H3 なりすましメールとDMARCを考える

InternetWeek2025

一般社団法人 日本ネットワークインフォメーションセンター

概要(1/2)

適切ではない「なりすまし」メールをなくすことは、**インターネットに関わるすべての人が共有できる理想**です。その実現に向けた方策として、送信ドメイン認証やDMARCポリシーによる運用が挙げられます。

しかし現在、多くの組織では「p=none」に設定されており、「p=quarantine」や「p=reject」への移行に踏み出しにくい状況が見られます。転送やメーリングリストといった運用上の事情、ポリシーを設定する側と受信者側の立場の違い、さらには利用者からの**「メールが届かない」といった問い合わせへの対応などが、移行の障壁となっているのではないでしょうか。**

概要(2/2)

本ハンズオンでは、DMARCポリシーとメール受信について、**技術的な詳細に踏み込みすぎない形**で手を動かしながら、「p=quarantineでも安心して運用できる状態」へ進める方法を考えていきます。また、「p=rejectにしても大丈夫か」といったケースを題材にディスカッションを行い、現実的な対応策を参加者の皆さまとともに探ります。

- p=noneから先に進めない背景とその整理
- 転送やメーリングリスト（From書き換え方式など）との関係
- 検証結果をどのように示し、どのように対応につなげるか

実運用に携わる方、メールの安全性を高めたい方、そして「適切ではない『なりすまし』メールのない世界」を目指すすべての方にとって、有益な時間となることを目指し、「p=rejectにできる世界」への道筋を描きます。

対象者と必要なもの

- 対象者
  - 技術導入や運用の検討および判断をされる方
  - なりすましメール対策に興味のある方
- 必要なもの
  - SSHクライアント(TeraTerm、PuTTY等)・SSHのポート番号として22番ポートを利用します。
  - UNIXコマンドラインに関する基本的な知識が必要です。

- 技術習得を目的としたものというよりも話題の軸を設定することを主眼にしています。
- すべてをコース通りに“終わらせる”必要はありません。話題の焦点を合わせるためのハンズオンとお考えください。

本ハンズオンでは...

- ターミナルを使ったり、アンケート(匿名)を取ったり、ディスカッションをしたりするセッションです。アンケートURL等、適宜ご案内します。
- アンケートやディスカッションにはご視聴のみなさまも是非ご参加ください。

※ 本セッションにおける発言の内容はご所属を代表するものではなく、あくまで本セッションのディスカッションのためのものです。将来に渡って何かをお約束するものではありません。建設的なディスカッションへのご協力をお願いいたします。

目次

- **ご案内**
- **DMARCおさらい**
- **DMARCポリシーと業務上の「起きること」**
- **ハンズオンに先立つご確認**
- **ハンズオン**
  - p=quarantine
  - お問い合わせ対応
  - 他部署他組織のメールサーバ ほか
- **ディスカッション**
  - 事前アンケート
  - 事例紹介

DMARCおさらい

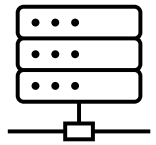
総務省「ISP におけるネットワークセキュリティ技術の導入及び普及促進に関する調査」DMARCハンズオン勉強会 エッセンス

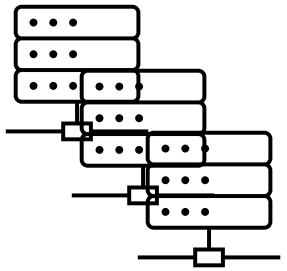
一般社団法人 日本ネットワークインフォメーションセンター

DMARC設定ステップ

自社

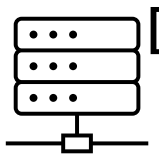
メール送信先





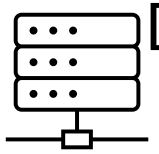
1. 最初

2. SPF or DKIM



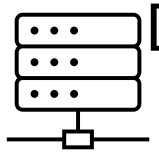
ドメイン名をかたる他のサーバへの対策

3. DMARC
p=none



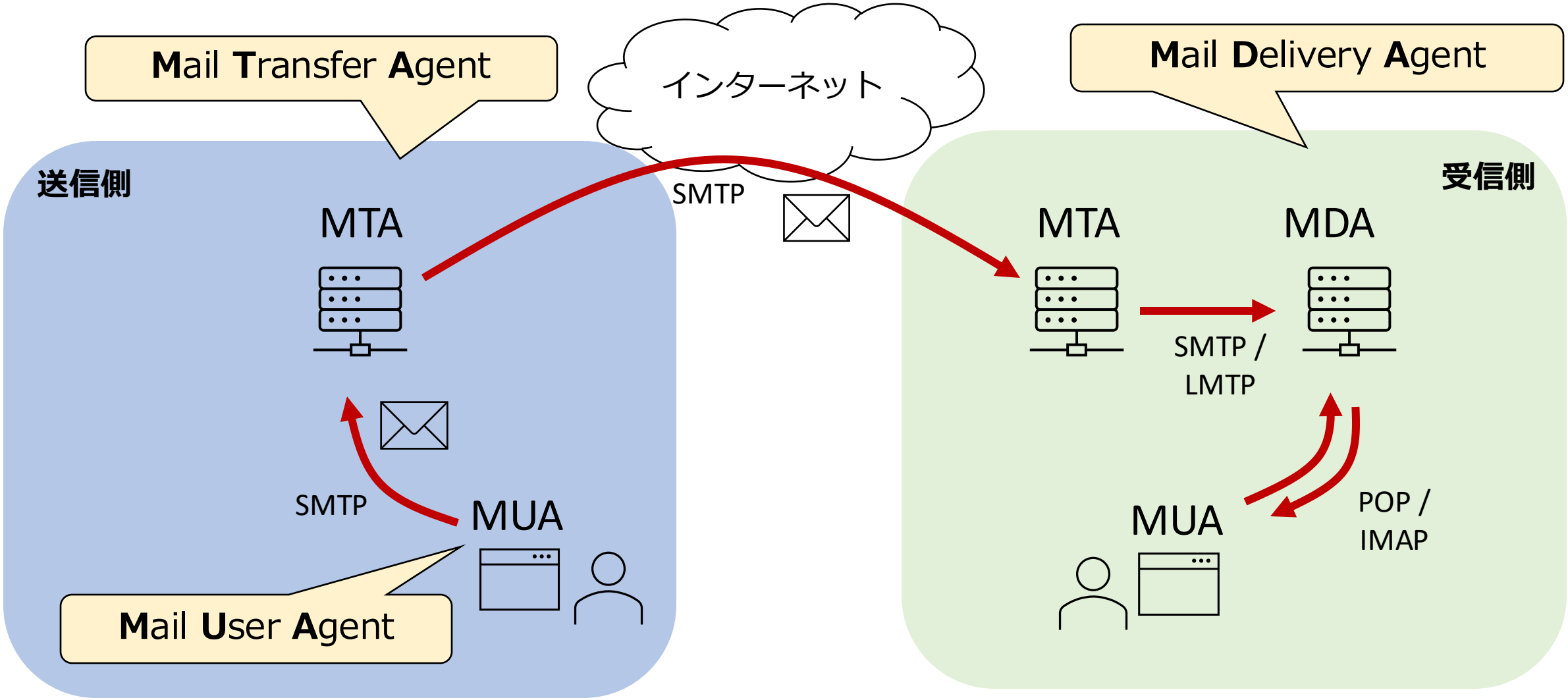
本文のFromと合っている前提＋ドメイン名をかたる他のサーバやメールへの対応方針を伝えられる／DMARCレポートを受け取れる

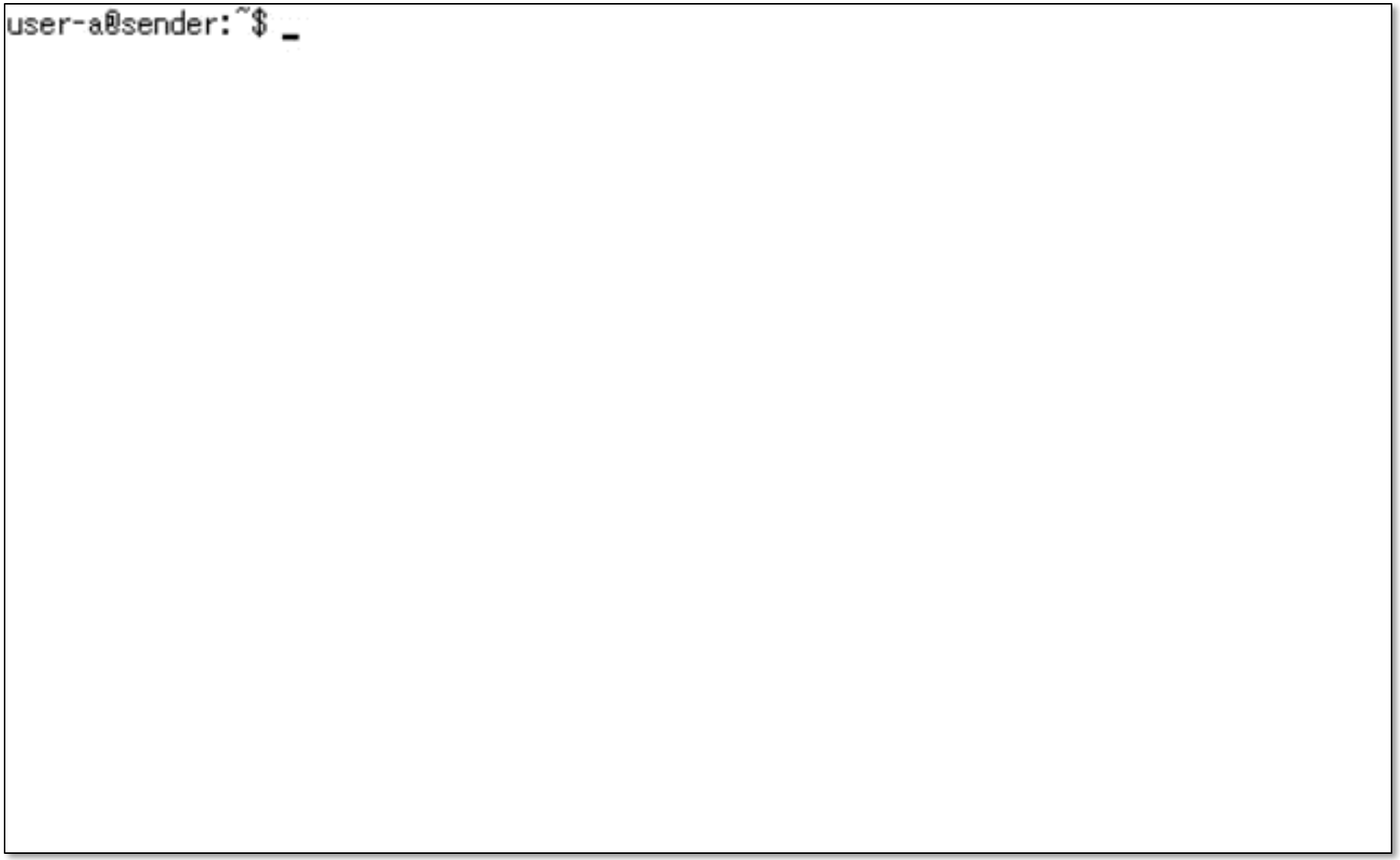
4. DMARC
p=quarantine



ドメイン名をかたるメールを受信先で「隔離」してもらう。

メールはどのように送られ届くのか




user-a@sender:~$ _

Fromの種類

- SMTPではテキストでやり取りされる。
- MTAとMTAの間でも同様のやり取りが行われる。
- メールの戻り先を指定するためのEnvelope Fromと受信側のメールクライアントがユーザに表示するために使われるヘッダFromには異なるものを指定することが可能である。（従来のメーリングリストの配送等）

エンベロープ From（Envelope From）

※RFC5321 送受信プロトコルに関する標準

ヘッダ From（Header From）

※RFC822 テキストメッセージの形式に関する標準

```
$ telnet server.apple.example. 25
Trying 192.168.0.1...
Connected to server.apple.example.
Escape character is '^]'.
220 apple.example ESMTP
EHLO apple.example
250-server.apple.example
.
MAIL FROM: user-a@apple.example
250 2.1.0 Ok
RCPT TO: user-b@orange.example
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Date: Mon, 07 Oct 2024 17:30:21 +0900
From: User A <user-a@apple.example>

Hello world.
.
250 2.0.0 Ok: queued as 016D3C2013A
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

受信メールでのFrom

エンベロープFromより

ヘッダFromより

受信メッセージ

Return-Path: <user-a@apple.example>
Received: from server.orange.example (server.orange.example [192.168.1.1])
by mda.orange.example with LMTPA;
Mon, 07 Oct 2024 17:30:53 +0900
Received: from server.apple.example (server.apple.example [192.168.0.1])
by server.orange.example with ESMTP
for <user-b@orange.example>; Mon, 07 Oct 2024 17:30:45 +0900
Received: from apple.example (mua.apple.example [192.168.0.2])
by server.apple.example (Postfix) with ESMTP id 016D3C2013A
for <user-b@orange.example>; Mon, 7 Oct 2024 17:30:21 +0900 (JST)
Message-ID: <21945-1728289845-0@apple.example>
Date: Mon, 07 Oct 2024 17:30:21 +0900
From: User A <user-a@apple.example>

Hello world.

エンベロープFromとヘッダFrom

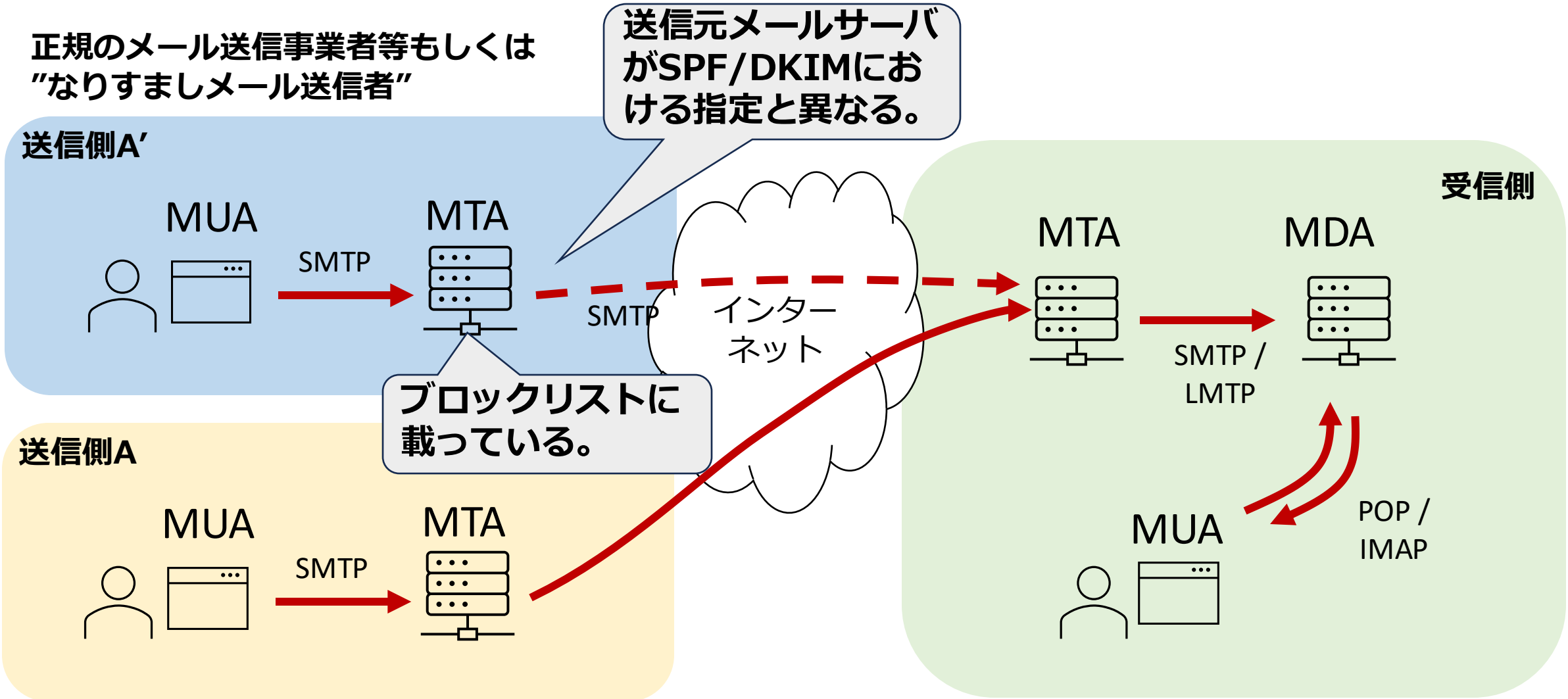
エンベロープFrom: 送信メールサーバが名乗ったメールアドレス

- SMTPコマンドではMAIL FROMコマンドで名乗る。
- エラー時の差し戻し先などとして使われる。

ヘッダFrom: メールメッセージに記載されている送信元メールアドレス

- SMTPコマンドではDATA内に記載する。
- メッセージヘッダのFrom:欄としてユーザに表示される。

届いたり届かなかったりする理由



SPFとDKIM

DMARCハンズオン勉強会 資料より

一般社団法人 日本ネットワークインフォメーションセンター

Sender Policy Framework (SPF)

（標準）RFC7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1

メール送信者のドメイン名を元に、そのメールを**送信したサーバが正しいものであるかどうかを確認**するための技術

DNSに「**v=spf1 ip4:203.0.113.0 -all**」といったTXTレコードを設定し、受信サーバはその情報を元に**送信元サーバのIPアドレスを照合**できる。

送信者と受信者におけるSPFの意義

送信者目線

- 正規の送信サーバのIPアドレスをDNS上で定義できる。
- 定義がどれだけ信頼できるかを定義できる。

受信者目線

- 受信したメールが、正規の送信者が定義したIPアドレスから来たものか判断できる。
- 正規のIPアドレスでなければ"なりすましメール"の可能性を疑える。

SPFのレコード例

nic.ad.jp. 86400 IN TXT "v=spf1 ip4:202.12.30.0/24 ip4:192.41.192.0/24 ip6:2001:dc2::/32 include:spf1.nic.ad.jp -all"

DNSのSPFに関わるTXTレコード

SPFメカニズム	目的
ipv4	IPv4アドレスを指定する。
ipv6	IPv6アドレスを指定する。
a / mx	あるA/MXレコードに紐づくドメイン名を指定する。
include	SPFレコードがTXTに記述された他のドメイン名を指定する。
all	記述されていない他のすべての条件を指定する。

SPFの判定例

```
nic.ad.jp. 86400 IN TXT "v=spf1 ip4:202.12.30.0/24
ip4:192.41.192.0/24 ip6:2001:dc2::/32 include:spf1.nic.ad.jp -
all"
```

DNSのSPFに関わるTXTレコード

エンベロープ from	ヘッダfrom	送信元IPアドレス	判定
nic.ad.jp	**nic.ad.jp**	**202.12.30.1**	**pass**
nic.ad.jp	nic.ad.jp	202.1.209.100	permfail
nic.ad.jp	**nic.jp**	**202.12.30.1**	**pass**
nic.ad.jp	nic.jp	202.1.209.100	permfail

ドメイン名が異なっていても、送信元IPアドレスが照合できればOKと判定する。(SPFにおいては)

DomainKeys Identified Mail (DKIM)

（標準）RFC6376: DomainKeys Identified Mail (DKIM) Signatures

メール送信者のドメイン名を基に、そのメールを**送信したサーバが正しいものであるかどうかを確認**するための技術

送信サーバが署名をメールヘッダに付与し、受信サーバがDNSで公開された公開鍵を用いて検証する仕組みである。署名を付与するメールヘッダとして「**DKIM-Signature:**」が使われる。

送信者と受信者におけるDKIMの意義

送信者目線

正規の送信サーバが使う署名鍵(秘密鍵)に対応する検証鍵(公開鍵)をDNS上で定義できる。

メール送信時に電子署名を付加できる。

受信者目線

- 受信したメールが正規の送信サーバが使っている署名鍵で署名されているか確認できる。
- 電子署名を検証して内容が改ざんされていないか確認できる。

DKIMのレコード例

DNSのDKIMに関わるTXTレコード

selector1._domainkey.nic.ad.jp. 3600 IN TXT "v=DKIM1; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCmj3kQlgH/5RKWZxsq09T5VxjaLeffSRI3wCVC0ox7uPaHqseVV7MVHKtX0bOxILmWOIv/pD1lO/YOflCrIp0PuDTh9JavrMP0+sji/SWhhQ0V6oaaUWJLvEQQ6NwRy/+LAHN4EqTCK5bMTuKsncfplUaUQtjCGz37/uqxqcrQZwIDAQAB"

p=の文字列がDKIM署名に使われた鍵ペアの検証鍵

DKIMの判定例

DKIM署名が記載されたメールヘッダで指定されているドメイン名(Signing Domain Identifier - SDID)

ヘッダfrom	署名ドメイン名	署名検証	判定
nic.ad.jp	**nic.ad.jp**	**成功**	**pass**
nic.ad.jp	nic.ad.jp	失敗	fail
nic.jp	**nic.ad.jp**	**成功**	**pass**
nic.jp	nic.ad.jp	失敗	fail

ドメイン名が異なっていても、署名検証に成功すればOKと判定する。(DKIMにおいては)

DMARC

一般社団法人 日本ネットワークインフォメーションセンター

Domain-based Message Authentication, Reporting, and Conformance (DMARC)

(標準) RFC7489: Domain-based Message Authentication, Reporting, and Conformance (DMARC)

SPFやDKIMを使用してメール送信サーバの正しさを検証し、"なりすましメール"対策を**強化する**ための技術

送信者のDNSで**DMARCポリシーを公開し、受信サーバがそれを参照してメールの処理方法(ディスポジション)を決定する。**「**v=DMARC1; p=reject**」といったDNSのTXTレコードが使用される。

送信者と受信者におけるDMARCの意義

送信者目線

受信者がSPFやDKIMの検証に失敗したときに、そのメールをどう扱うべきか定義できる。

アラインメントの概念でSPFやDKIMの効果を強化できる。

受信者がどのような判定を行ったかフィードバックをもらえる。

受信者目線

- SPFやDKIMの検証に失敗したときにどう扱えばよいのかを、ドメイン名ごとに確認・適用できる。
  - 隔離すべきか拒否すべきかなどを判定できる。

送信者のドメイン名の”なりすまし”対策そのものはSPFとDKIMで行うことができる。DMARCはこれらに基づく**メールの隔離や拒否の方針をDNSで伝達する**ことと、**第三者による”なりすまし”行為の発生状況を、メール数の割合で把握することができる**ところに意義がある。

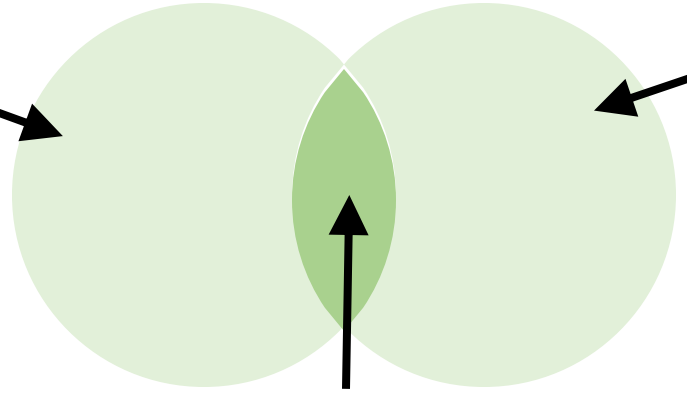
DMARCの判定基準

SPF判定 pass

エンベロープFromのドメイン名にあるSPFレコードを元に判定

SPFアラインメント pass

ヘッダFromのドメイン名とエンベロープFromが一致するかどうかで判定
つまりヘッダFromだけをみればよい状態になる。



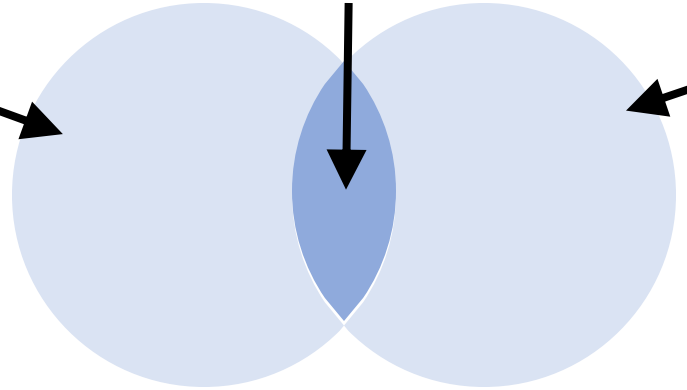
どちらかを満たせば DMARC pass

DKIM判定 pass

DKIMレコードと署名ドメイン名を元に判定

DKIMアラインメント pass

ヘッダFromのドメイン名と署名ドメイン名が一致するかどうかで判定
つまりヘッダFromだけをみればよい状態になる。



RFC7489: Domain-based Message Authentication, Reporting, and Conformance (DMARC) より

アラインメント

アラインメントとは、メールのヘッダFromのドメイン名とSPFやDKIMで指定されたドメイン名が一致していることを確認する仕組み

※DMARCプロトコルにおける概念であり、SPFとDKIMのプロトコルには含まれない。

用語	意味
SPFアラインメント （SPFに関するアラインメント）	ヘッダFrom とエンベロープFrom のドメイン名が一致するか
DKIMアラインメント （DKIMに関するアラインメント）	ヘッダFrom ドメイン名と署名ドメイン名 (d=) が一致するか

SPF/DKIMの判定とDMARCの判定

エンベロープ from	ヘッダfrom	送信元IPアドレス	SPF判定	アラインメント	DMARC判定
nic.ad.jp	**nic.ad.jp**	202.12.30.1	**pass**	**pass**	**pass**
nic.ad.jp	nic.ad.jp	202.1.209.100	permfail	pass	fail
nic.ad.jp	**nic.jp**	202.12.30.1	**pass**	**fail**	fail
nic.ad.jp	nic.jp	202.1.209.100	permfail	fail	fail

署名ドメイン名	ヘッダfrom	署名検証	DKIM判定	アラインメント	DMARC判定
nic.ad.jp	**nic.ad.jp**	**成功**	**pass**	**pass**	**pass**
nic.ad.jp	nic.ad.jp	失敗	fail	pass	fail
nic.ad.jp	**nic.jp**	**成功**	**pass**	**fail**	fail
nic.ad.jp	nic.jp	失敗	fail	fail	fail

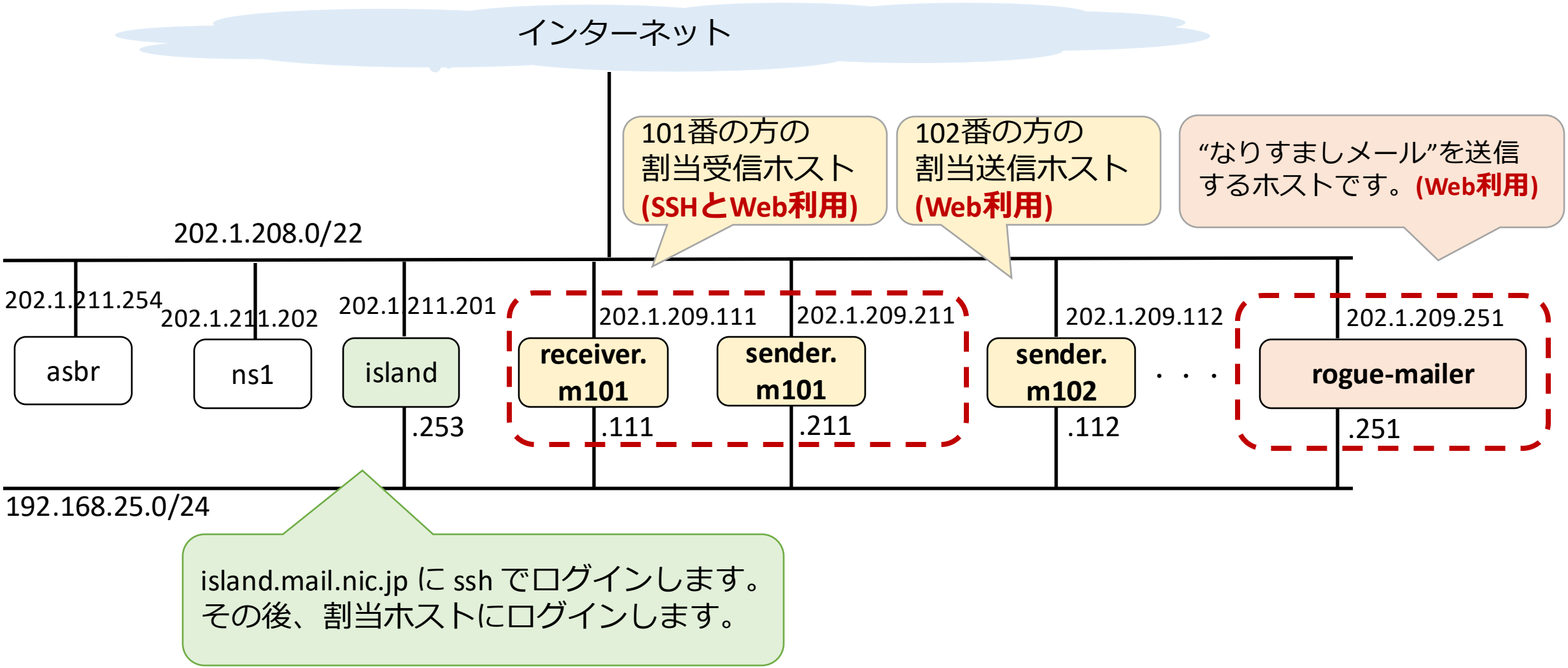
DMARCレポート

種類	DNSでの指定	概要	業務上の意味
集約(Aggregate)レポート	rua=	受信側がある期間に受け取った該当ドメイン宛・該当ドメイン差出メールの、SPF／DKIM認証・アライメント・ポリシー適用状況などを集計して、ドメイン所有者に報告するもの。	送信元IPアドレスとSPF, DKIM, DMARCの判定結果や数の可視化。 自組織のドメイン名に対する”なりすまし”や設定漏れなど。
失敗(Failure)レポート	ruf=	認証・整合性チェックに失敗した個別メッセージについて、受信側が詳細をドメイン所有者にフィードバックすることを意図したもの。	個々のなりすましに関する情報源になる。 プライバシーの懸念。

ハンズオン

一般社団法人 日本ネットワークインフォメーションセンター

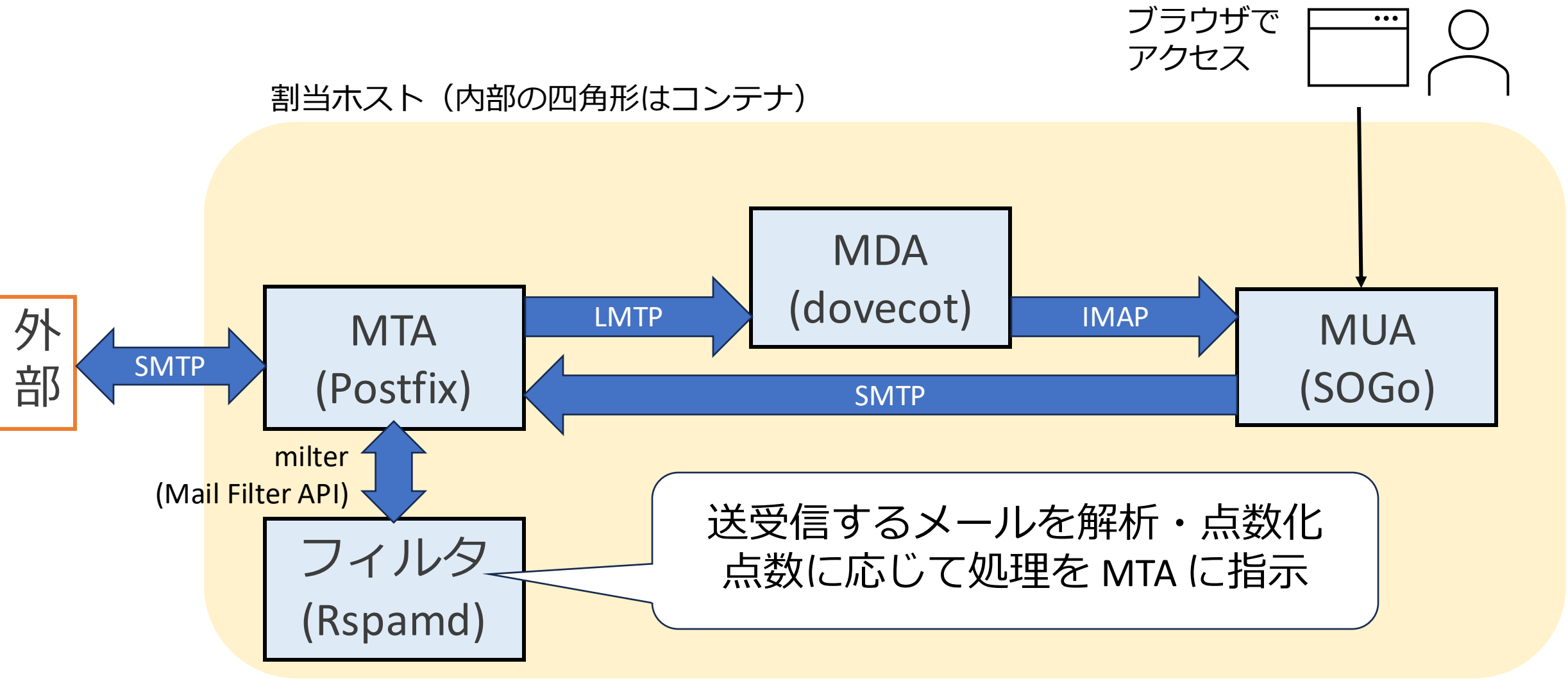
DMARCハンズオン環境



割当各ホストの役割

- **割当ゾーン(mXXX.mail.nic.jp.)のDNS権威サーバ (knot)**
  - 割当受信ホストのみ
- **割当ゾーンのメールサーバ (mailcow)**
  - MTA (Postfix on docker)
  - MDA (dovecot on docker)
  - MUA (SOGo on docker)
  - メールフィルタ (Rspamd on docker)
- **DNSフルサービスリゾルバ (Unbound)**

割当ホスト内のメールフロー



Rspamdでのフィルタの概念

- **Rspamdでのフィルタの概念 (spamassasine とほぼ同じ)**
  - **シンボル:** ある観点から悪性の可能性がどれだけあるか予め定義
    - SPF, DKIM, DMARC, ARC, アドレス/ドメイン名レピュテーション...
  - **スコア:** ルールごとに悪性の可能性があれば加点
  - **最終的な判定 (disposition):** スコア閾値を設定
    - Discard, Quarantine, Add header, Graylist などなど
    - 通過するメールに対しシンボルを判定，累積スコアでdispositionを決定
  - **DMARCはあくまでシンボルの1つとして扱う**
    - あるシンボルに引っかかった時点で強制的にdispositionを決定もできる

0. 準備編 - 実験環境の利用

0-1. SSHログイン
0-2. DNSサーバの確認
0-3. DNSレコードの確認

0-1. SSHログイン

■割り当て**受信**ホストまでログインしてみましょう。

(リモート)$ **ssh user-a@island.mail.nic.jp**

island$ **ssh receiver.mXXX.mail.nic.jp**

■sudoします。
receiver$ sudo –s
receiver#

割当受信ホスト　receiver.mXXX.mail.nic.jp

0-2. DNSサーバの確認

■割当のDNSゾーンの権威サーバ(knot)が動作していることを確認します

receiver# **systemctl status knot.service**

● knot.service - Knot DNS server
Loaded: **loaded** (/usr/lib/systemd/system/knot.service; enabled; preset: enabled)
Active: **active (running)** since Thu 2025-10-30 14:58:23 JST; 1 day 1h ago

0-3. DNSレコードの確認

■割当のDNSゾーンで公開しているレコードを確認します

receiver# **cat /var/lib/knot/mXXX.mail.nic.jp.zone**

```
mXXX.mail.nic.jp.            20    SOA    (省略)
mXXX.mail.nic.jp.            20    NS     ns.mXXX.mail.nic.jp.
mXXX.mail.nic.jp.            20    MX     10 receiver.mXXX.mail.nic.jp.
MXXX.mail.nic.jp.            20    TXT    "v=spf1 ip4:202.1.209.YYY -all"
receiver.mXXX.mail.nic.jp.   20    A     202.1.209.XXX
sender.mXXX.mail.nic.jp.     20    A     202.1.209.YYY
ns.mXXX.mail.nic.jp.         20    A     202.1.209.XXX
```

DMARC等導入前のハンズオン

- **メール環境の利用**
  - Webメールを使ってみます。
- **“なりすましメール”の送信とメールヘッダの確認**
  - なりすましメール”を送信し、メールヘッダの変化を確認します。

1. SPF/DKIM/DMARC導入前 - "なりすましメール"の送受信

1-1. rogue-mailerから"なりすましメール"を送信
1-2. 割当受信ホストで"なりすましメール"を受信
1-3. "なりすましメール"のヘッダ・解析結果を確認

WebブラウザでWebMail利用

送信側

■ **“なりすましメール”送信ホスト**	https://mx.rogue-mailer.mail.nic.jp/
ユーザ名	user-a@rogue-mailer.mail.nic.jp
■ **割当受信ホスト** ■ **割当送信ホスト**	https://receiver.mXXX.mail.nic.jp/ https://sender.mXXX.mail.nic.jp/
ユーザ名	user-a@mXXX.mail.nic.jp

xxxは割り当てられた番号です。

※当日のみアクセス可能です

送信側

1-0. rogue-mailerのSOGoにログイン

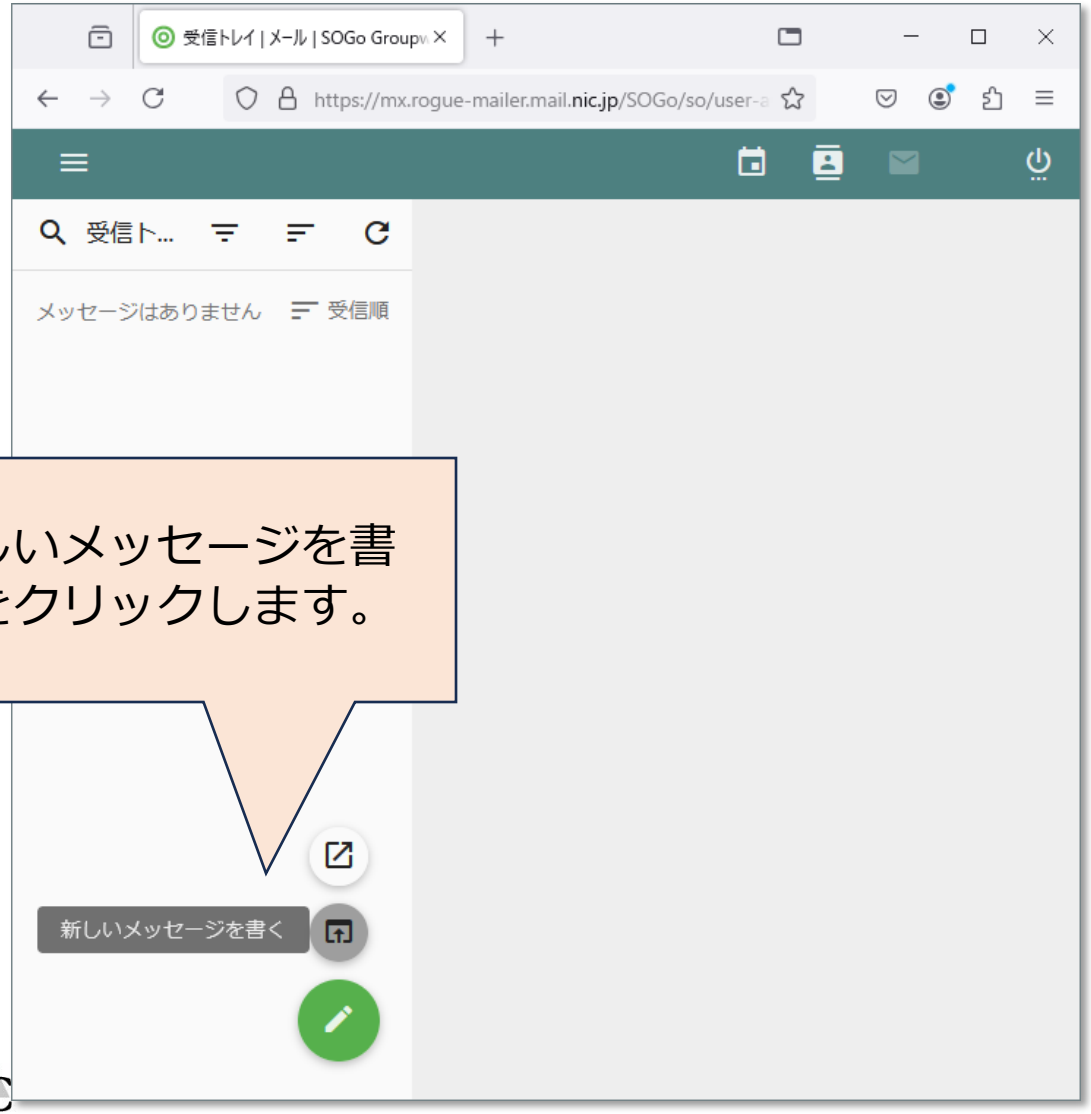
https://mx.rogue-mailer.mail.nic.jp/
にアクセスします。

ユーザ名　**user-a@rogue-mailer.mail.nic.jp**

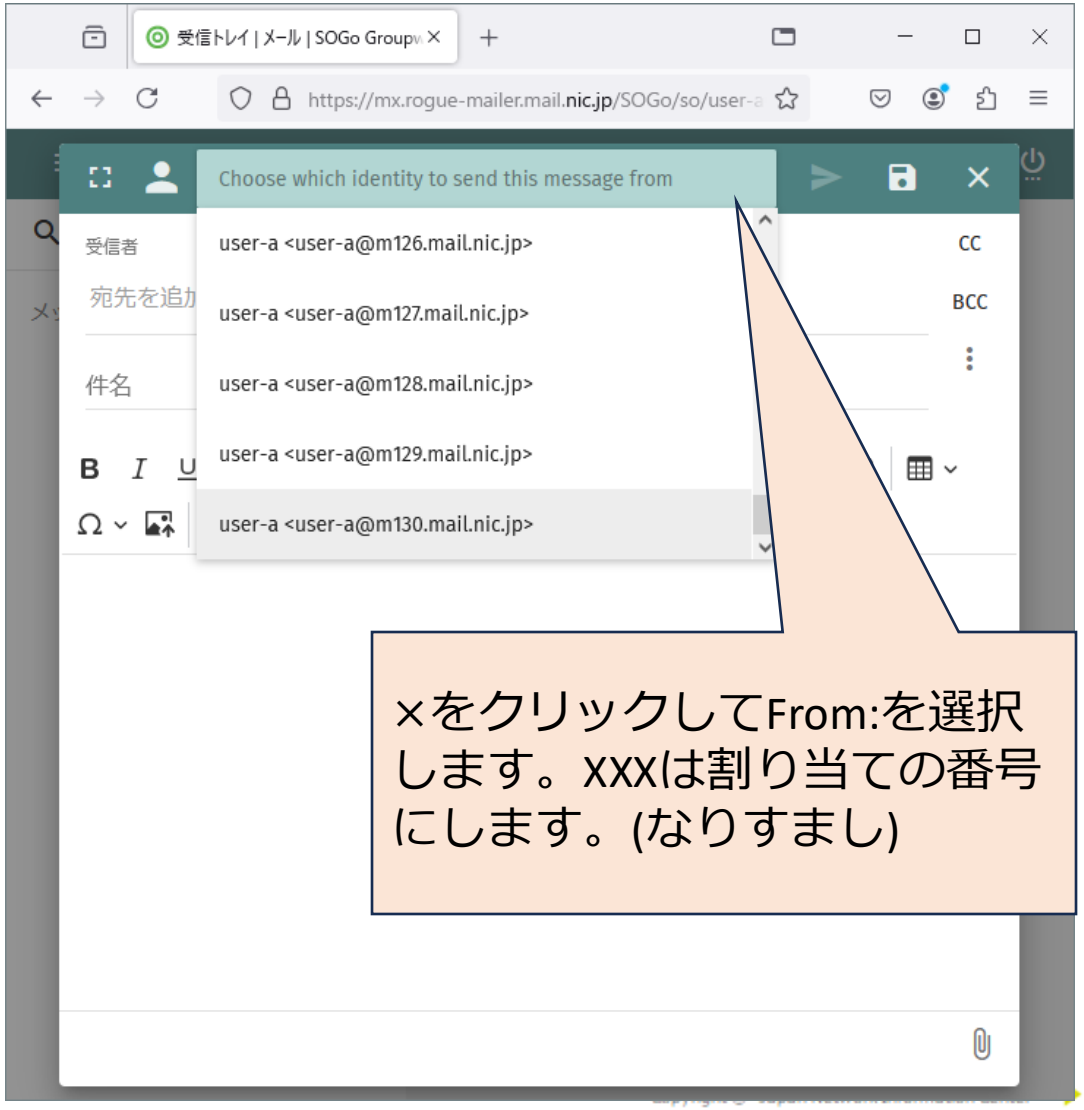
※SOGoのWebGUIはSOGoによるものです。以降同様。

送信側

1-1. rogue-mailerから”なりすましメール”を送信



「新しいメッセージを書く」をクリックします。



×をクリックしてFrom:を選択します。XXXは割り当ての番号にします。(なりすまし)

1-1. rogue-mailerから”なりすましメール”を送信

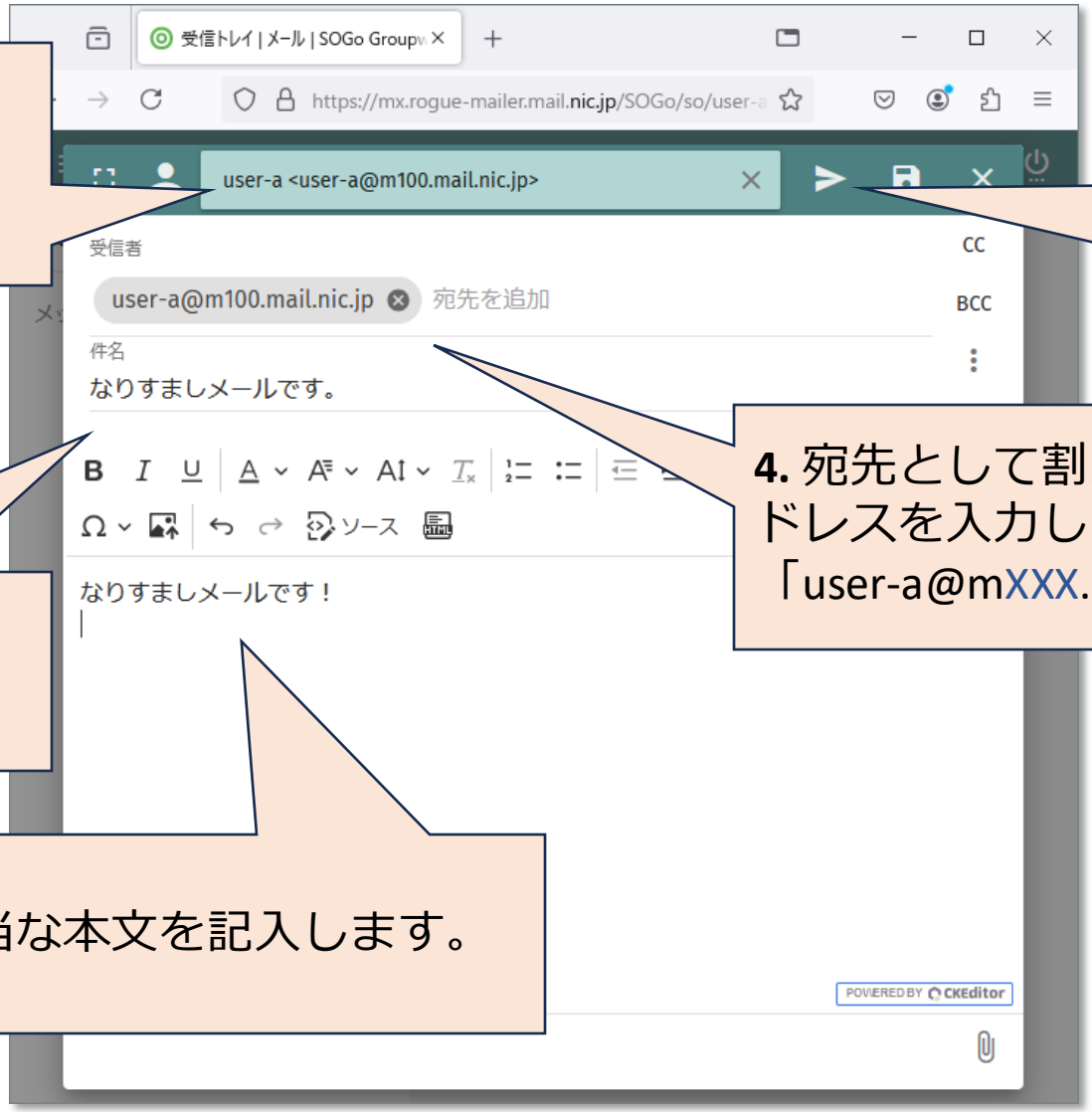
1. 送信元として割当のメールアドレスを選択します。
「user-a@mXXX.mail.nic.jp」

2. 件名をなりすましメールであることが分かるようにします。

3. 適当な本文を記入します。

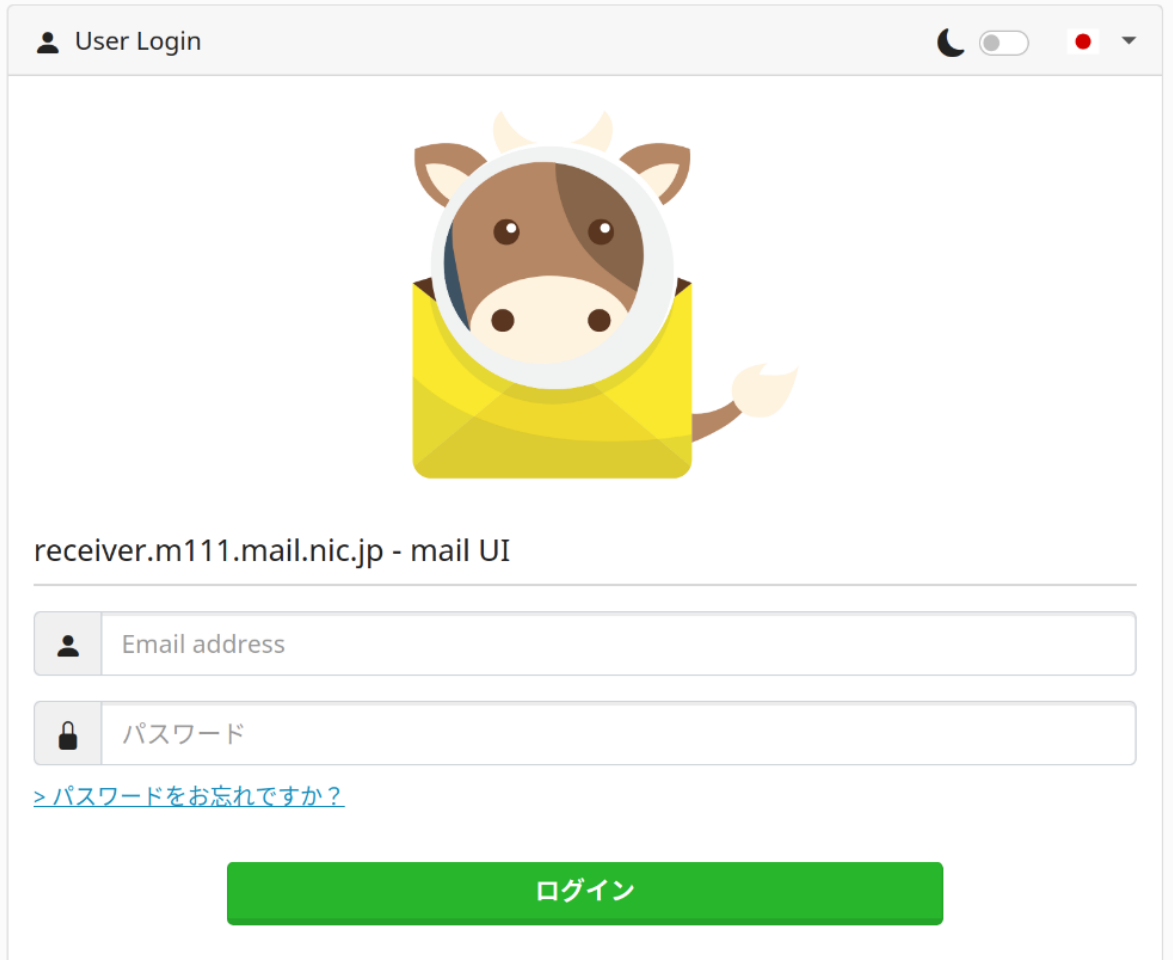
4. 宛先として割当のメールアドレスを入力します。
「user-a@mXXX.mail.nic.jp」

5. 最後に送信ボタンを押します。



受信側

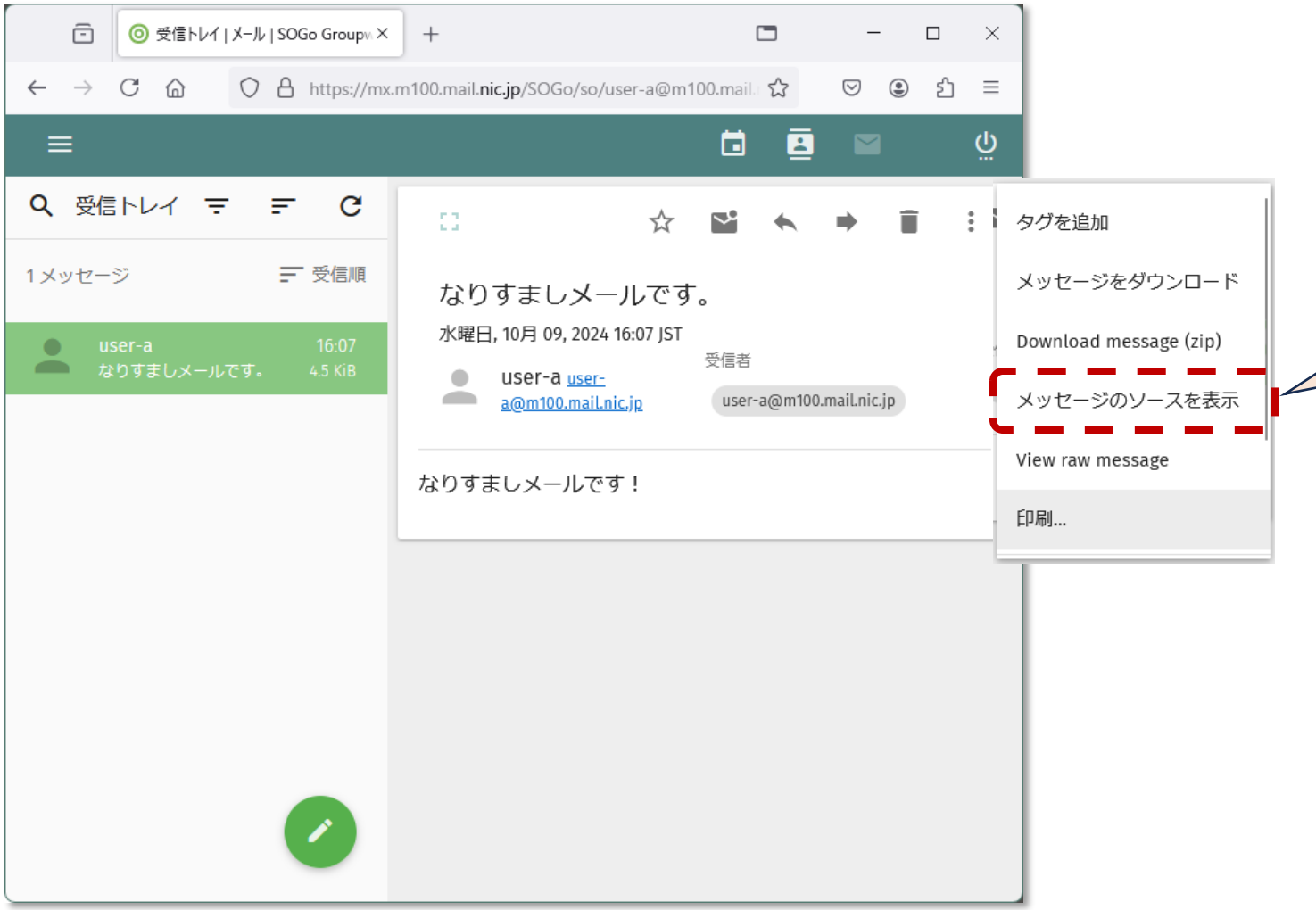
1-2. 割当受信ホストで"なりすましメール"を受信



https://receiver.mXXX.mail.nic.jp/
にアクセスします。

ユーザ名　**user-a@mXXX.mail.nic.jp**

1-2. 割当受信ホストで"なりすましメール"を受信



「メッセージのソースを表示」をクリックします。

1-3. "なりすましメール"のヘッダ・解析結果を確認

```
Return-Path: <user-a@m100.mail.nic.jp>
Delivered-To: user-a@m100.mail.nic.jp
Received: from receiver.m100.mail.nic.jp ([172.22.1.253])
        by 98ef6cf32de9 with LMTP
        id CFnpFLQrBmcm4QAACMyhBw
        (envelope-from <user-a@m100.mail.nic.jp>)
        for <user-a@m100.mail.nic.jp>; Wed, 09 Oct 2024 16:07:32 +0900
Received: from mx.rogue-mailer.mail.nic.jp (unknown [202.1.209.251])
        (using TLSv1.3 with cipher TLS_AES_256_GCM_SHA384 (256/256 bits)
         key-exchange X25519 server-signature RSA-PSS (4096 bits))
        (No client certificate requested)
        by receiver.m100.mail.nic.jp (Postcow) with ESMTPS id 47E67609A8
        for <user-a@m100.mail.nic.jp>; Wed,  9 Oct 2025 16:07:28 +0900 (JST)
Authentication-Results: receiver.m100.mail.nic.jp;
        dkim=none;
        spf=none (receiver.m100.mail.nic.jp: domain of user-a@m100.mail.nic.jp has no SPF policy when checking
202.1.209.251) smtp.mailfrom=user-a@m100.mail.nic.jp;
        dmarc=none
 :
```

エンベロープFromの値が入っている。

receiver.m100.mail.nic.jpが「Authentication-Results:」ヘッダを付与した。

DKIMの署名がなく、かつSPFのDNSレコードがなく、更にDMARCのDNSレコードがないことを示している。

1-3. "なりすましメール"のヘッダ・解析結果を確認

From: "user-a" <user-a@m100.mail.nic.jp>
To: user-a@m100.mail.nic.jp
User-Agent: SOGoMail 5.11.0
MIME-Version: 1.0
Date: Wed, 09 Oct 2025 16:07:24 +0900
Subject: =?utf-8?q?=E3=81=AA=E3=82=8A=E3=81=99=E3=81=BE=E3=81=97=E3=83=A1?=
 =?utf-8?q?=E3=83=BC=E3=83=AB=E3=81=A7=E3=81=99=E3=80=82?=
Message-ID: <45-67062b80-b-3af34340@209023568>
Content-Type: multipart/alternative; boundary="----=_=-_OpenGroupware_org_NGMime-69-1728457644.659308-2------"
X-Last-TLS-Session-Version: None
X-Last-TLS-Session-Version: TLSv1.3
X-Spamd-Result: **default: False [3.49 / 15.00];**
RDNS_NONE(2.00)[];
AUTH_NA(1.00)[];
MID_RHS_NOT_FQDN(0.50)[];
ONCE_RECEIVED(0.10)[];
MIME_GOOD(-0.10)[multipart/alternative,text/plain];
MX_GOOD(-0.01)[];
DMARC_NA(0.00)[nic.jp];
MAILCOW_DOMAIN_HEADER_FROM(0.00)[m100.mail.nic.jp];

ヘッダFromの値が入っている。

Rspamdが総合的なスコアとして3.49を付けており、閾値の15.00未満であることから、何もせずに配送されている。
（False=ディスポジションNoneの意味）

DMARCのDNSレコードがないことを示している。

1-3. "なりすましメール"のヘッダ・解析結果を確認

```
RCPT_MAILCOW_DOMAIN(0.00)[m100.mail.nic.jp];
BCC(0.00)[];
DIRECT_TO_MX(0.00)[SOGoMail 5.11.0];
MIME_TRACE(0.00)[0:+,1:+,2:~];
RECEIVED_HELO_LOCALHOST(0.00)[];
ARC_NA(0.00)[];
RCPT_COUNT_ONE(0.00)[1];
HFILTER_HOSTNAME_UNKNOWN(0.00)[];
TO_MATCH_ENVRCPT_ALL(0.00)[];
RCVD_TLS_ALL(0.00)[];
R_SPF_NA(0.00)[no SPF record];
TO_DN_NONE(0.00)[];
FROM_EQ_ENVFROM(0.00)[];
FROM_HAS_DN(0.00)[];
ARC_SIGNED(0.00)[m100.mail.nic.jp:s=dkim:i=1];
RCVD_VIA_SMTP_AUTH(0.00)[];
R_DKIM_NA(0.00)[];
RCVD_COUNT_ONE(0.00)[1];
ASN(0.00)[asn:131971, ipnet:202.1.208.0/22, country:JP];
TO_EQ_FROM(0.00)[]
X-Rspamd-Queue-Id: 47E67609A8
```

SPFのDNSレコードがないことを示している。

受信メールにDKIM署名がなされていないことを示している。

受信側

1-3. "なりすましメール"のヘッダ・解析結果を確認

```
------=_=-_OpenGroupware_org_NGMime-69-1728457644.659308-2------
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Length: 103

=E3=81=AA=E3=82=8A=E3=81=99=E3=81=BE=E3=81=97=E3=83=A1=E3=83=BC=E3=83=AB=
=E3=81=A7=E3=81=99=EF=BC=81

------=_=-_OpenGroupware_org_NGMime-69-1728457644.659308-2------
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
Content-Length: 122

<html><p>=E3=81=AA=E3=82=8A=E3=81=99=E3=81=BE=E3=81=97=E3=83=A1=E3=83=BC=
=E3=83=AB=E3=81=A7=E3=81=99=EF=BC=81</p></html>

------=_=-_OpenGroupware_org_NGMime-69-1728457644.659308-2--------
```

メールの本文（MIMEエンコードされている。）

送信側

4. DMARCの導入 - 送信者編

1. DMARCレコードの公開
   ※SPFによるDMARC判定を確認します。
2. rogue-mailerから"なりすまし"メールの送信
3. “なりすまし”メールの受信
   ※p=quarantine の状況を確認します。

送信側

1. DMARCレコードの公開(p=quarantine)

■割当受信ホスト(兼DNS権威サーバ)にSSHログインしゾーンファイルにレコードを記述します。

receiver# knotc zone-freeze mXXX.mail.nic.jp
receiver# vi /var/lib/knot/mXXX.mail.nic.jp.zone

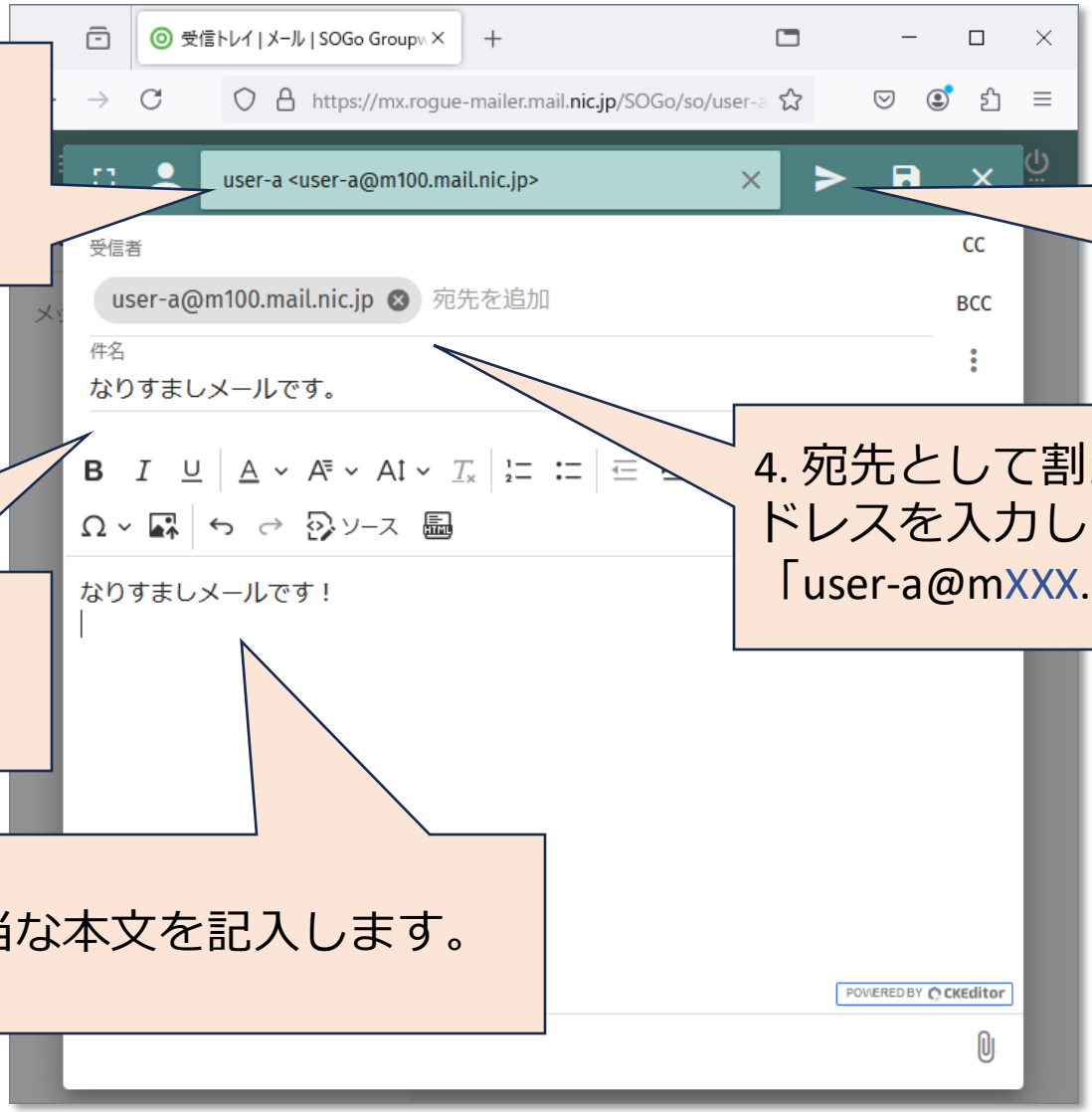
〇SOAシリアルを増加させます
「**2025103003**１１１１」 → 「**2025103004**１１１１」
〇変更する内容:
_dmarc.mXXX.mail.nic.jp.　20　TXT　 "v=DMARC1; p=**quarantine**;"

■knotのコマンドでゾーン情報を読み込みます。

receiver# knotc zone-thaw mXXX.mail.nic.jp
receiver# knotc zone-reload mXXX.mail.nic.jp
OK

2. rogue-mailerから"なりすまし"メールの送信

1. 送信元として割当のメールアドレスを選択します。
「user-a@mXXX.mail.nic.jp」



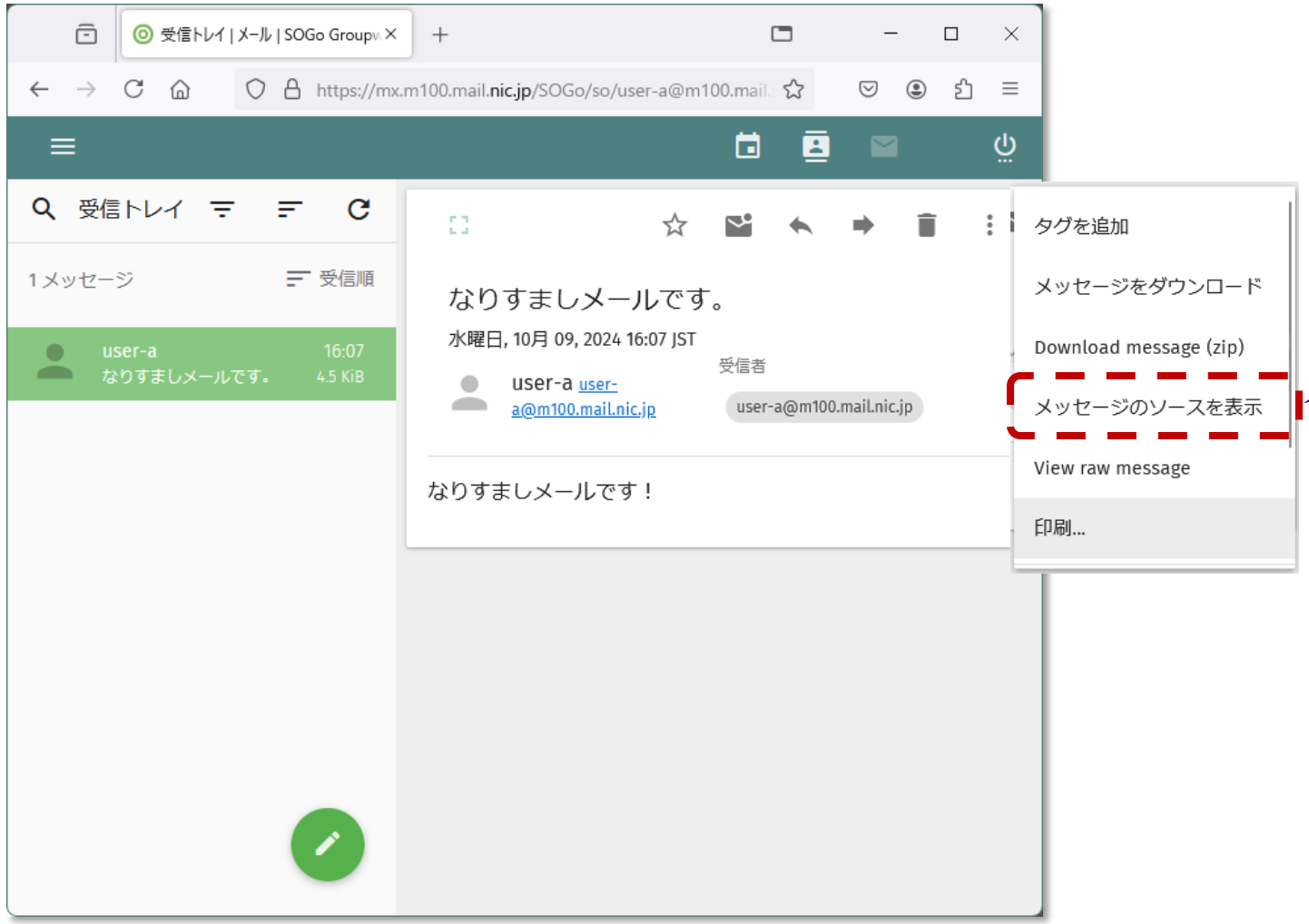
2. 件名をなりすましメールであることが分かるようにします。

3. 適当な本文を記入します。

4. 宛先として割当のメールアドレスを入力します。
「user-a@mXXX.mail.nic.jp」

5. 最後に送信ボタンを押します。

3. "なりすまし"メールの受信



「メッセージのソースを表示」をクリックします。

DMARCポリシーと業務上の「起きること」

- **p=none**
  - 正規メール・DMARC的なりすましメールは通常通り配送
  - DMARCレポートあり
- **p=quarantine**
  - 正規メールは通常通り配送
  - DMARC的なりすましメールは迷惑メール等へ
    - 正規メールなのに設定
  - DMARCレポートあり
- **p=reject**
  - ドメイン名のなりすましはない状態。
  - DMARCレポートあり

ハンズオン/ディスカッション：
「迷惑メール」フォルダはみていただけるか

一般社団法人 日本ネットワークインフォメーションセンター

ハンズオン/ディスカッション：
p=quarantine の時代の過ごし方

一般社団法人 日本ネットワークインフォメーションセンター

(体験談より)
DMARCレポート分析 - オープンソース v.s. サービス利用

- p=quarantineにする際にはDMARCレポート分析が重要。ただしこのためにコストは...まずはオープンソースで。
- ふたを開けてみると...XMLフォーマットではあるものの、各社によって内容が違っていて、しかも変化していることが判明。開発できるため、自社での対応を検討したが、キャッチアップするためのコストが...。

p=quarantineで起きる出来事に対応していくためにはDMARC分析を通じた各種の把握（DMARC観点での”なりすまし”の発生状況や本当の”なりすまし“等）のための「コスト」がかかる。人的コストを考えると必ずしもOSSが「安い」とは限らない模様。

ハンズオン/ディスカッション：お問合せ対応

一般社団法人 日本ネットワークインフォメーションセンター

ハンズオン/ディスカッション：経営的観点での判断

一般社団法人 日本ネットワークインフォメーションセンター

ここまでのまとめ

私なりの今の時点でのまとめ

- **p=noneである現状**
  - クラウド事業者/大手事業者においてこの先のp=noneの扱いが変わる可能性があることには変わらない。
  - p=quarantineが必須となった時にはじめて正規メールを届かせるための施策を取るには遅くなる可能性。事業へのインパクトを考えておく必要がある。
- **p=quarantineの間に起きることと注視したいポイント**
  - 「迷惑メール」フォルダには正規メールもなりすましメールも入る。
  - ユーザ対応・トラブルシューティング・正規メールを届けるためのための調査や連絡/設定など何らかの「コスト」はかかる。短期化が鍵か。
- **p=rejectへの判断は定量的**
  - 正規メールが届く基盤づくりができたときのゴールでもある。届かなかった時にも把握できる。送信側としては「できることはやった」状態。

ポイント
- DMARCレポート分析
- 送信サーバの把握　DNS管理部門連携
- 窓口や社内連携

ディスカッション

一般社団法人 日本ネットワークインフォメーションセンター

ご参加くださりありがとうございました。

JPNIC

一般社団法人 日本ネットワークインフォメーションセンター