迷惑メール対策を中心とした メールシステム構築
~闘うシステム管理者編~

安藤一憲
ando@iri-com.co.jp

このチュートリアルの構成

- メールの基礎知識
  - 普段目にするメールについての解説
  - 基本的設定のまとめ
- spam対策
  - 多様な受信対策の選択
  - 発信させない対策

DNSの重要性

- あるドメインのリソース情報へ到達する鍵
  - 順次NSを手繰ってデータを引きに来る仕組み
    - 手繰れないと破綻
  - IPアドレス付け替え時、ドメイン変更時に注意
    - TTLを一時的に小さくして対処
  - ほぼ全てのサービスに影響
    - FQDNを用いるもの全て
  - NSを死守すべし
    - 全ての基礎となるサービスのひとつという位置付け

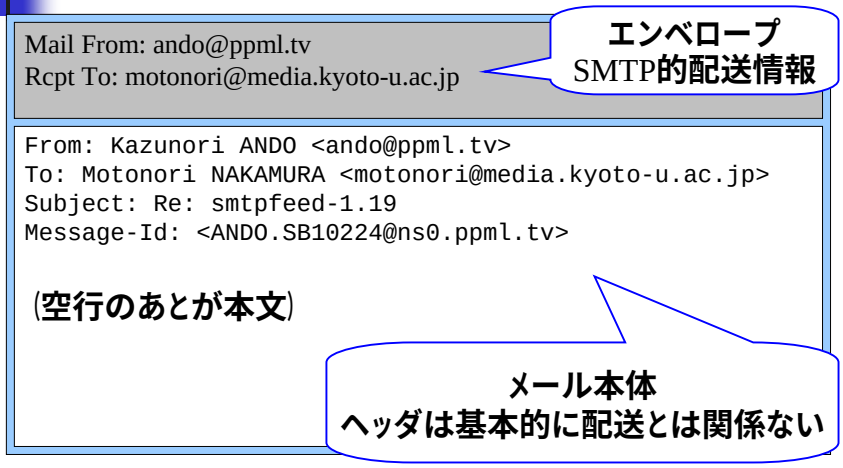
DNSとメール

- user@example.gr.jp
  - ここから配送先をどう見つけるか?
  - 手がかりはドメイン名の部分
- example.gr.jpのMXレコードを調べる
  - 配送にはMXとサーバのAレコードが必要
    - 最も効率が良いのは、MXを聞いたらMXだけではなくAが同時に返ってくる場合
    - 答えるnamedがMXとAを両方知っているのがベスト

MXはCNAMEではいけない

- O DontExpandCnames=False
  - RFC822,1123的にはたぐるのが正しい
  - IETFはCNAMEをたぐらない方向に動いている
  - sendmailではオプションになった
- DNSのMXにはCNAMEを指定してはいけない
  - RHSにはAを書く
  - そのAはMXを答えるnamedが知っていると良い

メール本体とエンベロープ



メール本体とエンベロープ (2)

Mail From: motonori@media
Rcpt To: ando@ppml.tv

配送経路情報が記録される
(記述形式は任意)

```
Received: from query.media.kyoto-u.ac.jp
        by ns0.ppml.tv with ESMTP
        for <ando@ppml.tv>; 3 Dec 2004 10:00:01 +0900
Return-Path: motonori@media.kyoto-u.ac.jp
From: Motonori NAKAMURA <motonori@media.kyoto-u.ac.jp>
To: Kazunori ANDO <ando@p      tv>
Subject: Re: smtpfeed-1.19
Message-Id: <ANDO.SB102340
```

(空行のあとが本文)

Return-Path:にSMTPのMail From:が
保存される (設定による)

メール本体とエンベロープ (3)

- To: に書くとそこに送られるのは…
  - 実はMUAの仕業
  - To: ヘッダに書いたアドレスをSMTP的な配送先情報としてMTAに渡しているだけ
  - 例えば、To:ヘッダがメーリングリストのアドレスなのに自分にメールが届くのはこのため

ヘッダの話 (1)

- Field-name: Field-body (standard)
  - From: 差出人アドレス
  - Sender: 差出人アドレスが不明確な場合に差出人を明示
  - To: 宛先アドレス
  - Cc: カーボンコピー
  - Reply-To: 返信先アドレス
  - Message-Id: 5年間固有のID
  - Subject: タイトル
  - Date: 差出時間
  - Return-Path: エラー返信先アドレス

ヘッダの話 (2)

- Field-name: Field-body (standard)
  - Received: 配送経路
  - In-Reply-To: どのメールに返信したかを示す
  - References: どのメールに返信したかを示す
- Resent系 (メールを再送信する場合の)
  - Resent-From: 差出人アドレス
  - Resent-Sender: 差出人アドレスが不明確な場合に明示
  - Resent-Reply-To: 返信先アドレス
  - Resent-Message-Id: 5年間固有のID
  - Resent-Date: 再送信日時

ヘッダの話（3）

- Field-name: Field-body
  - Precedence: 配送優先度
  - X-Authentication-Warning: アドレス詐称（？）
- （おまけ）MLドライバ等の付けるヘッダ
  - X-MLServer: fml
  - X-ML-System: ppml
  - X-Ml-Version: kkml
  - X-Distribute: distribute
  - Delivered-To: qmail等

文字の話

- 機種依存文字を使ってはいけない
  - ①②③④⑤⑥⑦⑧⑨⑩
  - Ⅰ Ⅱ Ⅲ Ⅳ Ⅴ Ⅵ Ⅶ Ⅷ Ⅸ Ⅹ
  - ﾄｳｾﾞﾝﾊﾝｶｸｶﾀｶﾅﾓﾀﾞﾒ
- 漢字コードはISO-2022-JPを使用する
  - SJISだめ、EUCだめ、UNICODEだめ
- さらなる制約もある
  - 例えばISO-8859-1な文字（ウムラウト付き文字等）とISO-2022-JPな漢字はメール上で混在できない

配送の実際 (1)

- 宛先アドレスのMXを引いてみる

MX 10 mail-g1.example.gr.jp
MX 10 mail-g2.example.gr.jp

- この場合はランダムでどちらかに配送

MX 10 mail-g1.example.gr.jp
MX 20 mail-g2.example.gr.jp

- この場合は10の方に配送して駄目だったら20へ

配送の実際 (2)

- MX RRの他にも答がいっぱい返ってくる

```
example.gr.jp   MX  10   mail-g1.example.gr.jp
example.gr.jp   MX  20   mail-g2.example.gr.jp
```

MX RR

```
example.gr.jp   NS   ns1.example.gr.jp
example.gr.jp   NS   ns2.example.gr.jp
mail-g1.example.gr.jp   A   202.250.31.150
mail-g2.example.gr.jp   A   202.250.31.151
ns1.example.gr.jp    A  202.250.31.148
ns2.example.gr.jp    A  202.250.31.149
```

additional information

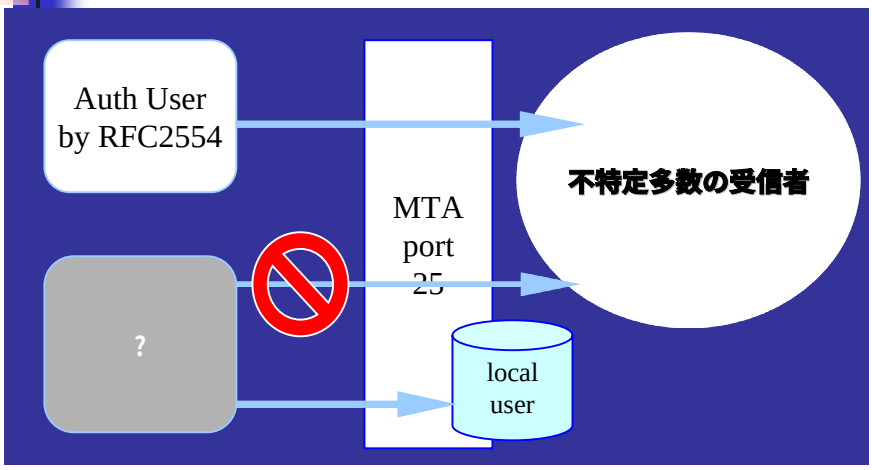
実際の配送 (3)

- Additional Information
  - MXを聞かれたNSがMXのAも知っている
    - MXとAがわかれば実際に接続しにいける
    - 1回のqueryで済むので効率が良い
    - MX RRを保持しているNSがAも保持することが重要
  - Additional InformationとMTA
    - 例えばSMTPfeedはAdditional Informationを利用
    - MTAが利用しなくても手元のnamedがcache
      - Aを聞きに行くとそこが答えるので速い

SMTP Authentication (RFC2554)

- SASL (RFC2222) を利用したRelay認証
  - sendmail-8.13では
    - 必要な作業
      - cyrus SASLライブラリをインストール
      - SASLを利用するようにsendmailをコンパイル
      - /usr/local/lib/sasl/Sendmail.confの準備 (必要なら)
      - /etc/sasldb.dbの準備 (saslpasswdコマンドでユーザ登録)
      - sendmail.cfの設定追加
    - 認証を通るとそのサーバ経由のRelay配送を許可

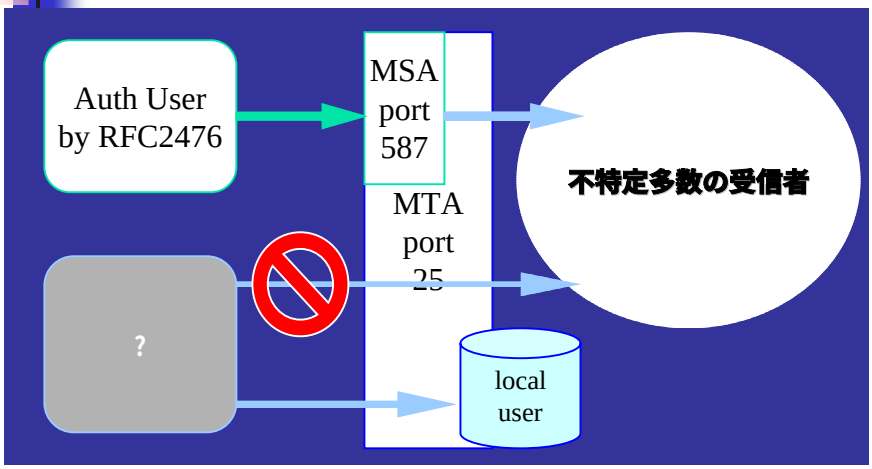
SMTP Authentication (RFC2554)



Message Submission (RFC2476)

- MSA (Message Submission Agent)
  - **メールを「出す」新たな枠組み**
    - Relayと**区別することでspam不正中継を防止**
      - SMTPでは**local宛のメールしか受けない**
      - Submission**による発信は自分のサイトからの接続だけを許可してさらに認証をかける**
    - port 587
      - sendmail-8.11**以降は**defaultでMSA**になる**
      - MSP (MessageSubmissionProgram/**クライアント**) **からの接続を受け付ける**

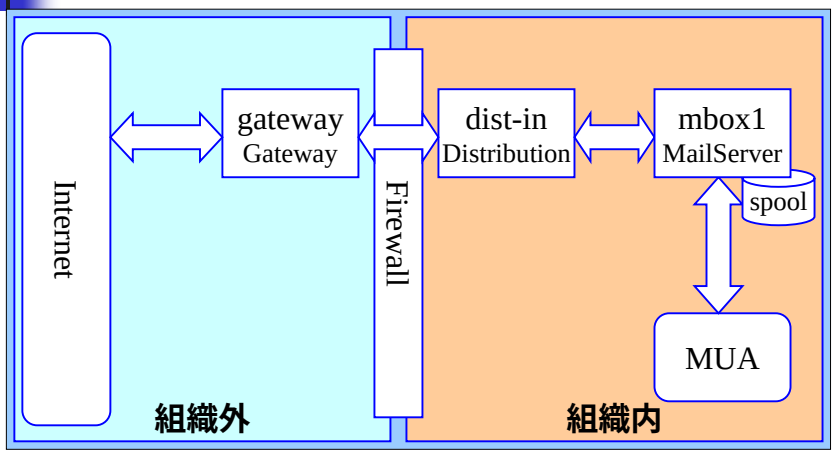
Message Submission (RFC2476)



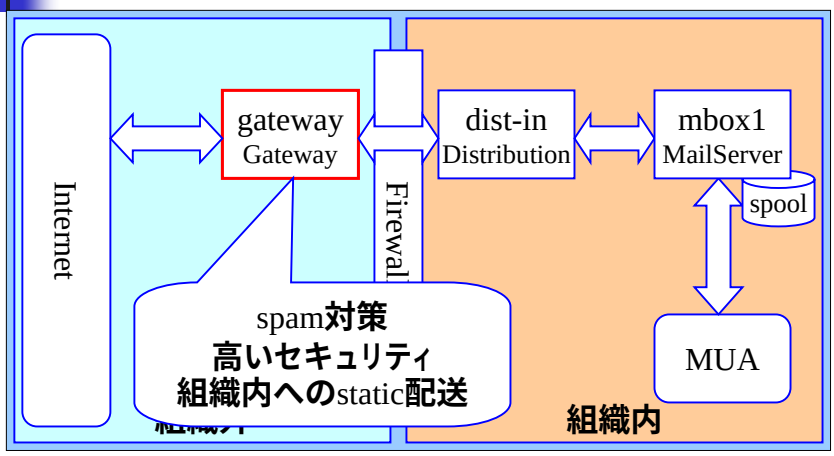
配送設定の基本要素

- MX配送かstatic (静的) 配送か?
  - 対外配送はMX配送
  - 組織内部の配送はどちらか選択
    - 組織内部で独自のDNSの定義をしている場合
    - 集中サーバならstatic(mailertable)でもいける
  - resolv.confで参照するDNSサーバを指定

配送モデル



配送モデル:Gateway



Gateway

- 特別に必要な機能
  - 内側サーバへのstatic配送
    - FEATURE(`mailertable‘)
  - スパム不正中継の防止対策
    - FEATURE(`access_db’)
    - FEATURE(`blacklist_recipients')
    - Milter

Gateway: mailertable

- static配送ルールを書く
- 設定ファイル名は/etc/mail/mailertable

/etc/mail/mailertable

```
.example.gr.jp     smtp:[dist-in.example.gr.jp]
.example.ad.jp     smtp:[non-mx.example.ad.jp]
.example.com       esmtp:mx.example.co.jp
```

```
# makemap hash /etc/mail/mailertable < /etc/mail/mailertable
```

このコマンドでmailertable.dbが生成され本設定が完了

Gateway: access_db

- 拡張されたspamlistの設定
- 設定ファイル名は/etc/mail/access

/etc/mail/access

```
spammers.net                    REJECT
spammer@ube.com                 ERROR:5.7.1:551 Relay denied
spam@uce.uce.com                DISCARD
example.gr.jp                   RELAY
localhost                       RELAY
127.0.0.1                       RELAY
```

```
# makemap hash /etc/mail/access < /etc/mail/access
```

このコマンドでaccess.dbが生成され本設定が完了

Gateway: blacklist_recipient

- 自ドメインのあるアドレスが狙われた場合の措置手段
- /etc/mail/accessに設定を付加できるようになる

/etc/mail/access

```
bogus_user@                 REJECT
bogus.example.gr.jp         ERROR:550 Bogus host
junk@other.example.gr.jp    ERROR:550 Mailbox unavailable
```

```
# makemap hash /etc/mail/access < /etc/mail/access
```

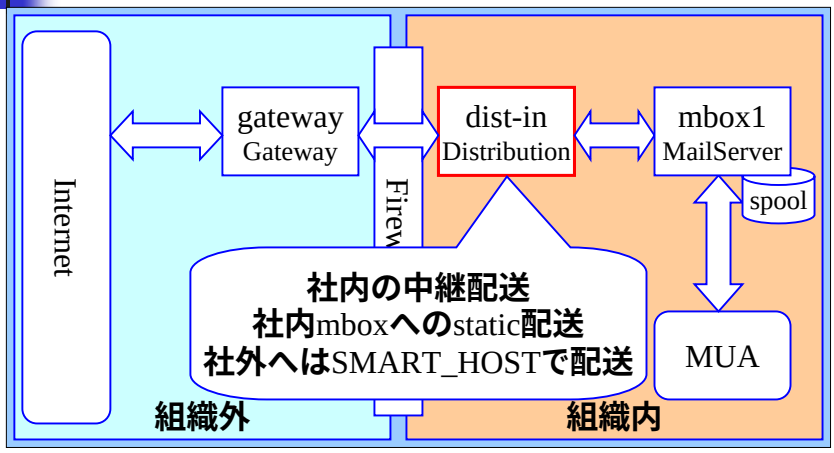
このコマンドでaccess.dbが生成され本設定が完了

Gateway: config.mcファイル

```
divert(0)dnl
VERSIONID(`$Id: config.mc,v 1.1 2001/12/4 13:48:05 ando Exp $')
OSTYPE(bsd4.4)dnl
DOMAIN(generic)dnl
FEATURE(`nocanonify')dnl
FEATURE(`mailertable')dnl
FEATURE(`access_db')dnl
INPUT_MAIL_FILTER(`myfilter', `S=local:/var/run/perl.sock')dnl
MAILER(local)dnl
MAILER(smtp)dnl
define(`confDOMAIN_NAME',`$w.$m')dnl
```

```
cd ${SENDMAIL_SRC}/cf/cf
make config.cf
make install-cf CF=config
```


配送モデル：社内中継サーバ



社内中継サーバ

- 特別に必要な機能
  - 社内メールサーバへのstatic配送
    - FEATURE(`mailertable‘)の利用
  - 自ドメイン以外へのメールをGatewayへ
    - クラス SMART_HOST にGatewayを設定

社内中継サーバ: mailertable

- 社内でのstatic配送ルールを書く
- 設定ファイル名は/etc/mail/mailertable

/etc/mail/mailertable

```
sub1.example.gr.jp      smtp:[mbox1.example.gr.jp]
sub2.example.gr.jp      smtp:[mbox2.example.ad.jp]
sub1.example.ad.jp      smtp:[192.168.10.25]
```

```
# makemap hash /etc/mail/mailertable < /etc/mail/mailertable
```

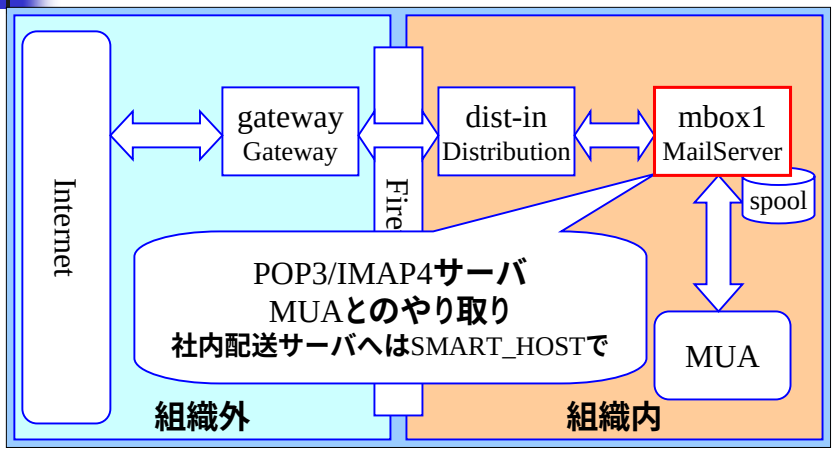
このコマンドでmailertable.dbが生成され本設定が完了

社内中継サーバ: config.mcファイル

```
divert(0)dnl
VERSIONID(`$Id: config.mc,v 1.1 2000/12/17 22:48:05 ando Exp $')
OSTYPE(linux)dnl
DOMAIN(generic)dnl
FEATURE(`mailertable')dnl
MAILER(local)dnl
MAILER(smtp)dnl
define(`SMART_HOST',`gateway.example.gr.jp')dnl
```

```
cd ${SENDMAIL_SRC}/cf/cf
make config.cf
make install-cf CF=config
```


配送モデル: 社内メールサーバ



社内メールサーバ

- 特別に必要な機能
  - 社内中継サーバへのstatic配送
    - クラス SMART_HOST の利用
  - 知らないドメインでもそのまま中継に渡す
    - 自分のドメインを付加しない
  - ドメイン名のマスカレード
    - マシン名だけ含まないアドレスでメールを出したい

社内メールサーバ: config.mcファイル

```
divert(0)dnl
VERSIONID(`$Id: config.mc,v 1.1 2000/12/17 22:48:05 ando Exp $')
OSTYPE(linux)dnl
DOMAIN(generic)dnl
FEATURE(`nocanonify')dnl
MASQUERADE_AS(`example.gr.jp')dnl
MASQUERADE_DOMAIN(`myhost.example.gr.jp')dnl
FEATURE(`limited_masquerade')dnl
FEATURE(`masquerade_envelope')
FEATURE(always_add_domain)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Dmexample.gr.jp
define(`SMART_HOST',`dist-in.example.gr.jp')dnl
```

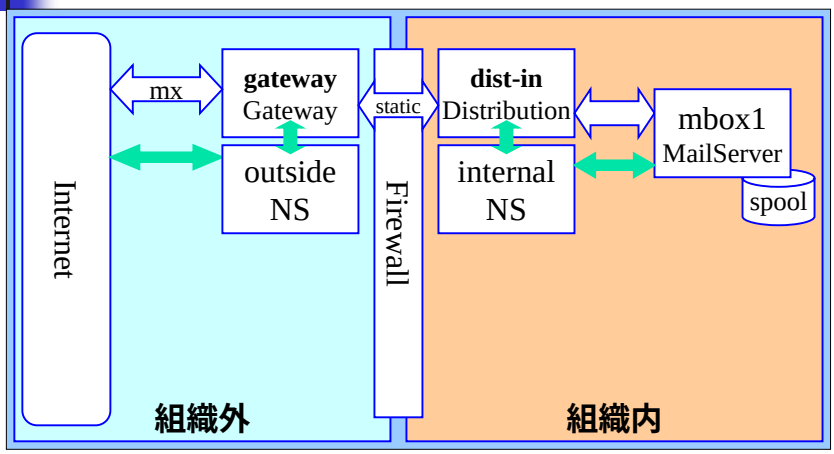
マスカレードするドメインの範囲を指定。

```
MASQUERADE_DOMAIN(`example.gr.jp')dnl
FEATURE(masquerade_entire_domain)dnl
```

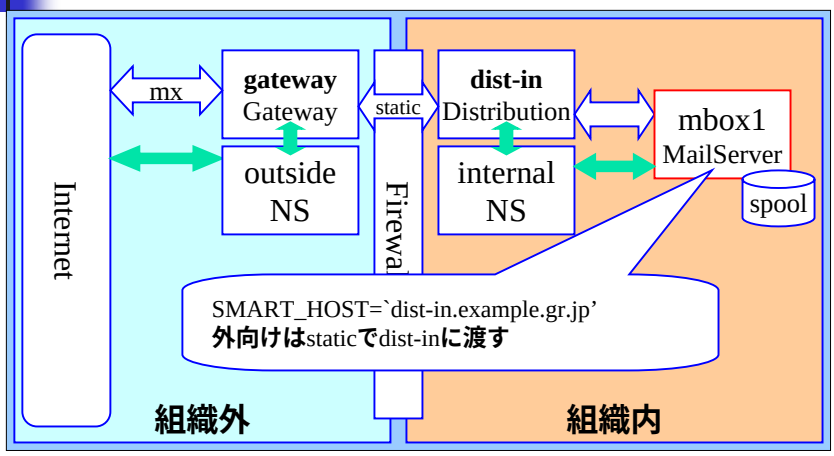
とするとそのドメイン以下全部のマシン名をマスカレード

```
cd ${SENDMAIL_SRC}/cf/cf
make config.cf
make install-cf CF=config
```

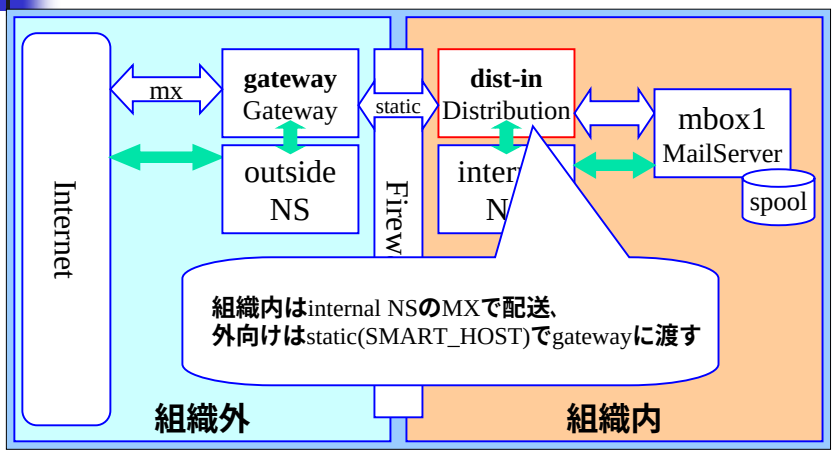
配送モデル（社内DNS利用）



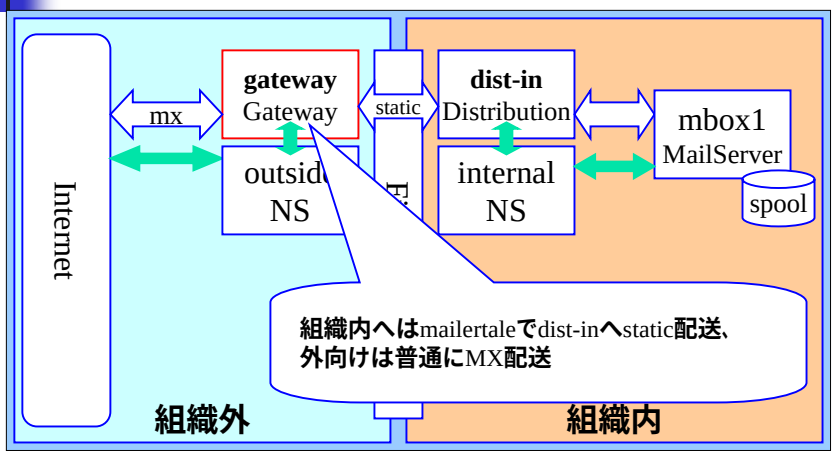
内部MSの設定



内側distributionサーバの設定



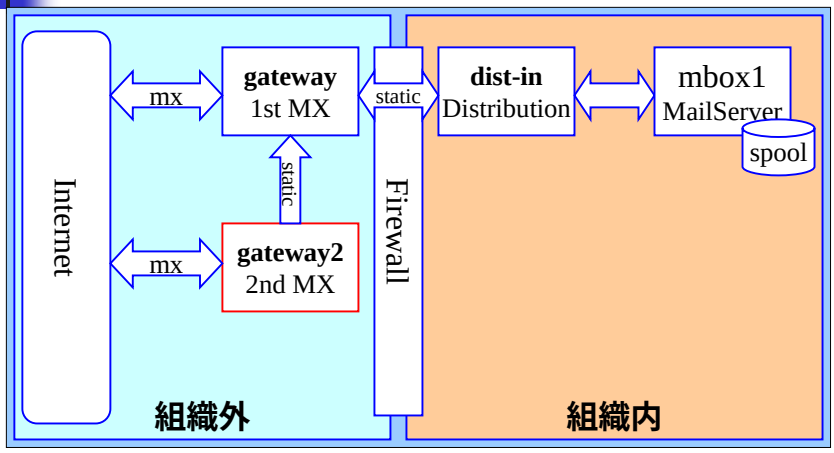
外側サーバの設定



対外受信ホストの多重化

- MXを複数にする理由
  - メールが集中して負荷が高い場合
    - 一時的にため込む
    - 可能なら2nd MXは1st MXとは独立に配信
  - メンテナンス用
    - 片方が停止しても受け取りに支障を出さない

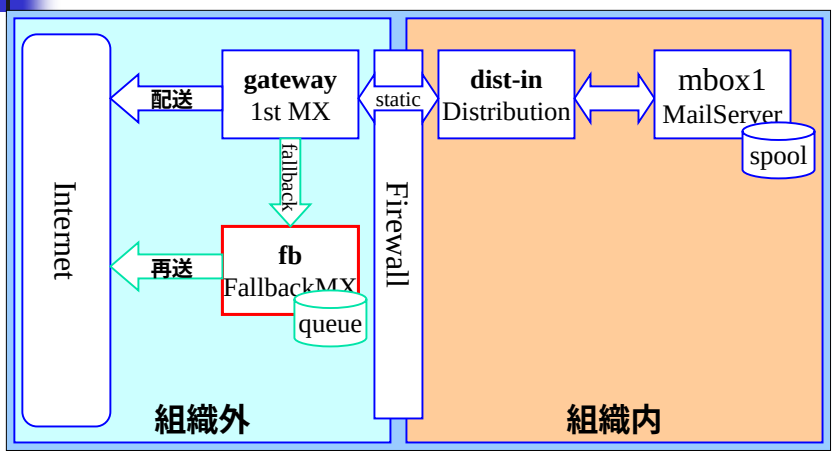
2nd MXのある配送モデル



FallbackMX

- 再送専用ホスト
  - 再送queueを特定のホストに集める
    - DNSが引けなかった場合
    - 全MXに対してメールが送れなかった場合
  - ネットワーク的なトラブルがすぐわかる
  - 再送を試みる期間の調整
  - `define(`confFALLBACK_MX',`fb.example.gr.jp')dnl`

FallbackMXのある配送モデル



設定の勘所 (1)

- まずはstatic配送＝mailertable
  - sendmailは配送先判定で真っ先にmailertableを見る
  - 全丸投げstatic＝LOCALRELAY
    - spool (localuser) がいない→LOCALRELAY
- ドラえもんstatic＝SMART_HOST
  - localとstaticで配送先がわからなかったら →SMART_HOST
- SmartHost設定がない＝MX配送
  - MXもAも引けない→エラー

設定の勘所 (2)

- spam対策はaccess_dbに集約
  - 身内のrelay配送の可否
  - spamlist的設定
  - blacklist-recipient設定
- DNSBLももちろん使える
  - デフォルトはMAPS RBL
  - もちろん他も指定可能

設定の勘所 (3)

- ドメイン名の書き換え
  - MASQUERADE_ASで書き換え後のドメイン指定
- 範囲指定
  - FEATURE(`limited_masquerade')だと
    - MASQUERADE_DOMAINに指定したドメインだけ書き換え
  - FEATURE(`masquerade_entire_domain')だと
    - MASQUERADE_DOMAINに指定したドメイン以下の全ドメインを書き換え

知っておくべきメールアドレス

- MAILBOX NAMES FOR COMMON SERVICES, ROLES AND FUNCTIONS (RFC2142) で挙げられているもの
- 例えば、
  - abuse@example.gr.jp
    - いざという場合の問い合わせ先
  - postmaster@example.gr.jp
    - メール配送についての問い合わせ先
  - hostmaster@example.gr.jp
    - DNSについての問い合わせ先

MLの周辺アドレス

- 周辺アドレスの例
  - owner-hoe@example.gr.jp
    - sendmail的にちょっと考慮されたMLの発信者アドレス
  - hoe-admin@example.gr.jp
    - 管理者のaliasとして使われることがある
  - hoe-request@example.gr.jp
    - RFC2142的管理者アドレス
  - hoe-errorsto@example.gr.jp
    - エラーメールの専用受信アドレスを用意している場合

SMTP/TLSの利用

- TLS (Transport Layer Security)
  - 乱暴に言うと、SSL接続への移行を視野に入れた接続の枠組みのこと
  - サーバ間SMTPを経路暗号化
  - sendmailでもこのTLSの枠組みを用いてSMTPの接続を暗号化することが可能
    - OpenSSLの利用が前提
    - 商用版では使えるようになっている製品もある

鍵の準備

- TLS (SSL) には認証用の鍵が必要
  - CA (認証局) から購入
    - 他社からのサーバ間接続でも認証付きで利用が可能に
  - 経路暗号化だけなら自前の鍵だけでOK
    - 鍵の配布範囲にTLSでの認証の利用が限定される
    - ユーザ認証はSMTP AUTHでやる

メール経由のウイルス (1)

- 添付ファイルが感染源であることが多い
  - マクロウイルス (Excel、Word、PowerPoint)
    - 中に忍ばせてあるOfficeオブジェクトが曲者
  - 実行形式ファイル
    - 不用意に実行してはいけない
  - JPEG画像
    - 実行ファイルを仕込むことが可能
    - HTMLメールの画像表示 (リンク) だけで危険

メール経由のウイルス (2)

- 自動的に実行されてしまう添付ファイル
  - .wav (nimda) とか.pif(Sircam)とか.scr(bugbear)とか
- 感染スピードの爆発的上昇
  - メール、HTTP、JavaScript、ファイル共有など複数経路で感染するワームの登場
  - 市販のウイルス対策プログラムのupdateが追いつかず、防ぎきれない例も多発
  - ウイルス除去プログラムが影響を除去し切れない例もある模様。
  - こまめにWindows updateを!

メール経由のウイルス (3)

- 添付ファイル
  - 元凶はMIME-multipart (便利さの代償?)
    - 入れ子構造でファイルを添付できる
      - 2段目にファイルを添付した後の1段目にウイルス添付 (nimda)
      - たまにデリミタの使い方を間違っているワームもある (Sircam)
      - 使われるContent-Typeも多様化している
  - 無限段まで入れ子をチェック
    - DoS対象になってしまうかも....

ウイルス･ワーム対策体制の例

- ウイルス対策プログラムを過信しない
  - ウイルスの感染の方が速い場合がある
- できるだけ速い情報の収集
  - ワームによるアクセスを監視（WWWサーバやIDSで）
  - 感染経路情報を示して警戒呼びかけ
    - なにもやらないのと比較して格段の防御になる
- 大量感染源になり得る部分での対策
  - メーリングリスト･ドライバで添付ファイルの拡張子チェック＋削除（メーリングリストでの添付ファイル使用の禁止）
  - Windowsのsecurity-updateに常に注意を払う

チェインメール

- 善意の協力依頼を装う（あるいは本物）
  - 「このメールを転載して下さい」が曲者
    - 無制限の転載を意図している場合には無視
    - 本来の目的を達成するには、期間や範囲を限定して一定数しか転載されない工夫を
- 不幸･幸福のメール
  - 「このメールを5人に転送しないと．．．」
    - 初心者の多い環境で流行りやすい

メール爆撃（Bombing）

- 2種類ある
  - 巨大なサイズのメールを送付
  - 膨大な数のメールを送付
    - どちらもspoolを膨らませる結果になる
    - loopと見分けがつきにくい場合がある
- サイズ制限、通数制限等の防御
  - メーリングリストではさらに深刻な問題に
  - O MaxMessageSize=500000

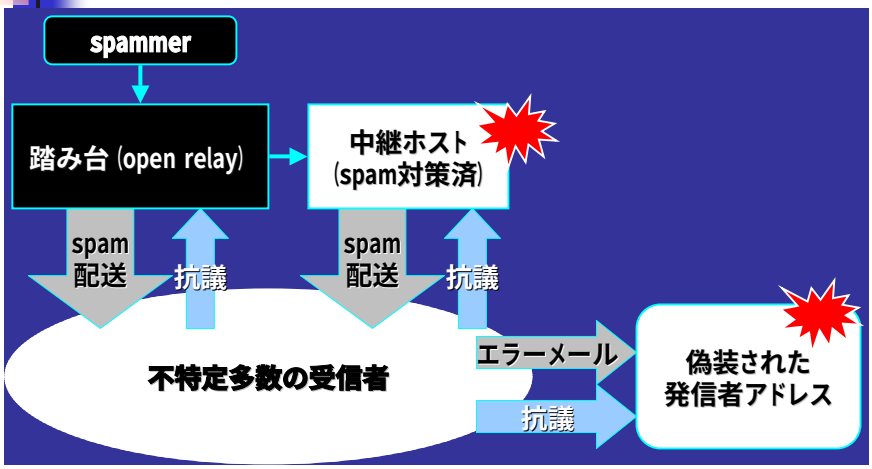
エラーメールの基礎

- エラーメール配信の枠組み
  - DSN（Delivery Status Notification）
    - Envelope From は null address（ <> ）
      - エラーメールに返信アドレスはない
- トラブルの種類を判定する手段
  - RFC1893（Status Code）：RFC2821に統合
    - Status: 5.1.1
      - 5.X.X　Permanent Failure
      - X.1.1　Bad destination mailbox address

エラーハンドリング問題

- 配送エラーコード (status code) の実装
  - 実際にRFCを守っているか？
    - sendmailやPostfix、SIMS等守っているものも多い
    - その他の対応はいまいち
      - MTAの数だけエラーハンドリングのプログラムが必要
      - 標準を守ろうとしないMTAは大迷惑なんだけど…
    - 大量にメールを配るところでは頭痛のタネ
    - 最近はウイルス通知メールの嵐
      - エラーメールの通知形式に準拠してくれないかなぁ….

spam中継の被害の構図



エラーメールによるRDDoS

- envelope-fromを詐称されてしまった場合
- 非常に多数のサイトから大量のエラーメール
- メインの1stMXが潰れそうになったら、1stMXをDNSから削除、TTLの短い2ndMXのみにする
  - RDDoSのエラーメールはDNSを新たに引いて2ndMXへ
  - 普段から良くメールの来る相手はDNSのcacheがあるので1stMXにメールが来る
  - DNSのcacheの生存時間を利用したエラーメールの振り分け
    - 岡山大学の山井先生の考えられた手法です（JANOG12）
      - RDDoS＝Reflected Distributed Denial of Service

大量のDoubleBounce

- エラーメールの配送エラー
  - 通常、エラーメールのエラーは消失する
    - エラーメールのsenderはnull-address
    - 例外がDoubleBounceの機能
    - DefaultではPostmaster宛になっている
  - envelope-fromの詐称による絨毯爆撃型spamの副作用として発生
  - DoubleBounceをOFFにする
    - ログをチェックすることが条件

必須の技術へ

- spam対策技術
  - 「来たときの対策」と「出させない対策」
    - SMTP Authentication (RFC2554)
    - Message Submission (RFC2476)
    - SMTP over TLS (RFC2487)
    - RBL/SBL
    - Bayesian filter
    - URL filter
  - メーリングリストではアドレス一覧を出さないこと
    - 例えばPPMLは一般参加者のwhoコマンドに対してGECOSの一覧を出す

最近の傾向 (1)

- 大規模化に伴う相対的な管理レベルの低下
  - ISP等では大規模化する一方
    - ユーザ管理の省力化を目的にディレクトリサーバを利用するケースも珍しくなくなってきている
  - 携帯電話メールのトラフィックの増加
    - 容量は小さいが通数はものすごい
  - MIME-multipartによる添付文書
    - 容量が大きいのでspool容量の再考が必要なケースも

最近の傾向 (2)

- spam送信側が高度に組織化されてきている
  - ゾンビPCを束ねて送信してくる
    - ワームやトロイの木馬を利用してゾンビPCを増やす
    - ゾンビPCは世界中に分散している
    - USに持っていかれたアドレスに世界中のゾンビPCからspam
      - CAN-SPAM法対策か? (US国内法の範疇外になる)
  - 日本語のspamもそのような送信法で送られてくるようになったらしい
    - 1つの国での対策はもはや限界

最近の傾向 (3)

- ようやくISPが動き始めたらしい
  - ユーザに対するspam対策手法の提供
    - ベイジアンフィルタの提供
    - 逆引きエントリの存在しないサーバからの受信拒否
      - ビジネスユーザを抱えていると危険ではあるが....
  - ISPでは「発信させない対策」をして欲しい
    - SMTP AUTHとTLSの併用で発信者認証を
      - パスワードを騙れば不正アクセスで検挙できそう?
      - 正式ユーザのspam発信は約款で禁止し明示的に罰則を
      - ダイアルアップのセグメントはport 25フィルタリング?

最近の傾向 (4)

- 常時接続の問題 (ゾンビPC対策)
  - ゾンビPCとそうでないPCの区別をどこで付けるのか?
  - SPFはドメイン内の送信ポリシーを記述できる
    - spammerはSPFのエントリーを書いてきている
  - 予防はウイルス対策と同等の扱いが必要
  - 本気でやるならport 25フィルタリングとISPの中継サーバとのサーバ間認証で送信制限するしかない
    - 住みづらい世の中になったものだ. . .
    - ISPにその余裕ある?

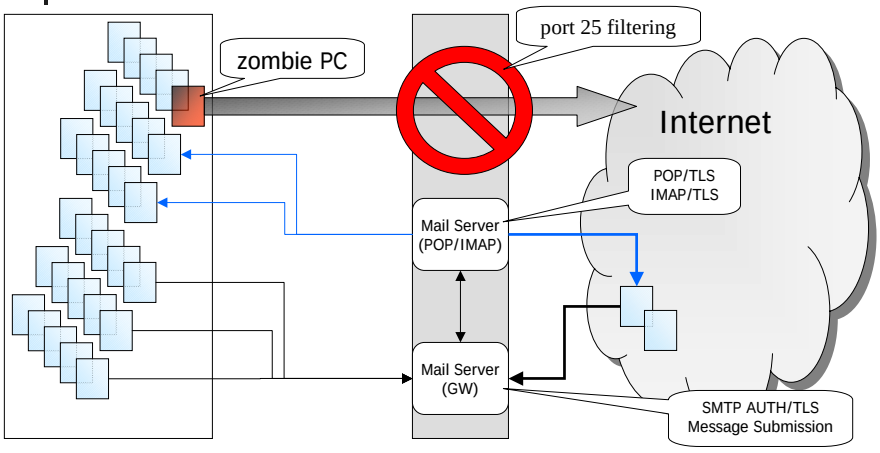
最近の傾向 (5)

- ISPはデフォルトでport 25をフィルタする?
  - 固定IPユーザには登録者だけport 25を通す
    - サーバ・マシン管理で自己責任を果たせることが必要
    - ISPのメールサーバを利用→問題なし
    - 自前メールサーバを利用→ゾンビPC対策が必須
  - それだけ厳しい状況になりつつある
    - 快適な環境を得るために我慢しなければならない部分
    - Phising等の犯罪の発生
    - spamの国際化
      - ゾンビPCは数十万台と言われている

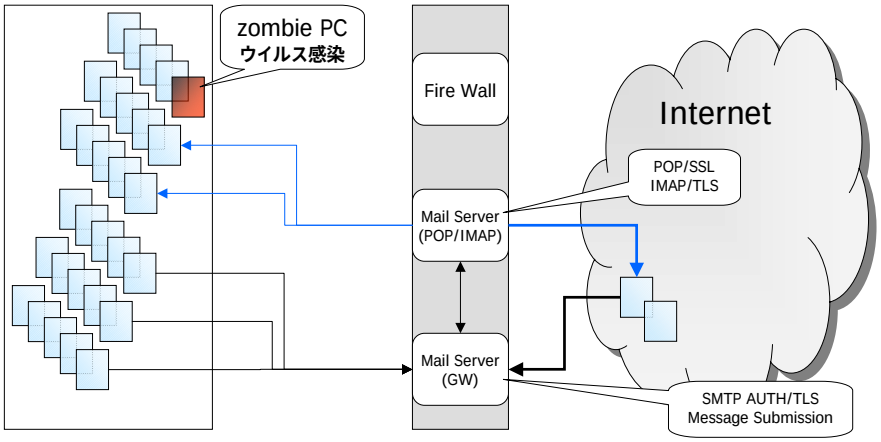
最近の傾向 (6)

- 家電製品にIP接続するものが出現
  - 一部製品はLinuxベース
  - 管理者権限でパスワードなしでアクセスできるものが!
    - ゾンビPCにするには持ってこいの素材
    - ベンダーの方は是非製品のセキュリティチェックを
    - telnetは開いてるわ、FTPもsambaも. . .
    - リモートからrebootできてしまう. . .
    - オンメモリ動作 (メモリ上にファイルシステム) しているものは電源OFFで一切の証拠が消滅. . .

ISPでの基本構成



企業での基本構成



アドレス詐称・隠蔽問題 (1)

- bombing等では発信者アドレスが偽装される
  - spam発信者を偽装して発信者をbombing
- MLに他人のアドレスを登録する
  - 自動登録でConfirmなしだとアウト
- 無料メールアドレスの転送機能
  - 誰に届くかわからないという意味で曲者

アドレス詐称・隠蔽問題 (2)

- Phishingが問題化
  - メールアドレスの詐称とWWWサイトの作りこみで個人情報を詐取する
  - 画像、バナーまで本物を使用
    - ページは本物でもID入力ウインドウが偽者の場合も
  - SSLでも証明書の中身まで確認しないとダメかも
  - メールでは詐称を防ぐ対策が必要に
    - アドレスは詐称できても発信サイトは隠しにくい
    - 発信者認証した後、その証拠をメールにどう残すか?

spam対策 (1)

- RBL (Realtime Blackhole List)
- SBL (Spam Blocking List)
  - spamの発信元を登録する闇魔帳
    - DNSと同じ枠組みで作られている
      - MTAがメール送信元のIPアドレスを照会
    - 残念ながら訴訟対策のためかどんどん有料化
      - ORDBでも寄付を募っている
    - 自分のサーバが登録された場合
      - メールを受け取らない所が出てくる

spam対策 (2)

- SPAMLIST (access_db)
  - 発信元についていずれかを指定して排除
    - メールアドレス (envelope from)
    - ドメイン
    - IPアドレス
- POP before SMTP
  - ISPで取り入れられている手法
    - POPアクセスの発信元に対してSMTP接続を許可する
    - 例えばqpopperにパッチを当てて実現する

spam対策 (3)

- Sender Base
  - spamを発信したことを記録している一種の信用 (reputation) サービス。
  - IPアドレス、IPブロックのオーナー、ドメイン、ドメインのオーナー等でグルーピングしている。
  - RBLの発展系とみることができる。

spam対策 (4)

- ベイズ推定を用いたフィルタ
  - 狙いはspamに登場する語句の出現傾向
    - 語句の出現傾向からspamかどうかを判定する
    - 辞書が比較的大きくなる
    - 言語依存 (現状で英語、日本語くらいならOK)
  - 弱い相手
    - 画像1枚、リンク1つだけのspam
    - 大量の一般的な文書に埋め込まれた広告
    - あの手この手の偽装手段

spam対策 (5)

- パターンマッチ
  - 例えば正規表現でパターンを指定
    - 個人で使ってもあまり効果はない
    - サーバで使用すると効果的
    - 誤判定リスクはパターン次第
    - 言語への依存性は実装次第

spam対策 (6)

- ヒューリスティック・フィルタ
  - 各部のパターンを抽出して確率で引っ掛ける
    - Fromヘッダの特徴
    - Subjectの特徴
    - Toの特徴
    - Receivedの特徴
    - Content-Typeの特徴
    - . . . と積み上げて判定する手法

spam対策 (7)

- URLをベースにしたspam排除
  - URLのパターンマッチ的な手法はよくある
    - 誤判定リスクは排除すべきURLの確認に依存
    - userinfoとquery部分を宛先ごとに改変している例
    - 言語依存性なし

spam対策 (8)

- デジタルシグネチャ (d-sig) のDB化
  - spamの各パートのd-sigを検知する
    - MIME multipart解析
    - d-sigが一致する (同一の内容の) partがあればspamと判定する
    - spamの内容も (ランダム文字列等で) その都度改変されるので、データの共有と更新が効果を上げる鍵になろう

spam対策 (9)

- SPF
  - AOLが採用している。
  - 自ドメインのメール発信ホスト/ポリシーをDNSに登録
  - 受信側はSMTP Senderから、登録された送信ホストからの発信かどうかをチェックする。
  - http://spf.pobox.com/

```
example.jp.  IN TXT "v=spf1 ip4:218.223.0.0/22 ip4:210.164.161.64/27
 mx a:accele.ope.example.jp a:sv04.example.jp a:jasmine.example.jp
 include:ico-g.com -all"
```

spam対策 (10)

- Microsoft Caller-ID
  - 自ドメインのメール発信ポリシーをXMLでDNSに登録
  - 受信側はメールヘッダの送信者から、登録された送信ホストからの発信かどうかをチェックする。
  - SPFと融合して発信サイト認証のRFCを作ろうという動きへ

spam対策 (11)

- Sender-ID
  - SPFとCaller-IDの融合規格として出てきたもの
  - IETF->IRTF->ASRG->MARIDでRFC化を目指した
  - Caller-IDの中にMSの未公開特許が含まれ、無償提供ながらライセンスに対する警戒感から頓挫 (送信者とみなされるべきヘッダの選択方法が特許になっている)
    - sid-filter (http://www.sendmail.net/)

spam対策 (12)

- Yahoo DomainKeys
  - 公開鍵暗号を利用した発信者サイト認証の仕組み
  - 公開鍵をDNSに掲載し、送信サーバでは正規に登録されたユーザからの送信メールに秘密鍵でサインして送信する。
  - ヘッダに記載された送信アドレスとDNSから得られる公開鍵を用いて、サインの正当性を検証する。
  - Yahoo,Google (Gmail) ,Sendmail等
    - dk-milter (SourceForge.net)

spam対策 (13)

- Channelled Address
  - 宛先に応じて自分のアドレスを変える
    - この宛先には自分のアドレスはこれで. . . と決めうち。
  - 返信先がその宛先用のアドレスかどうかでspam判定
  - USではAT&Tの特許があって使用許諾が必要。
    - WebMail形式のサービスとしてZoEmailというのがある。
    - 日本では講演者の特許検索の範囲では見つからず

spam対策 (14)

- 自動確認付きホワイトリスト
  - メールを出してきた相手に、「ほんとに送りたいならこのメールに返答してね」と返信し、そのメールに返答のあった送信者をホワイトリストに登録する。
  - MLの登録認証のしくみに似ている。

spam対策 (15)

- 流量制限
  - BruteForce型spamに対する対策
  - 同一送信元IPアドレスからのメールの受信数を制限
    - 動的に受信拒否動作をするものもある。
  - 同一送信元IPアドレスからのSMTP接続数を制限
    - Sendmailでも実装

spam対策の傾向 (1)

- アドレス偽装の問題化
  - Phishing (個人情報の詐取目的のメール) の横行
  - 発信サイト認証はPhishing対策の色合いが濃い
- ベイジアンフィルタはMUA側に実装
  - メールの振り分けをするため、POPサーバではアカウントが2つ必要になってしまう。IMAPならいいかも。
  - ISP側で一律フィルタすることは不可能
    - spamの定義が人それぞれで違うため
  - ナイーブなベイジアンフィルタはspammerの対抗策のためほぼ終焉。

spam対策の傾向 (2)

- spam発信側の技術の高度化
  - フィルタはアルゴリズムがわかると突破される
    - ベイジアンフィルタに対するWord-Salad等
  - ゾンビPCの存在
    - 持ち主の知らない間に発信サイトになっているPC
    - 常時接続ゆえの怖さ
  - WWWサイトに載せてあるアドレスにspamが来る
    - 実験済
    - 米国内のあるサイトに持っていかれたアドレスに世界中のゾンビPCからspamが届く

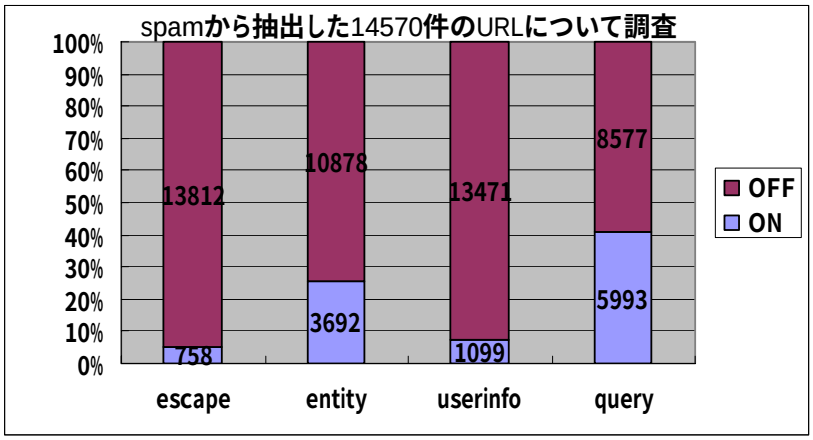
spam対策の傾向 (3)

- 受信対策から出させない対策へ
  - 発信者認証の積極的な採用を
  - 認証結果をメールに記録
    - 認証アドレスをSenderヘッダに記録
      - Senderヘッダは配送に影響しない→控えめな対応
    - 認証アドレスをSMTP senderにして発信
      - 完全に普及するとエラーメールRDDoSへの対策になる
  - 認証を通らないメール送信の遮断
    - port 25 filtering
    - IPアドレスブロックの管理責任

spam対策の傾向 (4)

- メール中のURLの詐称
  - 昨年の講演の時点でURLの隠蔽状況をまとめていたが、Phisingの横行で心配は現実のものとなった。
  - MUAの持つ脆弱性にはこまめなアップデートで対応
  - セキュリティ情報に常に関心を持つこと
    - 重大なものは社内にアナウンスすることも必要

URLの改変可能要素



spam対策の傾向 (5)

- JPEG画像にワームを仕込む
  - 今年問題化した新種
    - HTMLメールでリンクが張られているJPEG画像を読み込んだだけで感染
    - 根本はWindowsのライブラリの問題なので、そこを対策しないと、ブラウザもMUAも危ない。
    - 単純なspamかどうかの判定も難しくなってきた
    - ウイルス対策→ゾンビPC対策→spam対策という構図が発生

Spam対策の傾向 (6)

- シマンテック社の動き
  - 米国Security Focusを買収
  - 米国BrightMailを買収
- マカフィー社 (ネットワークアソシエイツ) の動き
  - DeerSoft (SpamAssassinの母体) を買収
    - Apache SpamAssassin ProjectでOpensource版も継続
- 被害をもたらしている主なウイルスもspamとして配信されてくる
  - ウイルスフィルタで排除すべき対象
  - ウイルス対策ベンダーがspam対策に動き出している

まとめ (1)

- メールサーバの管理は難しくなってきている
- spamを発信させないのは社会的要請
  - 不用意なメール中継をしないのは当然
  - ユーザの利便性を確保するため発信者認証を利用
  - 認証パスワードの漏洩を防ぐためにTLSを利用
    - Googleで引いてしっかりした設定方法のページがあったのでここでは設定のしかたは割愛してます
  - 無知なユーザをも守る施策を!

まとめ (2)

- 受信したspamへの対策はMUAの機能が鍵
  - 例えばMozilla Thunderbirdの場合
    - ベイジアンフィルタ装備
    - アドレス帳にあるかどうかでフィルタ設定が可能
      - ホワイトリストとして利用できる
    - spamフィルタにはヘッダ情報を付加するものが多い
      - 任意のヘッダ情報を見るようにフィルタ設定が可能
  - ユーザの教育が欠かせない

まとめ (3)

- POP/IMAPサーバについても
  - 企業の場合、出先 (モバイル環境) からの使用を考えなければならない
  - POP/SSL、IMAP/TLSといった技術を利用すること
    - 平文パスワードの飛ぶ状態での使用は絶対に避けること
    - APOPなら良いか?
      - APOPはCHAPと似ていて、サーバから渡される文字列とshared passwordの文字列を合わせて、サーバ側とクライアント側でMD5ハッシュ値を計算し、クライアントの計算結果をサーバ側で検証することで正当性を認証する仕組み. ユーザ名はばれる。

その他注意すべき話題

- MD5 (128bit) が破られると...
  - 以下のような認証が破綻
    - APOP
    - SMTP AUTH/SASL (CRAM-MD5、Digest-MD5)
    - PAP/CHAP (ダイアルアップの認証)
    - 被害甚大だが既にハッシュ値の衝突が起きるデータの生成方法が論文発表されたりしている模様
- SHA1 (160bit) が破られると...
  - SET (電子決済) が破綻
    - 考えたくない事態に...

付録 (devtools/Site/siteconfig.m4)

```
APPENDDEF(`conf_sendmail_ENVDEF', `-DMILTER')
APPENDDEF(`conf_sendmail_ENVDEF', `-DSASL')
APPENDDEF(`conf_sendmail_LIBS', `-lsasl')
APPENDDEF(`confINCDIRS', `-I/usr/local/include/sasl1')
APPENDDEF(`confLIBDIRS', `-L/usr/local/lib')
APPENDDEF(`conf_sendmail_ENVDEF', `-DSTARTTLS')
APPENDDEF(`conf_sendmail_LIBS', `-lssl -lcrypto')
```

付録（社内ホスト設定例）

```
VERSIONID(`$Id: config.mc,v 1.5 2004/12/04 12:27:36 ando Exp ando $')
OSTYPE(bsd4.4)dnl
DOMAIN(generic)dnl
MASQUERADE_AS(`example.jp')dnl
MASQUERADE_DOMAIN(`accele.example.jp')dnl
FEATURE(`limited_masquerade')dnl
FEATURE(`masquerade_envelope')dnl
EXPOSED_USER(`root postmaster')dnl
FEATURE(`mailertable')dnl
FEATURE(`nocanonify')dnl
FEATURE(`access_db')dnl
FEATURE(`blacklist_recipients')dnl
FEATURE(`accept_unresolvable_domains')dnl
FEATURE(`no_default_msa')dnl
MODIFY_MAILER_FLAGS(`LOCAL', `+S')
MAILER(local)dnl
MAILER(smtp)dnl
Dmexample.jp
Dwaccele
define(`confDOMAIN_NAME', `$w.$m')dnl
define(`confTO_IDENT', `0s')dnl
define(`confCF_VERSION', `IW2004 Sample')dnl
define(`confMAX_QUEUE_CHILDREN', `100')dnl
define(`confMIN_QUEUE_AGE', `1m')dnl
define(`confAUTH_MECHANISM', `[GSSAPI KERBEROS_V4 DIGEST-MD5 CRAM-MD5]')dnl
TRUST_AUTH_MECH(`CRAM-MD5 DIGEST-MD5')
dnl INPUT_MAIL_FILTER(`sid-filter', `S=inet:8891@localhost')
INPUT_MAIL_FILTER(`dk-filter', `S=inet:8891@localhost')
define(`confCACERT_PATH', `/etc/ssl/CA/certs/')
define(`confCACERT', `/etc/ssl/CA/ca.crt')
define(`confSERVER_CERT', `/etc/ssl/CA/certs/server-ca.crt')
define(`confSERVER_KEY', `/etc/ssl/CA/private/server.key')
```