a forward pointer to Section 5, where the exact meaning of these 3 fields is explained.

Minor Issues (5)

(Alexey) Section 3.2.

```
o  "mx": MX identity patterns (list of plain-text strings).  One or
      more patterns matching a Common Name ([RFC6125]) or Subject
      Alternative Name ([RFC5280]) DNS-ID present in the X.509
      certificate presented by any MX receiving mail for this domain.
```

The references above are confused. I think you want to say:

```
o  "mx": MX identity patterns (list of plain-text strings).  One or
      more patterns matching a Common Name or Subject Alternative
      Name ([RFC5280]) DNS-ID ([RFC6125]) present in the X.509
      certificate presented by any MX receiving mail for this domain.
```

In particular, I think Common Name is defined (maybe by reference) in RFC 5280 and DNS-ID is defined in RFC 6125..

There is similar text elsewhere in the document, that also needs updating.

Minor Issues (6)

(Alexey) Section 3.2.

```
For example: "mx: mail.example.com mx: .example.net" indicates
```

Maybe insert <CRLF> or \r\n between mx fields above, as the above example is not syntactically valid per your ABNF.

```
that mail for this domain might be handled by any MX with a
certificate valid for a host at "mail.example.com" or
"example.net".
```

Minor Issues (7)

(Alexey) Section 3.2.

```
sts-policy-max-age-value = 1*10(DIGIT)
```

Your ABNF allows for leading 0s. Are leading zeroes Ok? If not, you either need to make the ABNF more restrictive or you add an ABNF comment saying that. For example for the latter:

```
sts-policy-max-age-value = 1*10(DIGIT)
                           ; leading 0s are disallowed
```

If leading 0s are allowed, you don't have to do anything (you can say that explicitly). I just wanted to double check.

Minor Issues (8)

(Alexey) Section 3.2.

```
sts-policy-ext-value     = 1*(%x21-3A / %x3C / %x3E-7E)
                             ; chars, excluding "=", ";", SP, and
                             ; control chars
```

I just want to double check that you really want to be that restrictive in the policy format? If extensions want to add a field with human readable text, at least allowing for space might be useful. I don't see much reason to prohibit "=" and ";" here either.

Minor Issues (9)

(Alexey) Section 3.3.

```
3.3.  HTTPS Policy Fetching

   When fetching a new policy or updating a policy, the HTTPS endpoint
   MUST present a X.509 certificate which is valid for the "mta-sts"
   host (e.g.  "mta-sts.example.com") as described below, chain to a
   root CA that is trusted by the sending MTA, and be non-expired.
```

I think here you are repeating what is already mandated by RFC 5280. I am wondering what else have you missed. Would it be better just to point to RFC 5280 here? Or at least add "See RFC 5280 for more details about certificate verification".

Also, what about various certificate key usage fields? Some libraries/applications verify that certificates used are allowed for intended purposes.

Minor Issues (10)

(Alexey) Section 3.4.

```
3.4.  Policy Selection for Smart Hosts and Subdomains

    When sending mail via a "smart host"--an intermediate SMTP relay
    rather than the message recipient's server--compliant senders MUST
    treat the smart host domain as the policy domain for the purposes of
    policy discovery and application.
```

I don't think your definition of smart host is quite right. Email already uses intermediate SMTP relays which are specified by MX records. These don't have to correspond to "message recipient's server"s.

I suggest replacing the "smart host" definition with something like this:

```
    When sending mail via a "smart host"--an administratively
    configured intermediate SMTP relay, which is different from
    the message recipient's server as determined from DNS --compliant
    senders MUST treat the smart host domain as the policy domain for
    the purposes of policy discovery and application.
```

.

Minor Issues (11)

(Alexey)

- I am a bit uneasy about requiring to use SNI extension (as opposed to requiring to implement it). I think the text is not entirely consistent about MUSTs there. I suggest authors review the whole section for consistency.

- RFC 3207 (SMTP STARTTLS extension) must be a Normative Reference, as it is required to implement (and understand) this document.

Minor Issues (12)

(Chris)

I suggested a security consideration. Here's a slightly edited version of the same consideration:

```
This mechanism causes an MTA (an automated system) to adopt the
role of an HTTPS client in a scenario where the HTTPS server may
be hostile to operation of the MTA. A full HTTP stack is a large
amount of code that may contain coding errors that expose the MTA
to new implementation vulnerabilities due to the increased attack
surface. This threat can be partially mitigated by using a
hardened HTTPS client library that has been tested against a
fuzzing HTTPS test server. This threat can also be partially
mitigated by isolating the HTTPS code into a separate process that
does not have access to the normal MTA machinery and making sure
the MTA machinery gracefully handles a wedged HTTPS co-process.
```

I think this text would be good to include in the security considerations section.

.