spam 検知情報の XML による共有

2007MI171 丹羽　清志　　2008MI016 藤田　公孟　　2008MI277 山内　裕太
指導教員　後藤　邦夫

1　はじめに

ネットワーク社会と呼ばれ, インターネットを利用する人が増えている．あらゆる手段で個人情報を取得し, 得たメールアドレスに向けて高い頻度で大量に営利目的の迷惑 (spam) メールを配信する手口で, 犯罪の手段の 1 つとして spam メールが利用され, 社会問題となっている．メールの受信者からすると, spam メールを受信してしまうこと自体がそもそも迷惑である．また, 受信をするメールサーバからすると, spam メールを大量に送られることにより, サーバの処理能力が低下する被害がある．また, spamhaus.org や Surbl, Server Authority といったブラックリストの共有をし, 対策をする例もあるが, spam メールだけではなく, 必要なメールもブロックしてしまうという問題がある．単一の情報ではなく, spam メールのヘッダから複数の情報をもとに判断することで問題が解消される．また, 大学や, 会社といった組織間で共有することで spam メールの対策をすることができると考えた.

そこで本研究では, spam 検知情報の XML による共有を提案する．組織ごとに定義されている記述方法や管理方法を統一することで, 膨大な spam メールのヘッダ情報を整理することに役立ち, 扱うデータの意味を判断することができるようになるので, XML を用いる．さらに, 情報の共有により, 受信制限の設定を細かくし, 現状より効率よく spam メールを防ぐことができるという利点がある.

なお, 丹羽はシステムの構築・考案を, 藤田は実験を, 山内は環境構成を担当する．なお, プログラムの作成は 3 人で協力し作成する.

2　システムの概要

この節では, 本研究で行う共有方法の概要と共有する情報, 管理方法について説明する.
共有方法には, データベースサーバを集中して管理する方法と, 分散して管理する方法の 2 通りがある．それぞれの利点と欠点は表 1 である [4].

2.1　共有方法

表 1　各管理方法の利点と欠点

方法	利点	欠点
集中	通信回数が少ない	サーバの負担の集中
分散	負担が分散される	通信回数が多い

本研究では, 大規模の組織数での共有を想定している．サーバ間での通信回数が非常に多くなると考え, 集中してデータベースサーバを管理する方法を採用する．なお, 中央データベースサーバへ spam 情報を送信する際には, SSL 公開鍵認証を用いて, spam 検知情報を暗号化し, 送信する．さらに, データベースサーバに記述ができるのは, 各組織の報告者だけとする．そして, 入力フォームへ入るまでには, パスワードを設定し, セキュリティの向上を計る．また, 組織内のその他は参照のみ可能というグループ分けをする．入力フォームに記述された情報は, CGI プログラムを通して XML 文書に変換され, 組織内のデータベースサーバに格納される．同時に XML 文書は TCP(Transmission Control Protocol) を使い中央のデータベースサーバに送信される．中央データベースサーバは, 1 日 1 回各組織のデータベースサーバに TCP を使い最新の spam 検知情報のデータリストを送信する．この全体図が図 1 である.

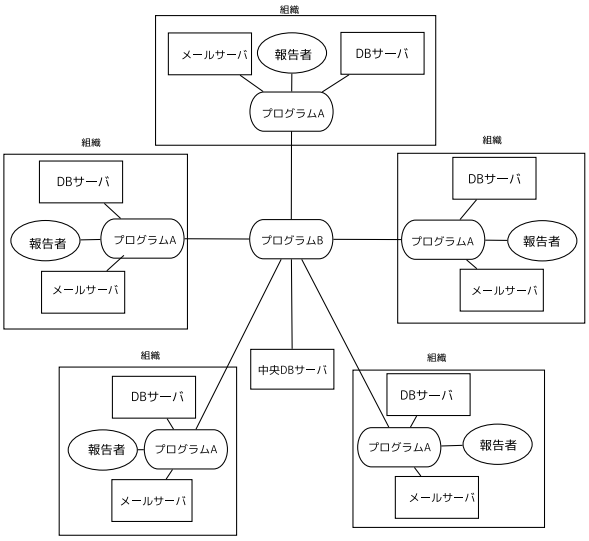

図 1　XML の共有方法

共有までの流れは以下のようになる.

1. 組織ごとにデータベースサーバを導入する.
2. 報告者は各組織で管理している spam メールのヘッダ情報と本文を CGI で作られた入力フォームに書き込むことで報告する.
3. 報告された spam 情報はプログラムを介して, 自動的に XML に整理され, データベースサーバへ記述する.
4. データベースサーバは, XML に整理された spam 情報をテーブルに格納する.

5. プログラムは, TCP(Transmission Control Protocol) を介して, 格納された情報を中央データベースサーバに送信する.
6. 組織ごとのデータベースサーバでは, 中央データベースサーバから決まった頻度で情報を受け取る.

2.2 プログラムの処理内容と役割

プログラム A とプログラム B の説明は次の通りになる.
プログラム A

- 入力フォームを表示する. また, 入力された情報を XML に変換して, 組織内のデータベースサーバとプログラム B に渡すプログラム.

プログラム B

- プログラム A から受け取った情報を中央データベースサーバに返すプログラム. また, 一定周期で, 中央データベースサーバから受け取った情報を各組織のデータベースサーバに送るプログラム.

2.3 spam メールから得る情報

情報を共有するにあたって, spam メールのヘッダから読み取る情報は以下のものとする.

- IPadd – 送信者の IP アドレス
- Received – 送信されるさいのサーバーの経路情報
- From – 送信元のメールアドレス
- Date – 送信された日時, 時間
- Timezone – タイムゾーン
- Subject – 件名
- Body – 本文

さらに, 本研究では spam メールそれぞれに管理番号をつけ, 組織 No ＋番号という形でデータベースサーバに保存する. 番号は入力フォームから送信した時の西暦 4 桁月日時分秒の 14 桁からなる管理番号をつける. 管理番号をつけることによって, spam 情報の重複を防ぎ, どの spam 情報かを特定することが可能となる. こういった理由から, 一度検索した spam 情報を瞬時に検索することができる. 以下に例を示す.

- 組織 A が 2011 年 4 月 25 日 16 時 20 分 11 秒に送信した spam 情報の管理番号 – A.20110425162011
- 組織 B が 2009 年 2 月 2 日 6 時 0 分 1 秒に送信した spam 情報の管理番号 – B.20090202060001
- 組織 C が 2002 年 10 月 5 日 6 時 20 分 51 秒に送信した spam 情報の管理番号 – C.20021005062051

2.4 テーブルの定義

テーブルを作るうえでのテーブルの定義は以下のものとする.

- Man.No – char(15)
- spamdata – DB2XML.XMLVARCHAR

Man.No は主キーで, どの spam データを特定するために用いる. また spamdata には XML 形式のファイル内容すべてを格納する.

2.5 XML でのタグセット管理

XML で管理するタグセットについて述べる. 方法案として以下の 3 通りある [1].

方法 1 タグセットを集中管理し, 各組織にタグの追加を認めない.
利点)

- タグを利用した情報収集, 交換が容易である.
- 組織内であればタグセットは集中管理しやすい.

欠点)

- 各組織が共通のタグで記述する必要があるため, 組織間で情報を共有するには不向きである.

方法 2 各組織がまったく自由にタグを定義できる.
利点)

- 各組織がそれぞれのタグで記述できるので, 組織間で情報を共有するには向いている.

欠点)

- 他の組織から情報を取得するためだけでも, タグの変換が要求されるので, 事前に共有する情報を決めておかなければならないため, 組織独自の視点で集めた情報を交換しにくい.

方法 3 タグセットの管理を階層化し一部を集中管理する.
利点)

- 集中管理されるタグを利用した情報収集ができ, 独自情報の記述も可能である.
- 方法 1, 2 の利点を上手く併せ持ったもの.

欠点は特にない.
よって本研究では, 方法 3 を用いて実験を行うことにした.

2.6 システムの運用方法

情報共有にかかわる組織が協力して上位のタグセットを作成する. 次に, 各組織がそのタグセットの下に組織独自のタグを追加し, 追加したタグの一覧を他の組織に報告する.

3 システムの実現

この節ではデータベースの処理と実現したシステムの具体例を述べる.

3.1 データベースサーバ

使用するデータベースサーバの案として以下の 3 つがある.

- DB2 Express-C 9.7[2]
- eXist
- Oracle Berkeley DB XML

本研究では, データベースサーバとして, DB2 Express-C 9.7 を利用する. 利用する理由として, 以下の 3 点がある.

- 文書格納方法の定義である DAD ファイルを利用することで, 容易に XML 形式の情報をテーブルに格納する機能がある.
- sql, Xquery ともに使用することができる.
- C, C++, Java, PHP, COBOL などのプログラミング言語が使用可能である.

3.2 データベースサーバでの処理

本研究におけるデータベースサーバでの処理は図 2 になる.

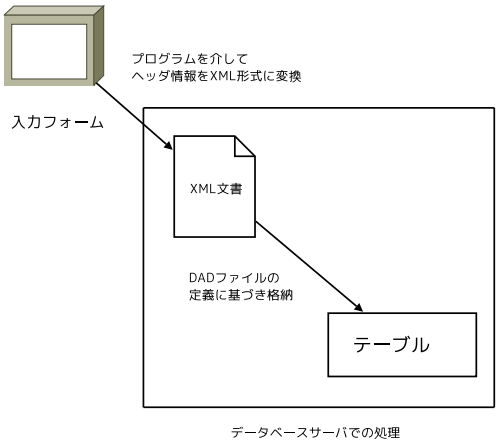

図 2　データベースサーバでの処理

プログラムを介して XML 形式に変換されたヘッダ情報を Document Access Definition(以下、DAD) ファイルによる関連付けをする. DAD ファイルで指定した path の情報を抽出し, テーブルへ格納する.
本研究では, XML Column を利用する. XML Column は, テーブルに格納する XML 文書について, あらかじめ検索条件として用いる要素を決め, その個所のデータに基づくインデックスをサイドテーブルという形で構築する. また, サイドテーブルに含まれない要素でも, 検索は可能である. XML Column を利用する理由は以下の 2 つである.

- 頻繁に検索する条件が決まっている.
- 元の XML 文書も残したい.

3.3 メールヘッダ情報の整理

spam メールのヘッダと本文の例を例 1 に示す.

例 2 ヘッダと本文の例

```
Received: from 210.165.10.13 (HELO mail.
goo.ne.jp) (210.165.10.13)by mta555.mail.
kks.yahoo.co.jpwith SMTP; Fri, 21 Oct
2011 02:21:08 +0900
Received: (qmail 29094 invoked from
network); 21 Oct 2011 02:21:08 +0900
Received: from unknown (HELO smtp01.mail.
goo.ne.jp) (172.27.1.20)by localhost.
mail.goo.ne.jp with
SMTP; Fri, 21 Oct 2011 02:21:08 +0900
Date: Fri, 21 Oct 2011 02:21:06 +0900
From: test_spam_1@mail.goo.ne.jp
To: test_spam_2@yahoo.co.jp
Subject: 当選しました
Body: これは迷惑メールです
```

このメールヘッダからは送信元からのサーバ経路情報, IP アドレス, 送信日時, 送信元のアドレス, 送信先のアドレス, 件名が記載されていることがわかる. このヘッダ情報をスキーマの定義に基づいて, XML で整理した記述例を例 3 に示す. なお, 本文については, 全文を記載するのではなく, 一部抜粋したキーワードのみを記載する.

例 2 XML 変換処理をした例

```
Man.no: D.111212142517
<?xml version="1.0"?>
<spam>
<IPadd>172.27.1.20</IPadd>
<Received1>from 210.165.10.13 (HELO mail.
goo.ne.jp) (210.165.10.13)</Received1>
<Received2>from unknown (HELO smtp01.mail.
goo.ne.jp)(172.27.1.20)</Received2>
<From>test_spam_1@mail.goo.ne.jp</From>
<Date>Fri, 21 Oct 2011 02:21:06 +0900
</Date>
<Timezone>+0900</Timezone>
<Subject>当選しました</Subject>
<Body>これは迷惑メールです</Body>
</spam>
```

本研究では, 例 2 を入力フォームへ書き込むと, プログラムを介して例 3 のようになる. 例 3 の 1 行目, 2 行目以下をそれぞれ, テーブルの Man.no, spamdata に格納する. 格納した情報を DAD ファイルの定義に従って整理し, デフォルトビューを参照すると表 2 のようになる. 情報を指定することで, デフォルトビューに表示されているタグ以外の情報も表示することができる.
DAD ファイルの定義は, 以下の定義に基づいて記述

した.

- validation は検証をするかどうかを yes もしくは no で指定する.
- Xcolumn は, XML 文書を格納する場合, 保存方法の詳細を Xcolumn タグの中で定義する.
- table は, サイドテーブルの定義を記述する.
- column は, 検索対象として抽出したデータを格納するためのカラムを定義する.
- name 属性と type 属性は, それぞれカラムの名前と型を指定, どこから抽出するか指定するための単純ロケーションパスは, path 属性に記述する.

DAD ファイルの記述例は例 4 のようになる.

例 4 DAD ファイルの記述例

```
<?xml version="1.0" ?>
<dad>
<validation>no</validation>
<Xcolumn>
 <table name="side_IPadd" >
  <column name="IPadd"
   type="char(15)" path="/spam/IPadd"
   multi_occurrence="no" />
 </table>
 <table name="side_From" >
  <column name="From"
   type="varchar(64)" path="/spam/From"
   multi_occurrence="no" />
 </table>
</Xcolumn>
</dad>
```

表 2　テーブルに格納した例

Man.No	D.111212142517
side_IPadd	172.27.1.20
side_From	test_spam_1@mail.goo.ne.jp

4　実験と結果

実験環境は Ubuntu10.0.4 をインストールした PC を複数台用意し, それぞれを組織内のデータベースサーバと想定する.

4.1　実験の手順

以下の方法で実験をする.

1. 実際に受け取ったメールを spam メールと想定する.
2. ヘッダの情報を読み取り, その内容を入力フォームへ書き込む.
3. プログラムを通して XML に変換後, 組織内のデータベースサーバと中央データベースサーバに送る.
4. 組織内のデータベースサーバと中央データベースサーバは, テーブルに格納する.
5. 中央データベースサーバの情報を定まった更新頻度で他組織のデータベースサーバに送信する.

手順 1 から手順 5 の操作を繰り返す. 本研究では, 組織内のデータベースサーバと, 中央データベースサーバ, 更新後の他組織のデータベースサーバの 3 箇所で, 任意の情報を索引することにより共有の確認とする.

4.2　実験の結果

本研究の実験は, spam 情報を入力フォームへ書き込んだ. そして, 書き込まれた内容は, XML 形式で表示された. データベースへ挿入するコマンドを入力すると, 自動でテーブルに格納され, DAD ファイルの定義に基づき, デフォルトビューが作成された. また, 中央データベースサーバには, Man.no と XML 形式に変換された文書が届いた. 挿入することで中央データベースサーバでも同じ内容が参照することができた.

5　おわりに

本研究では, XML 形式への自動変換, 組織内への情報送信, 中央データベースサーバへの情報送信を行った. 本研究により, 複数の組織間で共有し, より多くの spam 情報を取得することで, spam メールを正確に受信制限できるようになった. また同研究室の青山の卒業研究, 電子メイルヘッダの調査による spam メイル判定の提案 [3] と組み合わせることで, より正確な spam メールの情報共有ができる. 今後の課題として次の点がある.

- 受け取った情報をデータベースに加えるプログラムの完成.
- 一定周期で, 周期間で受け取ったデータを受け渡すプログラムの完成.
- 送信する情報を選択できるプログラムの完成.

参考文献

[1]　服部哲, 田畑邦晃：Web サービスによる分散 XML データの共有方式, 神奈川工科大学研究報告 (2007).

[2] IBM: DB2 Express-C, http://www-06.ibm.com/software/jp/data/db2/v9/express-c/ (2011).

[3] 青山尚樹：電子メイルヘッダの調査による spam メイル判定の提案 (2012).

[4]　村井純：集中システムと自律分散システムの比較, http://www.soi.wide.ad.jp/class/20010002/slides/02/8.html (2000).