

DMARC – A FREE CONTROL TO PROTECT BRANDS

PRESENTED BY JASON WRIGHT

AGENDA

- Why would you want/need DMARC
- What is DMARC?
- Components of DMARC
- Configuring SPF
- Configuring DKIM
- DMARC Workflow
- DMARC Configuration
- DMARC Forensics
- Recap
- How to approach DMARC

WHY WOULD YOU WANT/NEED DMARC

- Palo Alto Unit 42's Response Report 2022 Outlines
 - BEC (Business Email Compromise) and Ransomware have been the most prevalent incident types.
 - "BEC heists, on average, earned cybercriminals in the range of \$286,000."
 - Domain spoofing/typo squatting/phishing attacks are typically what led to the initial BEC.
- Free control
- Protects your brand from email spoofing
- Most recently become a requirement for some cyber insurance policies
- Verifying that you're validating SPF/DKIM/DMARC for inbound emails can help prevent you from accepting a spoofed email
 - Note: Typically inbound rules for validating these lookups are not configured by default in O365 and many ESG (Email Security Gateways)

WHAT IS DMARC?

- Domain-based Message Authentication, Reporting & Conformance
 - RFC 7489
 - Builds on SPF and DKIM protocols
 - Purpose of DMARC is to give policies for message recipients to follow
 - Policies are P=None, P=Quarantine and P=Reject
 - Protects your brand from email spoofing
 - Should be configured for Parked Domains as well
 - Record is published in DNS and is available to anyone that can look up DNS
 - Completely free to implement

COMPONENTS OF DMARC

- Sender Policy Framework (SPF)
 - Formally Sender Permitted From
 - Path based email authentication technique
 - Specifies which IP addresses are authorized to send on behalf of the organization
 - Started as an experimental protocol by the IETF in 2006 as RFC 4408
 - Updated to RFC 7208 as of 2014

COMPONENTS OF DMARC

- DKIM – Domain Keys Identified Message
 - Signature based email authentication
 - Based off Domain Keys or DK, signature based email authentication developed by Yahoo
 - Documented as RFC 4870
 - Superseded by DKIM
 - Introduced by the IETF in 2007 as RFC 4871
 - Updated to RFC 6376 in 2011
 - Tells the recipient that the email came from trusted infrastructure
 - Will fail if the message is altered in traffic

CONFIGURING SPF

- Sender Policy Framework (SPF)
 - `v=spf1 include:_spf.google.com ~all`
 - `V=spf1 ip:192.168.0.0/24 ip:192.168.0.1 -all`
 - `V=Version SPFv1`
 - Mechanisms
 - ALL = Matches Always
 - A = Domain has an A record
 - IP4 = Sender is in a given IPv4 Range
 - IP6 = Sender is in a given IPv6 Range
- Mechanisms Continued
 - MX = domain has an MX record
 - PTR = Should be avoided – complicated DNS configuration with forward confirmed reverse DNS
 - EXISTS = Resolves to any address, pass
 - Include = references the policy of another domain. If that domain passes, it matches
- Qualifiers
 - + = Pass result, ? = Neutral, ~ = Soft fail, - = FAIL

CONFIGURING SPF

spf:google.com

Solve Email Delivery Problems



```
v=spf1 include:_spf.google.com ~all
```

Prefix	Type	Value	PrefixDesc	Description
	v	spf1		The SPF record version
+	include	_spf.google.com	Pass	The specified domain is searched for an 'allow'.
~	all		SoftFail	Always matches. It goes at the end of your record.

	Test	Result
✓	SPF Record Published	SPF Record found
✓	SPF Record Deprecated	No deprecated records found
✓	SPF Multiple Records	Less than two records found
✓	SPF Contains characters after ALL	No items after 'ALL'.
✓	SPF Syntax Check	The record is valid
✓	SPF Included Lookups	Number of included lookups is OK
✓	SPF Type PTR Check	No type PTR found
✓	SPF Void Lookups	Number of void lookups is OK
✓	SPF MX Resource Records	Number of MX Resource Records is OK
✓	SPF Record Null Value	No Null DNS Lookups found

CONFIGURING SPF

spf:_spf.google.com

Solve Email Delivery Problems



```
v=spf1 include:_netblocks.google.com include:_netblocks2.google.com include:_netblocks3.google.com ~all
```

Prefix	Type	Value	PrefixDesc	Description
	v	spf1		The SPF record version
+	include	_netblocks.google.com	Pass	The specified domain is searched for an 'allow'.
+	include	_netblocks2.google.com	Pass	The specified domain is searched for an 'allow'.
+	include	_netblocks3.google.com	Pass	The specified domain is searched for an 'allow'.
~	all		SoftFail	Always matches. It goes at the end of your record.

	Test	Result
✓	SPF Record Published	SPF Record found
✓	SPF Record Deprecated	No deprecated records found
✓	SPF Multiple Records	Less than two records found
✓	SPF Contains characters after ALL	No items after 'ALL'.
✓	SPF Syntax Check	The record is valid
✓	SPF Included Lookups	Number of included lookups is OK
✓	SPF Type PTR Check	No type PTR found
✓	SPF Void Lookups	Number of void lookups is OK
✓	SPF MX Resource Records	Number of MX Resource Records is OK
✓	SPF Record Null Value	No Null DNS Lookups found

CONFIGURING SPF

spf:_netblocks.google.com

Find Problems

Solve Email Delivery Problems

spf

```
v=spf1 ip4:35.190.247.0/24 ip4:64.233.160.0/19 ip4:66.102.0.0/20 ip4:66.249.80.0/20 ip4:72.14.192.0/18 ip4:74.125.0.0/16 ip4:108.177.8.0/21 ip4:173.194.0.0/16
```

Prefix	Type	Value	PrefixDesc	Description
	v	spf1		The SPF record version
+	ip4	35.190.247.0/24	Pass	Match if IP is in the given range.
+	ip4	64.233.160.0/19	Pass	Match if IP is in the given range.
+	ip4	66.102.0.0/20	Pass	Match if IP is in the given range.
+	ip4	66.249.80.0/20	Pass	Match if IP is in the given range.
+	ip4	72.14.192.0/18	Pass	Match if IP is in the given range.
+	ip4	74.125.0.0/16	Pass	Match if IP is in the given range.
+	ip4	108.177.8.0/21	Pass	Match if IP is in the given range.
+	ip4	173.194.0.0/16	Pass	Match if IP is in the given range.
+	ip4	209.85.128.0/17	Pass	Match if IP is in the given range.
+	ip4	216.58.192.0/19	Pass	Match if IP is in the given range.
+	ip4	216.239.32.0/19	Pass	Match if IP is in the given range.
~	all		SoftFail	Always matches. It goes at the end of your record.

CONFIGURING DKIM

- DKIM – Domain Keys Identified Message

- From: Example User <example@contoso.com>
- DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/relaxed;
- s=selector1; d=contoso.com; t=1429912795;
- h=From:To:Message-ID:Subject:MIME-Version:Content-Type;
- bh=<body hash>;
- b=<signed field>;

- DKIM – Domain Keys Identified Message

- V=version a= signing algorithm, d=signing domain identifier s=selector c=optional canonicalization algorithm for header and body q=optional default query method i=optional agent or user identifier t=signature timestamp x=expire time l=body length h=header z=optional header fields bh=body hash b=signature of headers and body

CONFIGURING DKIM

- Authentication-Results: mx.google.com;
- dkim=pass header.i=@chesapeake.edu header.s=selector2 header.b=huCQ9VrH;
- arc=pass (i=1 spf=pass spfdomain=chesapeake.edu dkim=pass dkdomain=chesapeake.edu dmarc=pass fromdomain=chesapeake.edu);
- spf=pass (google.com: domain of XXXXX@chesapeake.edu designates 40.107.96.133 as permitted sender) smtp.mailfrom=xxxxxx@chesapeake.edu;
- dmarc=pass (p=NONE sp=NONE dis=NONE) header.from=chesapeake.edu

CONFIGURING DKIM

dkim:chesapeake.edu:selector2

Find Problems

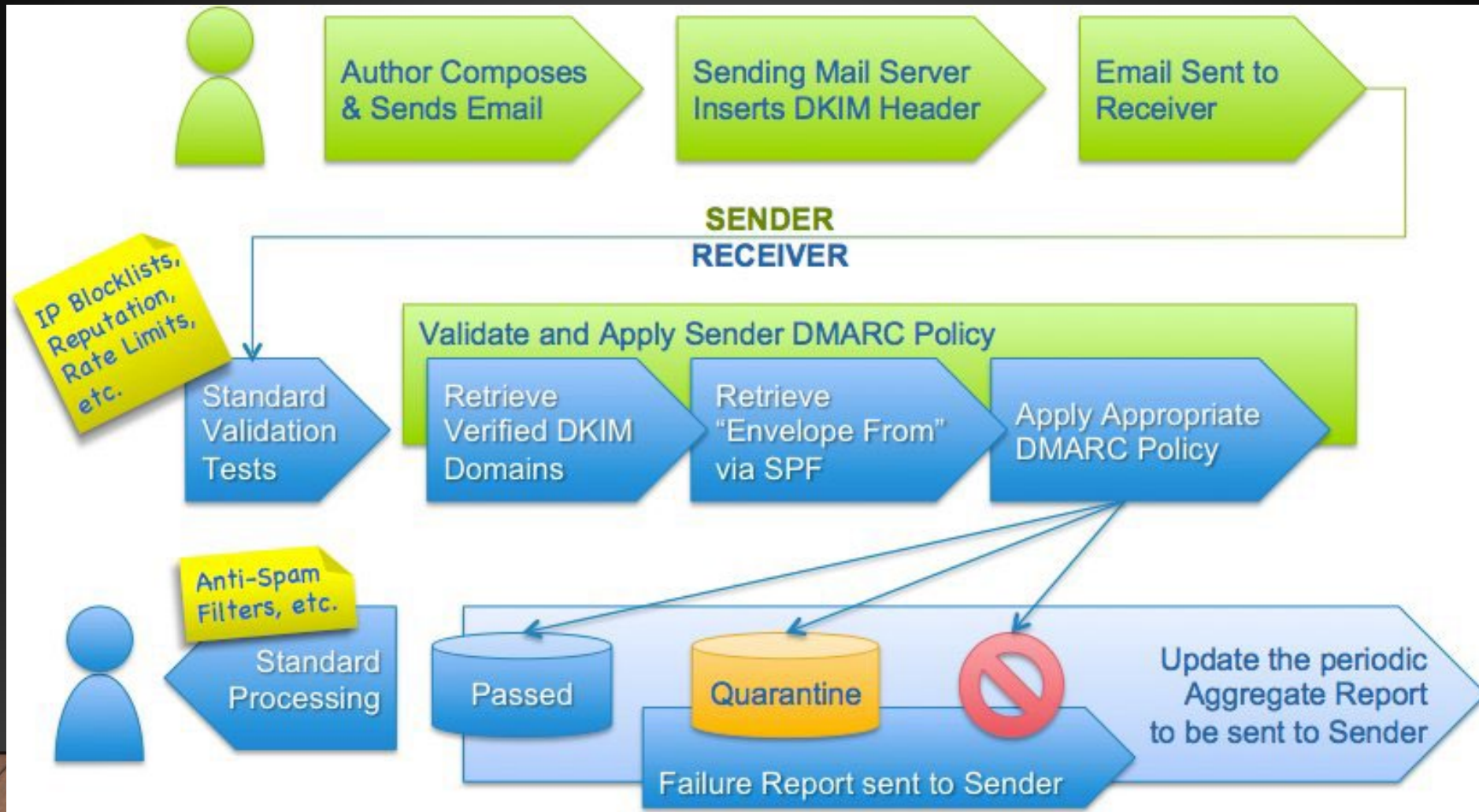
dkim

X EMAILS BOUNCING? MxToolbox has your email delivery solutions

```
v=DKIM1; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC7ojGv4DAmp4L8cIBERnZNVgjiCDhSZcbRqHAuw5KT1UdWFHR4DmZSe9te+y3aTQIHI/7JG3lK24qCLu4PidBIXn3I7b8xvnXkRfTeGdde/H7pNKrQG0PiLnKH1VGXQKMMQzKTKIsnv0mCYAdS9PpasSk86f9gF8D2uMqyDS1WCQIDAQAB
```

Tag	TagValue	Name	Description
v	DKIM1	Version	Identifies the record retrieved as a DKIM record. It must be the first tag in the record.
k	rsa (Length: 1024 bits)	Key Type	The type of the key used by tag (p).
p	MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC7ojGv4DAmp4L8cIBERnZNVgjiCDhSZcbRqHAuw5KT1UdWFHR4DmZSe9te+y3aTQIHI/7JG3lK24qCLu4PidBIXn3I7b8xvnXkRfTeGdde/H7pNKrQG0PiLnKH1VGXQKMMQzKTKIsnv0mCYAdS9PpasSk86f9gF8D2uMqyDS1WCQIDAQAB	Public Key	The syntax and semantics of this tag value before being encoded in base64 are defined by the (k) tag.

DMARC WORKFLOW



DMARC CONFIGURATION

- `v=DMARC1;p=reject;pct=100;rua=mailto:postmaster@dm
arcdomain.com`

- | • Tag Name | Purpose | Sample |
|------------|---|--|
| • v | Protocol version | <code>v=DMARC1</code> |
| • pct | Percentage of messages subjected to filtering | <code>pct=20</code> |
| • ruf | Reporting URI for forensic reports | <code>ruf=mailto:authfail@example.com</code> |
| • rua | Reporting URI of aggregate reports | <code>rua=mailto:aggrep@example.com</code> |
| • p | Policy for organizational domain | <code>p=quarantine</code> |
| • sp | Policy for subdomains of the OD | <code>sp=reject</code> |
| • adkim | Alignment mode for DKIM | <code>adkim=s</code> (strict or relaxed) |
| • aspf | Alignment mode for SPF | <code>aspf=r</code> (Strict or relaxed) |

DMARC CONFIGURATION

Aggregate Reports RUA

Combined data on a
group of emails

Not real-time, they are sent
everyday by default

Sent in XML format

No PII
Personal Identifiable Information

Supported in all
DMARC-compliant mailbox
providers

Forensic Reports RUF

Details of an
individual email

Sent almost immediately
after the failures

Plain text format

Contains PII
Personal Identifiable Information

Supported in only a
handful of mailbox
providers

DMARC CONFIGURATION

- Date and time range of the report
- The domain
- The IP address that sent the message
- Whether SPF and DKIM have passed or failed
- The DMARC policy applied
- The domain associated with SPF and DKIM

```
<?xml version="1.0" encoding="UTF-8" ?>
<feedback>
  <report_metadata>
    <report_metadata>
      <org_name>reporter_abc </org_name>
      <email>dmarc_ag_feedback@reporterdomain.com </email>
      <extra_contact_info>https://receiver.example/dmarc</extra_contact_info>
      <report_id>0001229911231 </report_id>
      <date_range>
        <begin>1569888000 </begin>
        <end>1569974399 </end>
      </date_range>
    </report_metadata>
  </report_metadata>
  <policy_published>
    <domain>example.com </domain>
    <adkim>r </adkim>
    <aspf>r </aspf>
    <p>reject </p>
    <sp>reject </sp>
    <pct>100 </pct>
  </policy_published>
  <record>
    <row>
      <source_ip>192.0.2.24</source_ip>
      <count>17 </count>
      <policy_evaluated>
        <disposition>none </disposition>
        <dkim>pass </dkim>
        <spf>pass </spf>
      </policy_evaluated>
    </row>
    <identifiers>
      <header_from>example.com </header_from>
    </identifiers>
    <auth_results>
      <dkim>
        <domain>example.com </domain>
        <result>pass </result>
        <selector>1234 </selector>
      </dkim>
      <spf>
        <domain>example.com </domain>
        <result>pass </result>
      </spf>
    </auth_results>
  </record>
```

reporter name

date range of the report

DMARC policy of domain evaluated

sending IP address

SPF/DKIM pass or fail results

SPF/DKIM alignment results

DMARC CONFIGURATION

dmARC:google.com

Find Problems

Solve Email Delivery Problems

dmARC



EMAILS BOUNCING? MxToolbox has your email delivery solutions



```
v=DMARC1; p=reject; rua=mailto:mailauth-reports@google.com
```

Tag	TagValue	Name	Description
v	DMARC1	Version	Identifies the record retrieved as a DMARC record. It must be the first tag in the list.
p	reject	Policy	Policy to apply to email that fails the DMARC test. Valid values can be 'none', 'quarantine', or 'reject'.
rua	mailto:mailauth-reports@google.com	Receivers	Addresses to which aggregate feedback is to be sent. Comma separated plain-text list of DMARC URIs.

DMARC FORENSICS

- Point them to a log aggregator, no one wants to read XML files trying to determine what service is sending email on behalf of the organization this way.
 - Tons of DMARC Analyzer/Aggregator options out there.
 - These services provide a graphical UI to analyze this information and they're typically free services up to a certain number of domains.
 - Typically cut off after 2-5 domains.
 - ValiMail, Dmarc Analyzer (bought by Mimecast), DMARCIAN, among others.

RECAP

- Why you want a DMARC Record
- What DMARC is
- The Components of DMARC
- How to configure the components (SPF/DKIM)
- DMARC Workflow, configuration and lookup and forensic records

HOW TO APPROACH DMARC

- DMARC Game plan
 - Decide on a log aggregator. No many free services to stare at XML Files.
 - Initial DMARC Configuration – P=None and point forensic records (RUA ONLY!!) to the aggregator. Wait for logs to trickle in.
 - Verify current position for SPF and DKIM for your know senders
 - IE – Check DNS, there should be SPF records for each sending service and a corresponding DKIM record. If not, one is missing and needs to be configured.
 - Verify position on validating inbound SPF, DKIM and DMARC lookups. If someone is going through the work of configuring this for their domain, trust that they did it right.
 - Open discussions with Marketing / IT on developing and maintaining a list of senders and the records in DNS.

HOW TO APPROACH DMARC

- DMARC Gameplan
 - Open discussions with Marketing / IT on developing and maintaining a list of senders and the records in DNS.
 - These records should be audited yearly depending on the environment.
 - SPF has a RFC limit of 10, after that you must use a A record lookup (Think Google Example).
 - Remove any stagnant records. If you stop using a service, make sure to remove them from the list.
 - Once all records are validating and no failures are being see via the aggregator, set p=quarantine and pct=XXX.
 - Eventually, P=reject.

WRAP UP

- DMARC is a Journey. The larger the environment, the longer the journey.
- Once its configured, just maintain it.
- Honorable Mentions
 - Microsoft Defender ARC - ARC will help reduce SPF, DKIM, and DMARC delivery failures that happen due to legitimate indirect mail flows.
 - Cloud to Cloud – Tenant to Tenant.
 - Configured by default by Microsoft and most cloud providers.

REFERENCES

- <https://dmarc.org/>
- <https://dmarc.org/wiki/Glossary#SPF>
- <https://dmarc.org/wiki/Glossary#DKIM>
- <https://dmarcian.com/>
- <https://thenewstack.io/palo-alto-networks-unit-42-publishes-2022-response-report/>
- <https://dmarc.org/overview/>
- <https://dmarcian.com/rua-vs-ruf/>
- <https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/use-dkim-to-validate-outbound-email?view=o365-worldwide>