Preventing Unauthorized Email Spoofing with DMARC

January 24, 2016
About Your Speaker

- Founding Team Member at Agari
- Previous Roles:
  - CTO of Brandmail Solutions
  - CTO of Concurro
  - CTO of 365 Media
  - Director, Strategic Projects, SAP
  - Applications Consultant, Oracle

John Wilson
Field CTO, Agari
About Agari

Agari’s Mission:

Eliminate email as a vector for Cyber Crime

Founded in 2009, Agari provides systems and services to help companies take back control of their email channel.

Agari’s 100+ clients include:
• 4 of the top 5 US banks
• 3 of the top 5 UK banks
• 4 of the top 5 social networks
Agenda

• Why cybercriminals love email
• Previous attempts to solve this problem
• The current best solution
• Technical details of SPF, DKIM, and DMARC
• Open source tools
• Edge cases and solutions
• Estimating DMARC adoption with Ashley Madison data
Email’s Fundamental Flaw

- Email has no built-in authentication
- Spoofing an email address is a trivial effort – no hacking required
Fraud Example #1: Pharma Spam

Manager Drugs <imbecilic@f5.com>
To: REDACTED <postmaster@outlook.com>
Levitra Cialis Pack  Price: USD 2.10 breadfruit

http://webcrex.com/officialism
Levitra
Cialis Pack  Price: USD 2.10
Cialis Pack includes pills of Cialis (generic) 20mg, Cialis Super Active 20mg, Cialis Soft Tabs 20mg, Cialis Professional 20mg
Manager Drugs

January 1, 2016 at 10:31 PM

MD
Fraud Example #1: Pharma Spam

Manager Drugs <imbecilic@f5.com>
To: REDACTED <postmaster@outlook.com>

Levitra Cialis Pack Price: USD 2.10
Cialis Pack includes pills of Cialis (generic)
20mg
Manager Drugs

http://webcrex.com/officialism

Levitra!
Cialis Pack Price: USD 2.10
Cialis Pack includes pills of Cialis (generic)
20mg
Manager Drugs
Fraud Example #2: Phishing

Dear Customer,

This mail is sent to you as a last notice.

For avoid suspension of your account please allow 2 minutes to the Apple customer service by clicking the link below:

Confirm my account now Å»

Thank you for choosing Apple.

Sincerely,

The Apple Customer Service
Fraud Example #2: Phishing
Fraud Example #2: Phishing

A screenshot showing a phishing email attempting to trick the user into entering their Apple ID and password on a malicious website (damiandrake.de) instead of the official Apple website.
Fraud Example #3: Malware

From: matthew.collins@lifelock.com
To: REDACTED <postmaster@outlook.com>

Subject: invoice

Attachments:
- augmenting mommy.zip

Sent: November 17, 2015 at 7:22 AM
Fraud Example #3: Malware

To: REDACTED <postmaster@outlook.com>

invoice

augmenting mommy.zip

VirusTotal

SHA256: d800579892050fb9f7d3b81b643380b84015ceec19c59501769b5e3c045aa104
File name: augmenting mommy.zip
Detection ratio: 43 / 55
Analysis date: 2016-01-05 23:40:02 UTC (1 minute ago)
Previous Attempts to Fix Email

There have been numerous attempts to reduce/eliminate unauthorized email spoofing.

These Include:

1. Sender Policy Framework (SPF)
2. SenderID
3. DomainKeys
4. DomainKeys Identified Mail (DKIM)
5. Author Domain Signing Practices (ADSP)
6. Visual Trust Indicators: Brandmail, GoodMail, Iconix, Google Gold Key
7. Domain-Based Message Authentication, Reporting, and Conformance (DMARC)
Visual Trust Indicators

Welcome to Brandmail Solutions, Europe’s leading email authentication and email ID solutions provider. At Brandmail, we work with both ISP’s and Sending Brands to provide the most robust and scalable preventative anti-phishing solution on the market. Based on Domain Keys and DKIM, the Brandmail solution is easy to enable and requires no proprietary technology for EmailSenders.

Our system is now deployed across a network of over 25 million active mailboxes in Europe. Leading brands utilizing our technology include:

- Otto
- Weltbild
- Postbank
- Facebook

The Brandmail Solutions Experience

- Patented Technology
- 100% Secure
- Inbox Branding
- Guaranteed Delivery
- Eliminate Phishing
- No Plug-in Required
- DKIM + DomainKeys

Secure Communications and Measurable Brand Impressions
And the current winners are...

There have been numerous attempts to reduce/eliminate unauthorized email spoofing.

These Include:

1. **Sender Policy Framework (SPF)**
2. **SenderID**
3. **DomainKeys**
4. **DomainKeys Identified Mail (DKIM)**
5. **Author Domain Signing Practices (ADSP)**
6. **Visual Trust Indicators: Brandmail, GoodMail, Iconix, Google Gold Key**
7. **Domain-Based Message Authentication, Reporting, and Conformance (DMARC)**
Relevant Standards

RFC 5321: Simple Mail Transport Protocol (SMTP)
RFC 5322: Internet Message Format
RFC 7372: Sender Policy Framework v1 (SPF)
RFC 6376: DomainKeys Identified Mail (DKIM) Signatures
RFC 7489: Domain-based Message Authentication, Reporting, and Conformance (DMARC)
DRAFT: Authenticated Received Chain (ARC)
Authenticated Email with DMARC

DMARC
- Visibility
- Policy
- Alignment
- Scalable

SPF
- Authenticates Envelope
- Breaks due to relaying
- Must Understand Ecosystem
- Receivers don’t honor policy model

DKIM
- Survives Relaying
- Authenticates the Signing domain
- Must Understand Ecosystem
- Nobody adopted policy model (ADSP)

SMTP
- Moves email around the Internet
- No Authentication
SMTP in Action

SERVER

220 mta1257.mail.bf1.yahoo.com ESMTP ready
250 mta1257.mail.bf1.yahoo.com
250 sender <test@test.com> ok
250 recipient <johnmwilson3@yahoo.com> ok
354 go ahead

CLIENT

helo sherlock-poc-00.qa.agari.com
mail from: <test@test.com>
rcpt to: <johnmwilson3@yahoo.com>
data
Subject: SMTP Demo
Message-Id: <159@agari.com>
From: test@test.com
This is a test message.
.
dirdel
250 ok dirdel
quit
A Question of Identity

```
helo sherlock-poc-00.qa.agari.com
mail from: <test@test.com>
rcpt to: johnmwillson3@yahoo.com
data
DKIM-Signature: v=1; d=agari.com; s=s1024; h=...
Subject: SMTP Demo
Message-Id: <160@agari.com>
From: test@example.com

This is a test message.
.
quit
```
A Question of Identity

heo sherlock-poc-00.qa.agari.com

mail from: <test@test.com>
rcpt to: johnmwilson3@yahoo.com
data
DKIM-Signature: v=1; d=agari.com; s=s1024; h=...
Subject: SMTP Demo
Message-Id: <160@agari.com>
From: test@example.com

This is a test message.
.
quit
A Question of Identity

hello sherlock-poc-00.qa.agari.com
mail from: <test@test.com>
rcpt to: johnmwilson3@yahoo.com
data
DKIM-Signature: v=1; d=agari.com; s=s1024; h=...
Subject: SMTP Demo
Message-Id: <160@agari.com>
From: test@example.com

This is a test message.
.
quit
A Question of Identity

```
helo sherlock-poc-00.qa.agari.com
mail from: <test@test.com>
rcpt to: johnmwilson3@yahoo.com
data
DKIM-Signature: v=1; d=agari.com; s=s1024; h=...
Subject: SMTP Demo
Message-Id: <160@agari.com>
From: test@example.com

This is a test message.
.
quit
```
This is a test message.
Sender Policy Framework (SPF)

$ dig apple.com TXT +short
"v=spf1 ip4:17.0.0.0/8 -all"

$ dig paypal.com TXT +short
"v=spf1 include:pp._spf.paypal.com
include:3ph1._spf.paypal.com
include:3ph2._spf.paypal.com
include:3ph3._spf.paypal.com
include:3ph4._spf.paypal.com
include:c._spf.ebay.com ~all"
Some Possible Mail Flows

Postfix -> GMail
Some Possible Mail Flows

- Postfix
- GMail

- Exchange
- Office365
- GMail
Some Possible Mail Flows

- Postfix → GMail
- Exchange → Office365 → GMail
- Sendmail → UCLA → Yahoo
SPF Issues

1. Breaks due to Relaying/Auto-Forwarding
2. Mailing Lists
3. Authenticates the Envelope Domain (MAIL FROM)
4. 10-lookup limit
5. DNS: TCP Failover
6. Don’t forget your NDRs and OOOs! (HELO fallback)
DomainKeys Identified Mail (DKIM)

DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/relaxed;
t=1452797764;
  s=m1; d=mktdns.com; i=@mktdns.com;
  h=Date:From:To:Subject:MIME-Version:Content-Type;
  bh=RRSNK9HgZTD19RkhqzAz6BCwQwOgMkeqlYhpQlNcBe0=;
  b=M3uhaNzDRcdjyb7W47UlAJFxNfEaxWQUutumTPg+ZDbpFITsRTvq5IisZNM+1N8D
    KUkl62WTUU5AR6Uza8P+Gx+E1+EDzvI67+K2hfwCMiJxGy+A2VyNzvyn3KPVL7QWmdx
    b8QGgwm2CD5MmpiGoof5fv42ILlfFIG+ghVyW4i4=

$ dig m1._domainkey.mktdns.com TXT +short
"v=DKIM1\;k=rsa
\;p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCrXrxjXj4TyZ0zoKUw3mlg
+zRZG3i6fkWmcqB9+/HSZdxUmStLhq5EKWQuXZfHZ6QsRUq/
ZKbp61OCXUPyY5k11FBkJjvb2qPlil/51OWcRMMDKsaJxtDHxXHtkjgEgWM8Xyes/
LEkQ5HlPQN+89DUEAJGf11gTa6rk1GxzCH01QIDAQAB"
DomainKeys Identified Mail (DKIM)

DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/relaxed;
t=1452797764;
  s=m1; d=mktdns.com; i=@mktdns.com;
  h=Date:From:To:Subject:MIME-Version:Content-Type;
  bh=RRSNK9HgZTD19RkhqzAz6BCwQwOgMkeqlYhpQ1NcBe0=;
  b=M3uhaNzDRcdjyb7W47UlAJFxNfEaxWQUtumTPg+ZDbpFITsIRTvq5IisZNM+1N8D
  KUk162WTTU5AR6Uza8P+Gx+El+EDzvI67+K2hfwCMiJxGy+A2VyNzvyn3KPVL7QWmdx
  b8QGgwm2CD5MmpiGoof5fv42ILlffIG+ghVyW4i4=

$ dig m1._domainkey.mktdns.com TXT +short
"v=DKIM1\;k=rsa
\;p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCrXrxjXj4Tyz0zoKUw3mlg
+zRZG3i6fkWmcqB9+/HSZdxUmStLhq5EKWQuXZfHZ6QsRUq/
ZKbP61OCXUPyY5k1lFBkjjbv2qPlil/51OWcRMDKsaJxtDHXxXHtkjgEgWM8Xyes/
LEkQ5HlPQN+89DUEAJGf11gTa6rk1GxzCh01QIDAQAB"
DomainKeys Identified Mail (DKIM)

DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/relaxed;
t=1452797764;
  s=m1; d=mktdns.com; i=@mktdns.com;
  h=Date:From:To:Subject:MIME-Version:Content-Type;
  bh=RRSNK9HgZT019RkhqzAz6BCwQwOgMkeqlYhpQ1NcBe0=;
  b=M3uhaNzDRcdjyrb7W47U1AJFxFxNfExWQUtumTPg+ZDbpFITSiRTvq5IisZNM+1N8D
  KUk162WTTU5AR6Uza8P+Gx+E1+EDzvI67+K2hfwCMiJxGy+A2VyNzvyn3KPVL7QWmdx
  b8QGgwm2CD5MmpiCOof5fv42IL1fFIG+ghVyW4i4=

$ dig m1._domainkey.mktdns.com TXT +short
"v=DK1M1;k=rsa
\;p=MIGfMA0GCSqGSIb3DQEBADQYREAwHDAgMBMGA1UdDQQ
+zRZG3i6fkWmcqB9+/HSZdxUmStLhq5EKWQuXZfHZ6QsRUq/
ZKbP6lOCXUPyY5k1lFBkjjbv2qPlIL/5lOWcRMDKsaJxtDHXxXHtkjgEgWM8Xyes/
LEkQ5H1PQN+89DUEAJGf1lgTa6rk1GxzC01QIDAQAB"
DomainKeys Identified Mail (DKIM)

DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/relaxed;
t=1452797764;
  s=m1; d=mktdns.com; i=@mktdns.com;
  h=Date:From:To:Subject:MIME-Version:Content-Type;
  bh=RRSNK9HgZT019RkhqzAz6BCwQwOgMkeqlYhpQlNcBe0=;
  b=M3uhaNzDRcdjrb7W47U1AJFxNfEaxWQUtumTPg+ZDbpFIItSiRTvq5IisZNMe1N8D
  KUKl62WTTU5AR6Uza8P+Gx+E1+EDzvI67+K2hfwCMiJxGy+A2VyNzvyn3KPVL7QWmdx
  b8QGgw2CD5MmpiCoof5fv42ILlfFIG+ghVyW4i4=

$ dig m1._domainkey.mktdns.com TXT +short
"v=DKIM1\;k=rsa\":p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQRznG3i6fckWmcqB9+/HSZdxUmStLhq5EKWQuXZfHZ6qsRUq/
ZKbP610CXUPyY5k1lFBkjjbvy2q PlLL/5lOWcRMDKsaJxtDHxxXHtkjgEgWM8Xyes/
LEkQ5H1PQN+89DUEAJGf11gTa6rk1GxzC01QIDAQAB"
DKIM Issues

1. Benign Message Modifications
2. Mailing Lists
3. Authenticates Signing Identity (DKIM d=)
4. Replay Attack
5. Key Size
DMARC

$ dig _dmarc.twitter.com TXT +short
"v=DMARC1; p=reject; rua=mailto:d@rua.agari.com; ruf=mailto:d@ruf.agari.com; fo=1"

v=DMARC1  Record applies to version 1 of the DMARC specification
p=reject Domain owner wishes to have failing messages blocked by receivers
rua=mailto:d@rua.agari.com Receivers should send summary data to d@rua.agari.com
ruf=mailto:d@ruf.agari.com Receivers should send forensic data to d@ruf.agari.com
fo=1 Domain owner would like forensic reports for messages failing 1 or more protocols
DMARC Data Flow
DMARC XML Aggregate Data Example

```xml
<xml version="1.0">?
<feedback>
  <report_metadata>
    <org_name>Yahoo! Inc.</org_name>
    <email>postmaster@dmarc.yahoo.com</email>
    <report_id>1393150703.9760</report_id>
    <date_range>
      <begin>1393027200</begin>
      <end>1393113599</end>
    </date_range>
  </report_metadata>
  <policy_published>
    <domain>alertsp.chase.com</domain>
    <adkim>r</adkim>
    <aspf>r</aspf>
    <p>reject</p>
    <pct>100</pct>
  </policy_published>
</feedback>
<record>
  <row>
    <source_ip>106.10.149.115</source_ip>
    <count>1</count>
    <policy_evaluated>
      <disposition>none</disposition>
      <dkim>pass</dkim>
      <spf>fail</spf>
    </policy_evaluated>
  </row>
  <identifiers>
    <header_from>alertsp.chase.com</header_from>
  </identifiers>
  <auth_results>
    <dkim>
      <domain>alertsp.chase.com</domain>
      <result>pass</result>
    </dkim>
    <spf>
      <domain>alertsp.chase.com</domain>
      <result>fail</result>
    </spf>
  </auth_results>
</record>
```
DMARC Forensic Feedback Example

Full Message Example (hotmail.com, outlook.com, etc.):

```
From: "Dr. Oz" <pillDr2543@facebook.com>
To: "REDACTED" <postmaster@hotmail.com>
Subject: Vi@ra Super Discount
Date: March 1, 2014 at 7:25:32 PM PST

Hey super stud man,
You can get the best meds here at our new pharmacy site 20% discounted v@laG@ra and C@ialS.
```

This is an email abuse report for an email message received from IP 85.26.183.63 on Sat Mar 1 19:25:32 PST 2014. The message below did not meet the sending domain's authentication policy.
For more information about this format please see http://www.ietf.org/rfc/rfc5325.txt.
Typical DMARC Policy Enforcement Flow

From: john@example.com
To: bob@gmail.com
Subject: Urgent Alert!

Do SPF or DKIM prove this message was sent by example.com?

Yes

Is the content OK?

Yes
INBOX

No

SPAM

What is example.com’s DMARC policy?

none

No

quarantine

reject

Yes

none

reject
DMARC Policy Enforcement in Action

telnet> Trying 66.196.118.35...
Connected to mta6.am0.yahoodns.net.
Escape character is '^]'.
220 mta1312.mail.bf1.yahoo.com ESMTP ready
HELO foon.paul.sf.agari.com
250 mta1312.mail.bf1.yahoo.com

MAIL FROM: <security@chase.com>
250 sender <security@chase.com> ok

RCPT TO: <johnmWilson3@yahoo.com>
250 recipient <johnmWilson3@yahoo.com> ok

DATA
354 go ahead
Message-Id: <1393480824.24.agari1393480824@foon.paul.sf.agari.com>
Date: Wed, 26 Feb 2014 22:00:24 -0800
Subject: Urgent Message From security@chase.com
From: security@chase.com
To: johnmWilson3@yahoo.com

This email message was sent by Agari to demonstrate just how easy it is to spoof an email address.

This message was sent to johnmWilson3@yahoo.com.

554 5.7.9 Message not accepted for policy reasons. See http://postmaster.yahoo.com/errors/postmaster-28.html
Connection closed by foreign host.
Limitations

- DMARC does not solve the “Friendly From” Problem:

  From: PayPal Security <paypay2182@hotmail.com

- Inbox differentiation may help here

<table>
<thead>
<tr>
<th>Primary</th>
<th>Social</th>
<th>Promotions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT&amp;T Customer Care</td>
<td>Your AT&amp;T wireless bill is ready to view - myAT&amp;T Acc</td>
<td>5:18 pm</td>
</tr>
<tr>
<td>Wells Fargo Online</td>
<td>Wells Fargo ATM/Debit/Prepaid Card: International Activi</td>
<td>8:49 am</td>
</tr>
<tr>
<td>Chase Card Services</td>
<td>Thank you for scheduling your online payment - E-mail S</td>
<td>Mar 1</td>
</tr>
<tr>
<td>Safeway just for U</td>
<td>Your Coupons are Here! - safeway Hand-picked coupons</td>
<td>Feb 28</td>
</tr>
<tr>
<td>Expedia.com</td>
<td>Final Details For Your Trip : Singapore - 03/01/2014 - (Itir</td>
<td>Feb 28</td>
</tr>
</tbody>
</table>

- If your mail provider doesn't support DMARC, the bad stuff still gets through
  - This is an ever-decreasing pool
  - Herd Immunity
From: john@yahoo.com

Messages fail SPF because the Mailing List Server is not in theyahoo.com SPF record.

Messages fail DKIM due to small message modifications, such as footers, subject line markup, etc.

List message blocked at Gmail, Hotmail, etc. due to authentication failure + p=reject
But wait it gets worse…

The Mailing List Server sees a bunch of SMTP 5xx errors due to yahoo.com’s DMARC p=reject policy.

The Mailing List Server automatically removes members from the mailing list after N messages that user bounce. To the Mailing List Server, DMARC rejections look no different than “mailbox full” or “no such user” errors.

After a couple of Yahoo users post to a mailing list, most of the mailing list users get unsubscribed since the yahoo.com messages were undeliverable to anybody at Gmail, Hotmail, Yahoo, AOL, Comcast, etc.
Options to deal with Mailing Lists

1. List Participant:
Post using an address that doesn’t implement a strong DMARC policy

Pros: No need to change mailing list software.
Cons: This is getting more and more difficult! How am I supposed to know my webmail provider just implemented DMARC p=reject?!

2. List Operator:
Send list messages “From:” the list itself, rather than the originator. Add a Reply-To: header with the originator’s email address.

Pros: Keeps the mail flowing, even when the sending address has a p=reject policy.
Cons: Changes the semantics of Reply vs. Reply-All. Requires the List Operator to cooperate. Not all mailing list software can support this.

Mailman: http://wiki.list.org/DEV/DMARC
Options to deal with Mailing Lists (2)

3. List Operator:
Status Quo, except don’t auto-unsubscribe for DMARC rejections.

Pros: Avoids having the whole list get unsubscribed due to a couple of p=reject messages.
Cons: Some users won’t be able to post to the list anymore. Different receivers use different SMTP error codes to denote DMARC rejections.

4. Receiver:
Detect mailing list traffic and override DMARC reject for list-originated traffic.

Pros: No change necessary on part of poster or list operator.
Cons: Opens a loophole for fraud. Anybody can spoof a message from user@paypal.com to the list, circumventing paypal.com’s DMARC policy.
Options to deal with Mailing Lists (3)

5. List Operator, Receiver:
Implement ARC (Authenticated Received Chain).

Pros: Lets DMARC play well with mailing lists without sacrificing security.
Cons: Requires the list operator AND the receiver to implement ARC.

http://arc-spec.org/ - Authenticated Received Chain specification
OpenDKIM

OpenDKIM is a community effort to develop and maintain a C library for producing DKIM-aware applications and an open source milter for providing DKIM service.

The project started from a code fork of version 2.8.3 of the open source dkim-milter package developed and maintained by Sendmail, Inc.

OpenDKIM is an open source implementation of the DKIM (Domain Keys Identified Mail) sender authentication system proposed by the E-mail Signing Technology Group (ESTG), now standardized by the IETF (RFC6376). It also includes implementations of the RFC5617 Vouch By Reference (VBR, RFC5518) proposed standard and the experimental Authorized Third Party Signatures protocol (ATPS, RFC6541).

- C library + milter
- Performs DKIM signing and DKIM signature validation
Open Source SPF Tools

Sender Policy Framework Implementations

There are several SPF libraries available. Many mail servers support SPF natively. Most popular mail servers also have extensions or unofficial patches available. The SPF project does not endorse any of the software listed below unless noted otherwise. We have not checked all of the software for fitness for any purpose. Please report SPF-supporting software to the spf-discuss mailing list or by contacting us directly.

Libraries

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Author(s)</th>
<th>Contact</th>
<th>Test Suite releases passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>libspf2</td>
<td>C</td>
<td>Shevek, Wayne Schlitt</td>
<td>spf-devel</td>
<td>?</td>
</tr>
<tr>
<td>RMSPF</td>
<td>C (Windows)</td>
<td>Roger Moser</td>
<td>Roger Moser</td>
<td>?</td>
</tr>
</tbody>
</table>
| Mail::SPF| Perl      | Julian Mehnle, Shevek| Julian Mehnle| 2008.08: since 2.0.06  
          |            |                      | 2007.05: since 2.0.04  |
| pyspf   | Python    | Stuart Gathman, Scott Kitterman, Terence Way | see Python Package Index | RFC 7208: since 2.0.9  
          |            |                      |                          | 2008.08: since 2.0.5  
          |            |                      |                          | 2007.05: since 2.0.4  |
| jSPF    | Java      | Stefano Bagnara, Norman Maurer | see website | 2007.05: since 0.9.6  |
| InterPC.SPF | .NET   | Eddy Minet           | InterPC.SPF Forum | 2007.05: since 1.1  |
libspf2

Welcome

libspf2 implements the Sender Policy Framework, a part of the SPF/SRS protocol. It is used by programs like Exim, Zmailer and MS Exchange to check SPF records and make sure that the email was not sent by a forged envelope sender, commonly used by spammers, scammers and email viruses/worms.

News

June 10th, 2013: libspf2 version 1.2.10 has been released, and is available here.

November 4th, 2008: libspf2 version 1.2.9 has been released, and is available here. An update is recommended.
OpenDMARC

OpenDMARC is a free open source software implementation of the DMARC specification. You can browse the source code, download the latest released version, and track the project here.

This work is part of an initiative of The Trusted Domain Project has been sponsored by TrustSphere, Cloudmark, Sendmail, NLNet, American Greetings, Gnome.

Documentation

- Package README
- libopendmarc
- opendmarc README
- opendmarc(8)
- opendmarc.conf(5)
- reports README
- opendmarc-import(8)
- opendmarc-reports(8)

- Implemented as a milter (mail filter)
- Performs DMARC verification and reporting
- Uses mySQL to store data between reporting intervals
DMARC Adoption
## DMARC Adoption – Consumer Mailbox Providers

85% of US, 60% world-wide consumer mailboxes today and growing

<table>
<thead>
<tr>
<th>Provider</th>
<th>Mailboxes</th>
<th>Data</th>
<th>Enforcement</th>
<th>Msg Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yahoo!</td>
<td>320 Million</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
</tr>
<tr>
<td>AT&amp;T (via Yahoo)</td>
<td>8 Million</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
</tr>
<tr>
<td>Rogers Communications (via Yahoo)</td>
<td>10 Million</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
</tr>
<tr>
<td>Verizon (via Yahoo)</td>
<td>6 Million</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
</tr>
<tr>
<td>Xtra, SBC, Ameritech, and additional Y! partners</td>
<td>20 Million</td>
<td>✓</td>
<td>✓</td>
<td>✓*</td>
</tr>
<tr>
<td>Google (Including 4.6M Apps Domains)</td>
<td>425 Million</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft (Hotmail, Live.com, Outlook.com, MSN)</td>
<td>380 Million</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>AOL</td>
<td>50 Million</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comcast</td>
<td>20 Million</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetEase (163.com, 126.com, yeah.net)</td>
<td>510 Million</td>
<td>✓</td>
<td>✓</td>
<td>✓**</td>
</tr>
<tr>
<td>xs4all.nl</td>
<td>1 Million</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mail.ru</td>
<td>300 Million</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yandex</td>
<td>200 Million</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LinkedIn</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Facebook</td>
<td>800 Million</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iCloud</td>
<td>15 Million</td>
<td>2016</td>
<td>2016</td>
<td></td>
</tr>
</tbody>
</table>

* URLs only; via private channel  ** Full body available upon request  * Via Private Channel
## DMARC Adoption – Business Mailbox Providers

<table>
<thead>
<tr>
<th>Open Source</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sendmail (via OpenDMARC)</td>
<td>production</td>
</tr>
<tr>
<td>Postfix (via OpenDMARC)</td>
<td>production</td>
</tr>
<tr>
<td>qmail (via qsmtpd and OpenDMARC)</td>
<td>production</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Commercial</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Apps</td>
<td>production</td>
</tr>
<tr>
<td>Cisco/IronPort</td>
<td>production</td>
</tr>
<tr>
<td>Proofpoint</td>
<td>production (limited, no reporting)</td>
</tr>
<tr>
<td>Symantec Cloud</td>
<td>production (no reporting)</td>
</tr>
<tr>
<td>Message Systems (via OpenDMARC)</td>
<td>production</td>
</tr>
<tr>
<td>Office 365</td>
<td>production (quarantine only, no reporting)</td>
</tr>
<tr>
<td>Alt-N Mdaemon</td>
<td>production</td>
</tr>
</tbody>
</table>
Estimating Receiver Adoption

1. Start with a list of email addresses
2. Look up the MX host for each
3. See if that host is known to support DMARC
4. Tally everything up

For #1 I will use the Ashley Madison breach dataset
Doing the MX Lookups

$ dig gmail.com mx +short
40 alt4.gmail-smtp-in.l.google.com.
20 alt2.gmail-smtp-in.l.google.com.
5 gmail-smtp-in.l.google.com.
30 alt3.gmail-smtp-in.l.google.com.
10 alt1.gmail-smtp-in.l.google.com.

$ dig netflix.com mx +short
1 aspmx.l.google.com.
10 aspmx2.googlemail.com.
10 aspmx3.googlemail.com.
5 alt1.aspmx.l.google.com.
5 alt2.aspmx.l.google.com.

$ dig ucla.edu mx +short
10 mx-ucb2.smtp.ucla.edu.
10 mx-asm2.smtp.ucla.edu.
10 mx-csb1.smtp.ucla.edu.
5 mx.smtp.ucla.edu.
10 mx-csb2.smtp.ucla.edu.
10 mx-ucb1.smtp.ucla.edu.
<table>
<thead>
<tr>
<th>Country</th>
<th>Estimated Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Venezuela</td>
<td>96.9%</td>
</tr>
<tr>
<td>Mexico</td>
<td>95.5%</td>
</tr>
<tr>
<td>USA</td>
<td>90.7%</td>
</tr>
<tr>
<td>New Zealand</td>
<td>89.1%</td>
</tr>
<tr>
<td>Canada</td>
<td>88.5%</td>
</tr>
<tr>
<td>Brazil</td>
<td>87.8%</td>
</tr>
<tr>
<td>Australia</td>
<td>86.7%</td>
</tr>
<tr>
<td>UK</td>
<td>86.4%</td>
</tr>
<tr>
<td>France</td>
<td>77.1%</td>
</tr>
<tr>
<td>Germany</td>
<td>42.0%</td>
</tr>
<tr>
<td>Japan</td>
<td>38.7%</td>
</tr>
</tbody>
</table>
Resources

http://dmarc.org - Everything about DMARC including the specification
http://dkim.org - Everything about DKIM
http://www.openspf.org/ - Everything you need to know about SPF
http://www.libspf2.org/ - An open-source library to do SPF
https://www.agari.com/resources/ - Agari technical and marketing documents, videos, whitepapers, etc.
http://www.stevejenkins.com/blog/2015/03/installing-opendmarc-rpm-via-yum-with-postfix-or-sendmail-for-rhel-centos-fedora/ - Step-by-step “how-to” to install and run opendmarc
http://wiki.list.org/DEV/DMARC - How to make Mailman play well with DMARC
http://arc-spec.org/ - Authenticated Received Chain specification
https://github.com/linkedin/lafayette - Project to ingest, store, index, and search ARF data
Thank You!