平成20年度卒業論文

論文題目

検索エンジンによるCovert Channelの検出

神奈川大学 工学部 電子情報フロンティア学科
学籍番号　200602824
久保　直也

指導担当者　木下宏揚　教授

目 次

図 目 次

第1章　序論

社会では機能的に分化し競争しているシステムによりセキュリティポリシー [8][9] が複雑になり，またインターネットの普及により情報の流出や改ざんが大きな問題となっている．近年, 社会ではネットワークがインターネットをはじめ, 急速に発展し, 巨大化・複雑化している．それによりアクセス権限も複雑に絡み合うようになっていて, その結果ネットワーク内では不正な情報経路が発生し, 情報流出の危険性が増大してしまっている．このような情報流出経路の解析法として Covert Channel[8][9] 解析がある．

しかしこの Covert Channel 解析には従来のように把握したコミュニティの ACL[8][9](Access Control List) のみを用いた Covert Channel の解析だけでは検出できないアクセス権の矛盾が存在する場合があるといった問題点がある．

いままでの我が研究室の Covert Channel 分析では把握したコミュニティの ACL のみを用いた Covert Channel の解析を行ってきた．しかしそれでは検出できないアクセス権の矛盾が存在する場合があるという点に注目し, 本研究では検索エンジン [2] によって得られた情報にオントロジー [1][10] を用いたセマンティックな解析手法を適用することで外的要因を考慮した場合の ACL の矛盾や経路を効率よく見つけることを目的とする

本稿では, 検出できない可能性のあるアクセス権の矛盾を検出するために, オントロジー DB を作成し検索エンジンで収集した情報を形態素解析・構文解析 [5] し RDF[3][6] 化し意味まで考慮したマッチングを行うことで外的要因まで考慮した ACL の矛盾や経路を見つける方法を提案する．

本論文の流れを以下に示す．

第 2 章では,Covert Channel と検索エンジン, オントロジー,RDF の知識について述べる.

第 3 章では, 検索エンジンを用いた Covert Channel の検出方法について述べる.

第 4 章では本研究のまとめ、今後の課題について述べる.

第2章　基礎知識

2.1　Covert Channel

2.1.1　間接情報フロー

Covert Channel [7][8][9] はアクセス行列において, Subject,Object, permission をアクセストリプルと定義したとき, その 3 点で起きる不正な情報経路である. この場合,Covert Channel はアクセス禁止のパミッションに矛盾する情報フロー（アクセス禁止のものもこのフローを使えばアクセス禁止の内容を閲覧したり, 書き換えることが出来てしまう）ともいう.（ 図2.1 参照, 各S=Subject, 各O=Object,R=読みこみ可能権限,W=書き込み可能権限）図 2.1 の場合, 矢印の流れで Subject1 が本来読めないはずの Object1 を読めてしまう. Covert Channel 流出の流れは以下のようになっている. これは間接情報フローとも呼ばれる.

・始点（Subject2・Object1）Subject2 が Object1 を読み込む

・中間点１（Subject2・Object2）Subject2 が Object2 に Object1 で読んだ内容を書き込む

・中間点２（Subject1・Object2）Subject1 が Object2 を読む.

・終点（Object1・Subject1）Covert Channel により間接的に Object1 の内容を読めてしまう

このように不正な情報流出が発生してしまうため, アクセス制御を行う推論エンジンとしては出来るだけ発生を抑制し, 検出と訂正を的確に行えるようにするのが情報フィルタに必要な機能である.

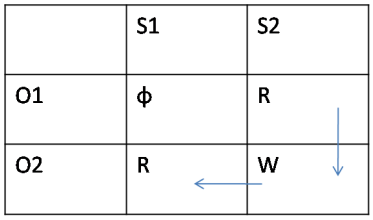

図 2.1: Covert Channel(間接情報フロー)

2.1.2 実際に発生する Covert Channel

不正な情報経路である Covert Channel を全て塞いでしまえば安全なシステムを構築することが出来るように見えるが, 単独では隠れチャンネル (Covert Channel) が存在しないようなコンピュータでもネットワークに接続されたコンピュータ群が協調することによって, 隠れチャンネルを構成できてしまう。つまり, 単独では安全なコンピュータでも, それがネットワークを構成すると安全ではなくなるような状況が簡単に存在し得るのである. このようなネットワーク構成機能の問題点が Covert Channel で利用される. 例えば以下のような例が挙げられる.

1 会社の機密データを社外へ持ち出したり、社外の人間（社外の PC）でも見れるようにする.mixi 等の SNS の個人データが掲示板やブログ等不特定多数へ流出

2 スパイウェア等, 個人 PC から情報を持ち出すためにこれを用いて通信を行い, 検知を困難とする. WWW 等、不特定大多数が利用するネットワークでは意図しなくてもカバートチャンネルが発生してしまう恐れがあるのでそういった情報網では比較的安易に情報漏洩が起こりうる. このように Covert Channel は今のネットワーク社会にとって情報を安易に流出させてしまう存在なのである.

2.2 フローレベル

Covert Channel において, ある Subject から Subject へと情報が流出する回数をフローレベルと定義する. 関わる Subject の数によっ

て,Covert Channel の程度は決まり, Subject 数を k としてフローレベル k と表記できる．フローレベル k はフローレベル 2 が基となっている如何なるフローレベル k においてもフローレベル 2 をベースとして生成される性質である．この性質からフローレベル 2 を抑えれば, そこから発生しうる多くのフローレベル k を防ぐことが出来, Covert Channel 分析と評価ができる．

2.3 アクセストリプル

Subject,Object,permission をアクセストリプルと呼んでいるがそれぞれの意味についてまとめる．Subject は主体の意味で Covert Channel では主に人間（Object に対する権限がそれぞれ異なるユーザ）を指す．Object は客体の意味で Covert Channel では主にデータ（文字や画像等のことで見れるユーザ, 書きかえられるユーザがそれぞれ異なる）を指す． この Subject と Object には以上のように二項関係が成り 0 立っていて互いの性質や関係を定義してやる必要がある．permission はファイル保護モードを言い,「READ」「WRITE」「EXE」の 3 種類の permission が,「ファイルの所有者」「グループユーザー」「その他のユーザー」のそれぞれに対して設定される権限をいう．本実験では permission は原則として「read」（読み込み可）・「write」（書き込み可）・「¬ read」（読み込み不可）・「¬ write」（書き込み不可）の 4 種類を表す．このアクセストリプルをプログラム上でまとめることで Covert Channel の検出・訂正を行う．

2.4 セキュリティモデル

セキュリティモデルは, アクセス制御システムを構築する上で, セキュリティポリシーを具体的な論理的形式で表現したものである. そこには制御したいサービスや組織構造が反映される. 最も単純な型では,permission(read,write, ¬ read, ¬ write) であり,Subject（主体）,Object（客体）を含めた３つでアクセストリプルと呼び, それをシステムで如何扱うかによってアクセス制御が行われる. 従来のセキュリティモデルには様々な種類があり, 使用される状況に応じて使い分けたり, 複数使用する等している.

2.5 情報フィルタ

情報フィルタとはCovert Channel 検出時にそのCovert Channel が無くなるように特定の権限を変更することである。情報フィルタには 4 種類の方法があり、それぞれ一長一短がある。3 種類はフロー経路の権限を禁止して遮断するのに対し、Read 権限を許可する方法は情報共有の拡大の意味を持つ。以前は読めなかった客体が修正により普通に読めるようになれば、不正経路ではなくなるのでCovert Channel 自体は無くすことができる。情報フィルタの具体的な処理を以下にまとめていく。Covert Channel が発生して検出された場合、以下、図 2.2 の (1)(2)(3)(4) のいずれかを適用すればCovert Channel が解消される。(1) (S1,O1) のREAD 権限を削除。(2) (S1,O2) のWRITE 権限を削除。(3) (S2,O1) にREAD 権限を添付。(4) (S2,O2) のREAD 権限を削除。上記のどの情報フィルタを選択するかは各コミュニティのセキュリティポリシーや主体のアクセス履歴、ユーザがどういう方針で処理するか定めるユーザーポリシーを考慮して決定するが、(1) から (4) のどの場合でも Covert Channel は訂正できる。

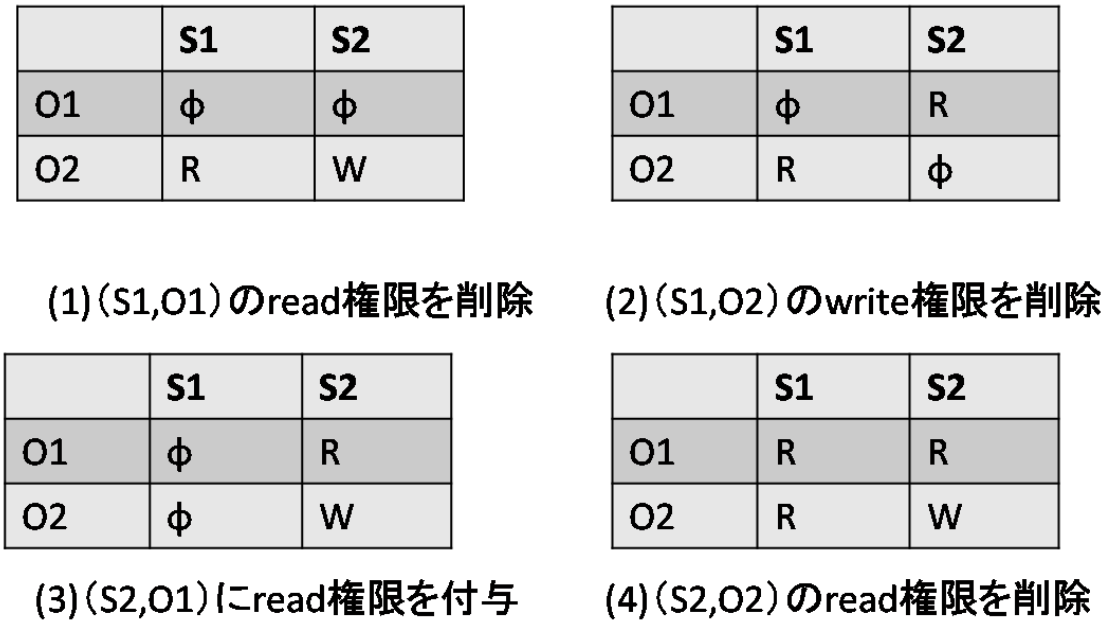

図 2.2: 情報フィルタ

2.6 検索エンジン

2.6.1 検索エンジン

検索エンジン [2] は大きく分けて，２種類ある．１つは「ディレクトリ型」もう一つは「ロボット型」である．

「ディレクトリ型」の検索エンジンは、サイト内容を人が判断して、カテゴリー別に探しやすい形整理して登録したものである．カテゴリー別のサイトのリストからアクセスしたいサイトを見つけ出しアクセスする．代表例としては「Yahoo! Japan」があげられる．

登録する際に、本当に価値がある情報かどうかという観点で人間のチェックが入るので、情報の信頼度・有用度は一般に高くなる．しかし一方，登録を人手で行う以上，登録できる量には限界がある．

インターネット上のサイトが爆発的に増えてしまった今となっては，この方式だけで大規模な検索エンジンを維持していくことは非現実的になりつつある．

「ロボット型」の検索エンジンはキーワードを指定して検索を行う

と，そのキーワードを含むページのURLの一覧が表示されるものである．Googleが代表例としてあげられる．

検索のもととなる情報は,「クローラ」「ロボット」などと呼ばれているプログラムが,インターネット上のサイトを定期的・自動的に巡回して,データベースに蓄積していく．人手を介さないので世界中の様々なサイトを検索対象とすることができるがそれだけに価値のある情報とそうでないものが混合されやすい．

Googleは,インターネット上の情報の価値を判断するために「ページランク」という概念を導入する．これは簡単に言うと「多くのページからリンクされているページは価値が高い情報が含まれているはずだ」「価値の高いページからリンクされたページは,同じように価値の高い情報を含んでいるはずだという考え方である．この考え方に従い検索対象をランク付けしたおかげで検索結果の有用性が高まることになる．

2.6.2 クローラ

クローラとはWeb上を自動的に巡回してWebページを収集する検索ロボットプログラムのことである。一般にクローラは、既知のHEML文書の新しいコピーを要求し文書中に含まれるリンクをたどり別の文書を収集するという動作を繰り返す。新しい文書を見つけた場合はデータベースに登録する。また、既知のファイルが存在しないことを検出した場合はデータベースから削除する。

2.6.3 インデックス

インターネット経由でダウンロードしたHTMLファイルは、検索可能な形でデーターベース化する必要がある。とてもシンプルな方法としては、ダウンロードしたHTMLファイルをそのままの形でディスク上に保存して、指定されたキーワードで文字列サーチするという方法があります。ですが、これは非常に時間がかかります。その

ため、実用的な検索エンジンを作るために、一般に次のような方法をとります。

1　HTML ファイルからタグ部分を取り除き、検索対象となる文章 (文字列) を取りだす。

2　文章 (文字列) を単語に分解する。

3　単語とそれを含む HTML ファイルの URL を、データベースに保存する。その単語が、URL を求める検索 (インデックス) の役目を果たす。

4　検索エンジンは、索引となるデータベースを検索して、指定されたキーワードに一致する単語を含む HTML ファイルの URL を特定する。

2.7　形態素解析と構文解析

形態素解析 [4][5] とは, 文章を意味のある単語に区切り, 辞書を利用して品詞や内容を判別すること. かな漢字変換や, 機械翻訳などに用いられる. コンピュータによる自然言語処理技術の一つ.

形態素とは文章の要素のうち, 意味を持つ最小の単位である.

例えば, ”She likes a cake or something like that “ に対し形態素解析を行うと

She/代名詞　like/他動詞　a/冠詞　cake/普通名詞 Or/接続詞　something/代名詞　like/前置詞　that/代名詞　./記号

といった情報が得られる. その後統語解析によって, 句間の修飾/非修飾関係といった統語的関係を明らかにする. 先の例であれば SHE は lile の主語　a　cake　と something　llike　that　が並列で名刺句になり,likes の目的語　といった情報が得られる.

WWW における検索エンジンなどでも, より正確な Web ページの特徴付けに形態素解析が利用されている.

一方, 日本語では単語ごとに区切らず続けて書くために, 形態素ごとの分割が難しい. 例えば, かな漢字変換の場合には, ひらがなのみ

で与えられた文章を区切る必要があるが, これは辞書を引きながら, 色々な区切り方を試していくことになる.

この時, 辞書にある名詞を形態素として区切ったり, 前後の品詞を見て文法的におかしい区切り方は省くなどの処理をするが, 複数の解釈が可能な文章もあり, 区切り方を一意に決定することはなかなか難しい. 特に長文になるほど区切り方の解釈が複雑になるため, ユーザの意図しない漢字変換をしてしまうことが増える.

構文解析では主に係り受けを解析する. 例えば子供の体力低下という文があった場合, 単語分割は子供｜の｜体力｜低下｜となり子供→体力　体力→低下と表される. 形態素解析構文解析両方を行うことで検索の精度が上昇する.

形態素解析, 構文解析エンジンとして茶筅, 南瓜等がある.

2.8 RDF

RDF[3][6] は, リソースの関係を主語, 述語, 目的語という 3 つの要素で表現する. トリプルの集合は RDF のグラフと呼ばれる. トリプルは「主語・目的語間の関係のステートメント（文）」を表すとされている. 例えば次の文

http://www.kubo.com には久保直也という作者がいますというのがあった場合,RDF の「文」として捉えると, 次のような要素から構成される

主語（Subject）	リソース	http://www.naoya.com
述語（Predicate）	プロパティ	作者
目的語（Object）	プロパティの値	久保直也

図 2.3: RDF 文の構成要素

RDF では, これらの関係を有向ラベル付きグラフを用いて表現する.

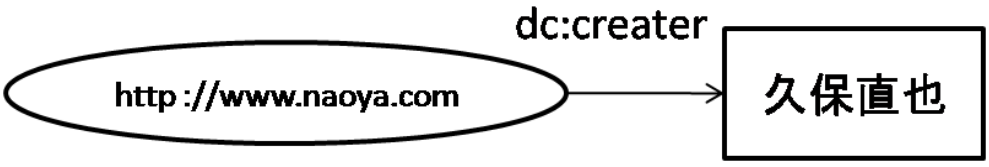

図 2.4: RDF のグラフ

RDF のグラフでは, リソースを楕円で, プロパティをアーク（矢印）で, リテラルを長方形で表現する. アークの矢印は主語を始点とし, 目的語を終点としている.

RDF では, 意味や定義を計算機にとって曖昧さなく示すために,URI 参照（URI とオプションのフラグメント識別子）による名前付けを行う. 前の例では, 主語はホームページなので URI で参照できることがすぐに分かるが, 述語（プロパティ）も,「作者」という単語ではなく,dc:creator という URI になっている. 語彙を表すために, 人によって使い方や与える意味が異なるかもしれない単語ではなく,URI 参照を用いることで, 確実な意味の交換を可能にしている.

ここでは目的語は文字列（リテラル）になっているが, 目的語も主語と同様,URI 参照で名前付けされるリソースとすることができる. つまり,RDF は主語, 述語, 目的語の全てを URI 参照によって名前付けして表現できるように設計されているわけである.

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xml:lang="ja">
  <rdf:Description rdf:about="http://www.kubo.com">
   <dc:creator>久保直也</dc:creator>
  </rdf:Description>
</rdf:RDF>
```

図 2.5: RDF を XML として表現する

この構文では,rdf:description　要素が「文」を示し,rdf：about 属性が「主語」となるリソースの URI を, 要素の内容が「述語」にあたるプロパティ（dc:creator 要素）と「目的語」となるリテラル値（神崎正英）を記述している（rdf:は RDF Model and Syntax の名前空間を示すものとします）.

2.9 オントロジー

2.9.1 オントロジー構成要素

Ontology[1][10] とは, 元来哲学の一分野であり日本語では存在論のことである．存在論は, 存在を体系的に扱う学問のことで形而上学の中でも常に重要な意味を占める分野である．人工知能の立場からは,「概念化の明示的な規約」と定義されている．概念化とは，対象世界に関して，興味を持つ概念とそれらのアイデアの関係とをさす．われわれが世界を認識し，そのモデルをコンピュータ内に作ろうとするときには，必ず世界を概念化する．すなわち，対象世界を構成する概念を洗い出し，それらの間の関係を整理し，概念を特徴づけ，他と区別する属性を明確にする．そして，それらの概念を記述するということである

2.9.2 オントロジーの構成要素

オントロジーは対象世界を説明するのに必要な概念と, それらの概念間の関係から構成される. 以下に, オントロジー構築する際に用いられる構成要素である is-a 関係・part-of 関係・attribute-of 関係・instance-of 関係について, 基本的な定義とその性質を述べる．1　is-a 関係

is-a 関係は概念の一般化-詳細化の関係を表している．例えば「昆虫」と「害虫」の間には害虫 is-a 昆虫という関係が成立する. この時, 一般化された概念 (ここでは「昆虫」) を上位概念, 詳細化された概念（ここでは「害虫」）を下位概念と呼ぶ.” 上位概念で定義された内容は全て下位概念でも成立する.” という継承の概念が利用される. また, 下位概念では上位概念の定義が詳細化される.

2　part-of 関係

part-of 関係は, ある概念とその概念を構成している部分に当たる概念との間の全体-部分関係を表す. この時, 部分に当たる概念を「部分概念」と呼ぶ. オントロジーにおける概念定義間 (class) の part-of 関

係は, モデル構築時に” それらの class の instance 間に全体-部分関係が存在する (可能性がある) こと” を定義している.

例えばトンボと. その構成要素（パーツ）である「複眼」の関係は複眼 (is-a)part-of トンボという関係が成立する. これはトンボと複眼の instance 間に全体-部分関係が存在する. すなわち、トンボの instance は複眼の instance を部分として持つことを表す.part-of 関係も, 概念を階層的に体系化する際に用いられる.

3 attribute-of 関係

attribute-of 関係は, ある概念を構成している属性情報, 即ち, 色・形状等を表す. 例えばトンボの構成要素である複眼の属性情報は丸い (is-an)attribute-of 複眼 (which is-a)part-of トンボという関係が成立する.

4 instance-of 関係

instance-of 関係は, 概念とその具体例 (instance) との間の関係を表す. 例えば, 害虫の instance である蚊は蚊 instance-of 害虫という関係が成立する.instance-of 関係には” x がある下位概念の instance であれば,x はその上位概念の instance でもある” という一般的性質がある. たとえば, 蚊は昆虫の instance かつ害虫 is-a 昆虫なので蚊は昆虫の instance でもある.

2.9.3 protege

ここでは Protege について説明する.

世界で最も有名かつ利用されているオントロジー構築支援ツール ユーザ登録数： 約 10 万人 (2008 年 7 月) protege により編集可能な要素として

ヘッダ要素

クラス要素

プロパティ要素

インスタンスがある.

第3章　提案

3.1　検索エンジンとCovert Channel

Covert Channel とは，ある object に権限がないのにも関わらず意図しない経路から情報が流出してしまう現象の事を言う.図1 の場合,object とはアクセスされる客体（データなど）の事をいい、subject とは object にアクセスする主体（ユーザーなど）である.R は read 権、W は write 権、φは権限なしの事である.矢印の流れで Subject3 が本来読めないはずの Object1 を読めてしまう. まず内的な ACL ではS3はO1を読むことができない.しかし外的要因を考慮した場合O1を読み書きできるS2がO8にコピペしてしまうことでSothersに読まれてしまう.S それをクローラが収集しそれをサーチすることでもしくは08から直接S3はO1を読むことができるようになってしまう.

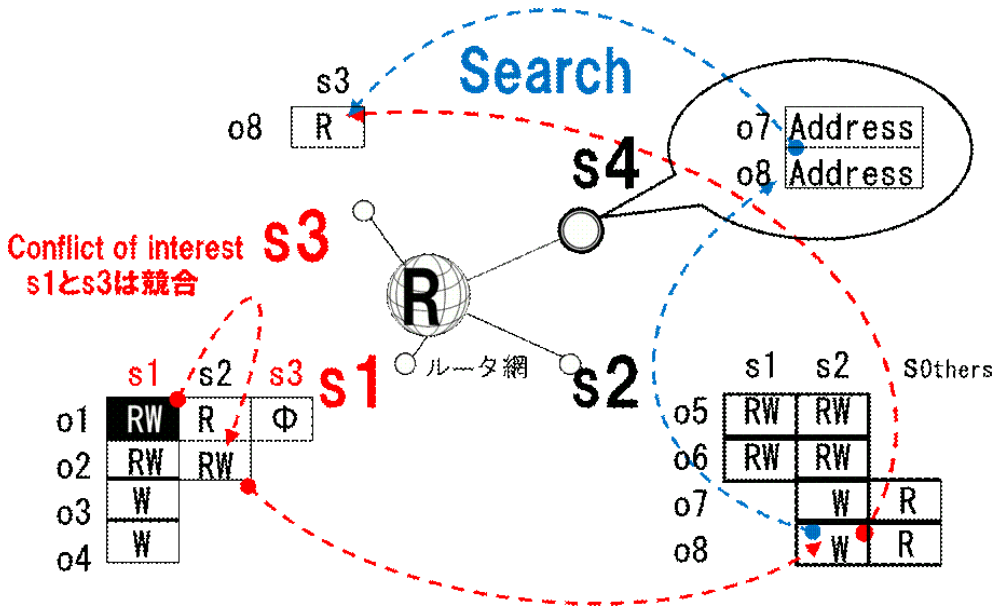

図 3.1: 検索エンジンと Covert Channel

3.2 Covert Channelの検出

従来のように把握したコミュニティの ACL のみを用いた Covert Channel の解析だけでは検出できないアクセス権の矛盾が存在する場合がある．そこで検査エンジンで得られた情報にオントロジーを用いたセマンティックな解析手法を用いることで ACL の矛盾や経路を効率よく見つけることを目的とする。

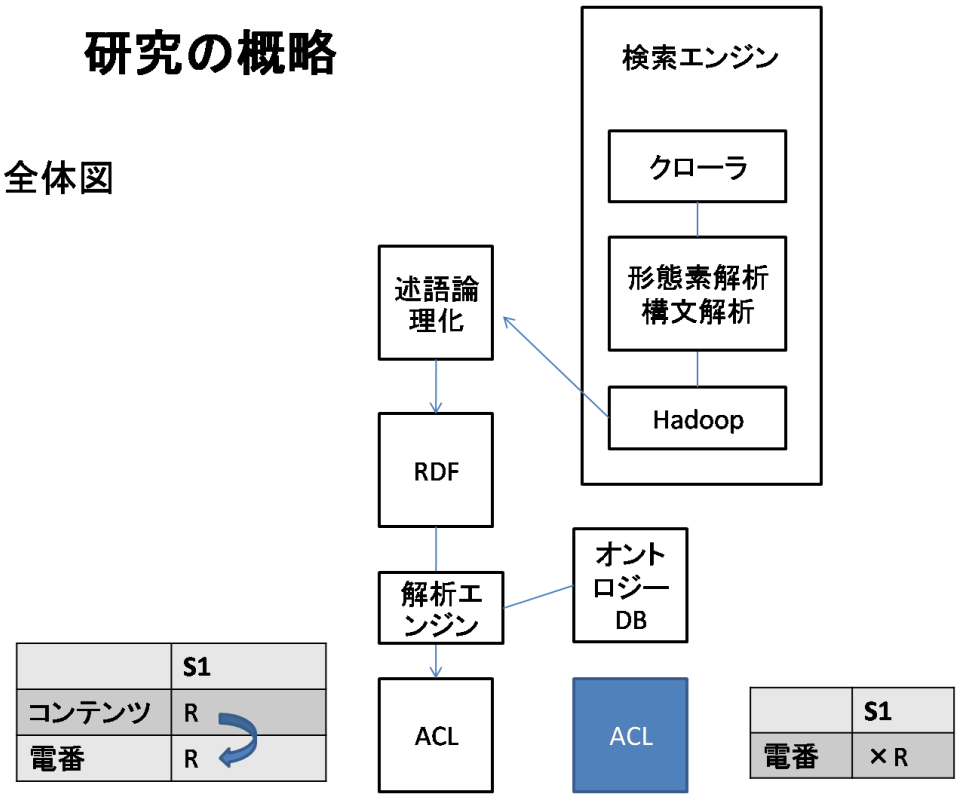

図 3.2: 検索エンジンによる Covert Channel 検出概略

3.3 Covert Channelの検出手順

ここでは検索エンジンを用いたCovert Channelの検索手順を示す.

１　クローラで収集された情報のタグを取り除き形態素解析、構文解析を行う.

形態素解析, 構文解析を行うことで, 検索の精度を上げることができる.

ここではHTMLファイルからタグ部分を取り除き、検索対象となる文章を取りだす．すると

koganeの日記 携帯 02/01の日記研究として使用します久保直也の個人情報は090-4216-0000です。

という文章が取り出せる．その文章を茶筅を使い単語に分解する。すると

kogane kogane kogane kogane 未知語
の の ノ ノ 助詞-連体化
日記 日記 ニッキ ニッキ 名詞-一般
携帯 携帯 ケイタイ ケイタイ 名詞-サ変接続
ホームページ ホームページ ホームページ ホームページ 名詞-一般
フォレスト フォレスト フォレスト フォレスト 未知語
02/01 02/01 02/01 02/01 未知語
の の ノ ノ 助詞-連体化
日記 日記 ニッキ ニッキ 名詞-一般
研究 研究 ケンキュウ ケンキュー 名詞-サ変接続
として として トシテ トシテ 助詞-格助詞-連語
使用 使用 シヨウ シヨー 名詞-サ変接続
し する シ シ 動詞-自立 サ変・スル 連用形
ます ます マス マス 助動詞 特殊・マス 基本形
久保 久保 クボ クボ 名詞-固有名詞-人名-姓
直也 直也 ナオヤ ナオヤ 名詞-固有名詞-人名-名
の の ノ ノ 助詞-連体化
個人 個人 コジン コジン 名詞-一般

情報 情報 ジョウホウ ジョーホー 名詞-一般
は は ハ ワ 助詞-係助詞
090-4216-0000 090-4216-0000 090-4216-0000 090-4216-0000 未知語
です です デス デス 助動詞 特殊・デス 基本形
というように分解することができる.

<html> <head> <meta http-equiv="Cache-Control" content="no-cache"> <meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS"> <meta Name="keyword" CONTENT=""> <title>koganeの日記 携帯ﾎｰﾑﾍﾟｰｼﾞ ﾌｫﾚｽﾄ</title> </head> <body bgcolor="#ffffff" text="#000000" link="#ff3333" vlink="#0000ff" alink="#00ff00"><tt>
 02/01の日記
  11:53 研究として使用します
――――――
 久保直也の個人情報は090-4216-0000です。

 □ 日記を書き直す
 □ この日記を削除
 <hr size="1" color="#000000">[戻る]
 <hr size="1" color="#000000"> <img src="/img/ad/papyless/pc/9-3105-200-46806001.gif"border=0>

 声優＋ｱﾆｿﾝｼﾝｶﾞｰ
両方学べる学校に行く
  <hr size="1" color="#000000"> (C)フォレストページ
  </tt> </body> </html>

図 3.3: クローラで収集された HTML 文書

図 3.4: chasen を使用して形態素解析を行う

2 Hadoop を用い計算処理を分散して行う

Hadoop は，Google 検索システムにおいて大量の「メタ言語のインデックス」を整理分類する．インターネット内に散らばったリソースのファイル名，ファイル内容の語を収集分析し，インデックスとしてまとめる機能．Map フェーズと Reduce フェーズの 2 つから成り，計算処理を分散して行う．Hadoop は，2 つのフェーズを Mastar と呼ばれるノードが Worker と呼ばれるノードへ処理を割り振り，分散処理を行う．

Map フェーズでは，膨大な文書等を複数の Worker(分散処理を行うマシン) 上で動く Map タスクへと分配する．Map タスクは並列実行により時間短縮が可能，渡された文書を単語毎にキーと値の対を出力する．つまり，Map とは，情報の分解・抽出を行う関数で，このフェーズで大量の情報を分解し，必要な情報を抜き出して出力する．

この時，Map 関数によって key/value のペアの集合が生成される．

Reduce フェーズでは，Map タスクで出力されたキーと値の対を収集し，キー毎に関連付けられた値を列挙しカウントし，その総計を取る．Reduce とは，Map フェーズで抽出された情報を集約し，それに対して計算を行い結果を出力する．尚，Reduce フェーズは分散処理させる事は出来ない．

形態素解析した結果を分散処理させることで効率化させることが可能である．

3　形態素解析, 構文解析された情報を述語論理化 RDF 化する.

述語論理化して,RDF 化しなければ意味まで考慮したマッチングが取れないため RDF 化を行う.

例えば久保直也の電話番号は 090-4216-0000 です, という文があった場合久保直也を主語 (リソース) 電話番号を述語 (プロパティ)090-4216-0000 を目的語 (プロパティの値) となる.

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf=http://www.w3.org/1999/02/22-
rdf-syntax-ns#

xmlns:mr3=http://mmm.semanticweb.org/mr
3#

xml:base="http://mmm.semanticweb.org/mr3
#">
 <rdf:Description rdf:ID="久保直也">
  <mr3:電話番号>090-4216-0000</mr3:電話
番号>
 </rdf:Description>
</rdf:RDF>
```

図 3.5: RDF の記述例

4　オントロジー DB を記述しておく.

例えば090-4216-0000instance-of電話番号is-a個人情報is-a久保直也といった記述ができる. ここではオントロジー DB の記述にはprotegeを使用する.

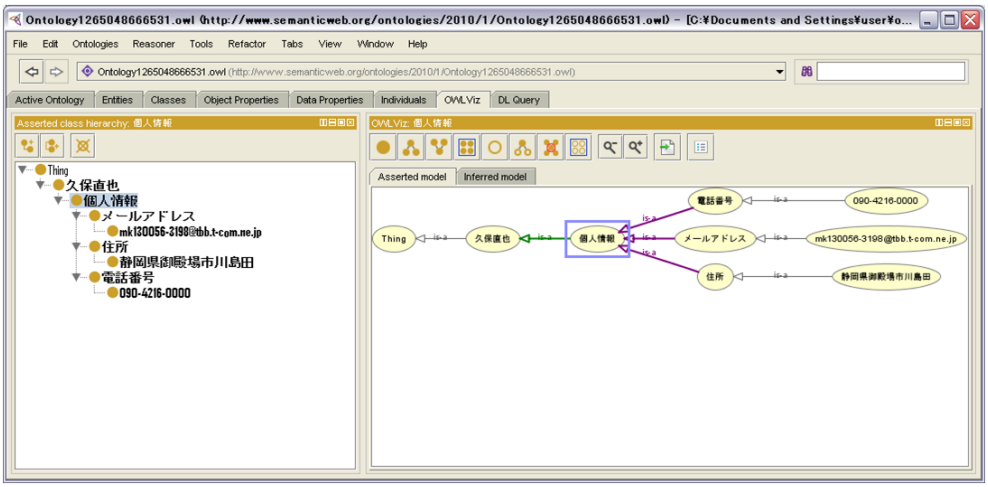

図 3.6: protege による記述

5　ACL を導く

RDF で検索された処理結果とオントロジー DB がら外的要因を考慮した ACL を導き出す解析エンジンにより ACL を検出する.

6　Covert Channel の検出

以上により,内的な ACL では読めないことになっている情報が Web 検索の結果を解析して得られた外的要因まで考慮した実質的な ACL では読めると言ったような矛盾を見つけることができる.

さらに矛盾があった場合 Covert Channel の経路を表示される．下の図を例にとると S1 が読み書きのできない電話番号情報を S2 が読み書きすることができた場合 S2 がその情報を読み S1 が読み書きのできるコンテンツにコピペしてそこから S1 が本来読み書きのできない電話番号情報を読み書きすることができるようになってしまった.といった経路を表示する.

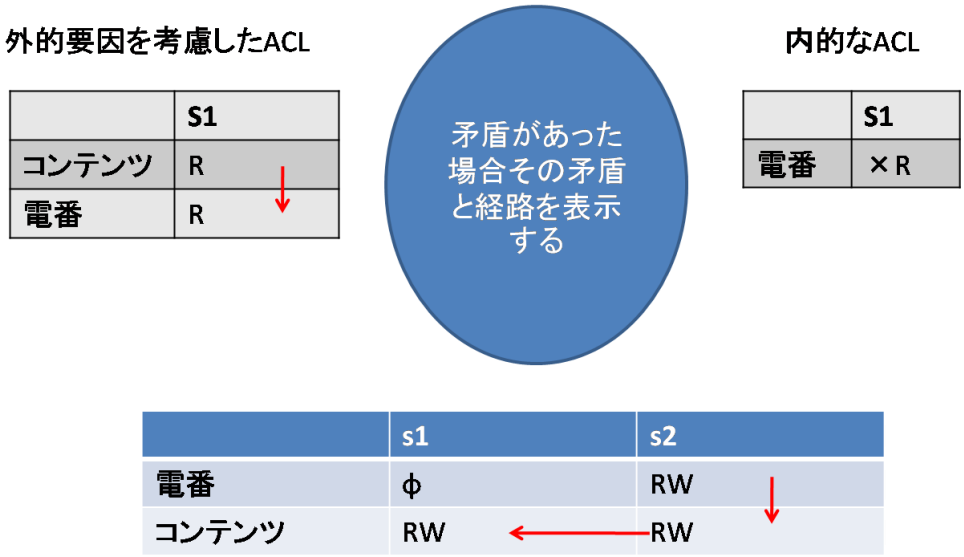

図 3.7: Covert Channel の検出

第4章　まとめ

本稿では検索エンジンを用いたCovertChannelの検出方法を提案した．これにより従来のように把握したコミュニティのACLのみを用いたCovertChannelだけでは検出できないアクセス権の矛盾が存在する場合も検索エンジンによって得られた情報にオントロジーを用いたセマンティックな解析手法を適用することで外的要因を考慮した場合のACLの矛盾や経路を効率よく見つけることが可能となり,従来のCovert Channel解析法での外的要因を考慮した場合検出できないアクセス権の矛盾が存在する,という問題点を解決することができるかもしれない.

これからの課題としてHadoopによるRDF導出効率化やRDFで検索された処理結果とオントロジーDBから外的要因を考慮したACLを導き出す解析エンジンの構築,ACLの矛盾や経路を検出するのに最も適したオントロジーの記述法の検討など検討していきたい.

第5章　謝辞

本研究を行なうにあたり，終始熱心に御指導していただいた木下宏揚教授と鈴木一弘助手に心から感謝致します．また，様々な面で数多くの有益な御助言をしていただいた東洋ネットワークシステムズ株式会社の森住哲也氏に深く感謝致します．さらに，公私にわたり良き研究生活を送らせていただいた木下研究室の方々に感謝致します.

関連図書

[1] 溝口理一郎, 古崎晃司, 來村徳信, 笹島宗彦：”オントロジー構築入門”オーム社 (2006)

[2] 星澤隆:”Ruby で作る検索エンジン”毎日コミュニケーションズ (2009)

[3] 神崎正英：”セマンティック・ウェブのための RDF/OWL 入門” 森北出版株式会社 (2005)

[4] ”ChaSen‘s Wiki”,http://chasen.naist.jp/hiki/ChaSen/

[5] 浅原正幸　松本裕治：”「茶筅」/「南瓜」を用いた形態素解析・係り受け解析”,http://chasen.naise.jp/chaki/t/2006-09-05/slides/enshu-slides-4up.pdf

[6] 神崎正英”セマンティック HTML/XHTML” ,http://www.kanzaki.com/book/shx/

[7] 戸田瑛人 森住哲也 鈴木一弘 木下宏揚”検索システムに組み込むセキュリティモデルに関して”社団法人 電子情報通信学会 (2009)

[8] 小松充史:”Covert Channel 分析制御のために推論を導入した情報フィルタに関する研究”2006 年度神奈川大学修士論文.

[9] 西田　学:”Covert Channel 分析と情報フィルタ選択のための推論機能の提案”2007 年度神奈川大学卒業論文

[10] 山下 祐平:”民具資料の Ontology 構築に関する研究”2006 年度神奈川大学卒業論文

第６章　質疑応答

Q1　オントロジーの記述があったが何オントロジーを構築したのか。また構築した数が少ないようだがそれはどうしてか。

個人情報オントロジーを構築しました。ここでは例として構築しただけなので１つの例としてしかオントロジーを書かなかったから。

Q2　ACL は最初から記述されている前提なのか

前提です。内的要因では読めないとされています。