

---

# **RFC 7489 & 8460 SMTP Report Monitoring Utilities.**

**Barnaby Shearer**

**Jun 09, 2022**



**CONTENTS:**

<b>1</b>	<b>RFC 7489 Domain-based Message Authentication, Reporting, and Conformance (DMARC)</b>	<b>3</b>
<b>2</b>	<b>RFC 8460 SMTP TLS Reporting</b>	<b>5</b>
2.1	Installation . . . . .	5
2.2	email_report_checker . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



RFC 7489 & 8460 SMTP Report Monitoring Utilities.

Note these are for my hobby domain, do not try running them on even moderate traffic MTAs



## RFC 7489 DOMAIN-BASED MESSAGE AUTHENTICATION, REPORTING, AND CONFORMANCE (DMARC)

First ensure your DMARC DNS TXT record contains *rua* and *ruf* to request reports:

```
_dmarc.zi.is. 3600 IN TXT "v=DMARC1; p=reject; pct=100; ruf=mailto:b+dmarc@zi.is;  
rua=mailto:b+dmarc@zi.is; adkim=s; aspf=s"
```

Then load the reports via IMAP:

```
dmarc-report-checker m.zi.is b@zi.is Archive > dmarc.json
```

And report your statistics

```
jq '[ .[].record.row | select(.source_ip == "68.183.35.248") | select(.policy_evaluated.  
→dkim == "pass") | .count | tonumber] | add' dmarc.json  
jq '[ .[].record.row | select(.source_ip == "68.183.35.248") | select(.policy_evaluated.  
→dkim == "fail") | .count | tonumber] | add' dmarc.json
```





## RFC 8460 SMTP TLS REPORTING

First create a DNS TXT record to request reports:

```
_smtp._tls.zi.is. 3600 IN TXT "v=TLSRPTv1;rua=mailto:b+tls@zi.is"
```

Then load the reports via IMAP:

```
tls-report-checker m.zi.is b@zi.is Archive > tls.json
```

And report your statistics

```
jq '[.[] | [.policies[].summary["total-successful-session-count"]] | add] | add' tls.json  
jq '[.[] | [.policies[].summary["total-failure-session-count"]] | add] | add' tls.json
```

### 2.1 Installation

```
python3 -m pip install email-report-checker
```

### 2.2 email\_report\_checker

Check DMARC reports.

`email_report_checker.dmarc.get_dmarc_report(msg: Message) → Any`

Read DMARC report from message.

`email_report_checker.dmarc.main() → None`

Entrypoint.

`email_report_checker.dmarc.report(mail: IMAP4) → List[Any]`

Read all DMARC reports from IMAP.

`email_report_checker.dmarc.xml2dict(xml: Any) → Any`

Convert xml to dict.

---

Check TLS reports.

`email_report_checker.tls.get_tls_report(msg: Message) → Any`

Read TLS report from message.

`email_report_checker.tls.main()` → None

Entrypoint.

`email_report_checker.tls.report(mail: IMAP4)` → List[Any]

Read all TLS reports from IMAP.

## INDICES AND TABLES

- genindex
- modindex
- search



## PYTHON MODULE INDEX

### e

`email_report_checker.dmarc`, [5](#)

`email_report_checker.tls`, [5](#)



## INDEX

### E

`email_report_checker.dmarc`  
    module, 5  
`email_report_checker.tls`  
    module, 5

### G

`get_dmarc_report()` (in module *email\_report\_checker.dmarc*), 5  
`get_tls_report()` (in module *email\_report\_checker.tls*), 5

### M

`main()` (in module *email\_report\_checker.dmarc*), 5  
`main()` (in module *email\_report\_checker.tls*), 5  
module  
    *email\_report\_checker.dmarc*, 5  
    *email\_report\_checker.tls*, 5

### R

`report()` (in module *email\_report\_checker.dmarc*), 5  
`report()` (in module *email\_report\_checker.tls*), 6

### X

`xml2dict()` (in module *email\_report\_checker.dmarc*), 5