情報流通における仲介サーバを利用した情報利用制限方式

5Q－06

三上 範賢　　中川 香織　　梶原 清彦
NTT 東日本 研究開発センタ
NTT 西日本 研究開発センタ

1. はじめに

近年企業間でインタネットを利用してデータを交換するいわゆる企業連携が活発化してきている。顧客データなどは、社外だけでなく企業内からも特定の人しかアクセスできない環境が重要になっている。

本稿では、仲介サーバを利用しデータ受信時に企業側の社内システムにデータを自動登録できる環境における登録の制限方式を提案する。

2. データ流通環境

以下で情報利用制限を検討する。

2.1. システム連携プラットホーム Bespa

Bespa の基本構成を図 1 に示す。

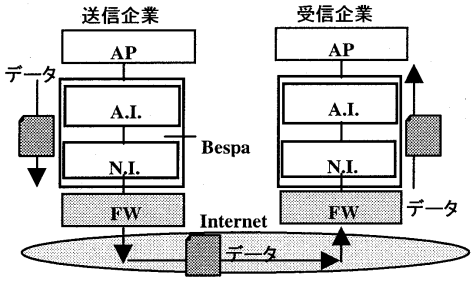

図 1 ：Bespa の構成図

Bespa は各アプリケーション（AP）に対応するアプリケーションインターフェイサ（A.I.）と Bespa 間のデータ通信をするネットワークインターフェイサ（N.I.）で構成される。A.I.は各 AP に対応しデータ変換を行うモジュール群がある。N.I.はファイアウォール（FW）通過のためのプロトコル変換等 Bespa 間の通信を行う。

2.2. Bespa における情報利用制限の定義

本稿の情報利用制限を次のように定義する。

(1)社内の特定の人だけが他社からのデータをアクセスできるようにする。

(2)社内でのデータ管理は既存の社内システムで行われている。

2.3. 情報利用制限の解決すべき課題と問題点

企業間のデータ送受信における課題と解決すべき問題点を示す。

課題 1：他社からのデータを特定の社内システムに安全に登録すること。

課題 2：社内外からの攻撃に対処すること。

3. 課題の解決方法

本稿で提案する方法は図 2 のような手順で実行される。

(1)送信 Bespa データと、データを処理するモジュールのスタンプを送る。

(2)受信 Bespa はデータに対応するモジュールを特定するため、設定ファイルを調べる。(受信 Bespa は開始時にモジュールローディング時にスタンプを計算しておく)

(3)そのモジュールのスタンプと、送られてきたデータに付属するスタンプが一致した場合のみ、そのモジュールを実行する。

このための準備として以下の作業が必要である。

(1)システムへの登録モジュールをデータを送る企業、受け取る企業間で確認・合意する。

(2)受け取る企業の Bespa（受信 Bespa）へ合意したモジュールを登録する。

(3)登録モジュールの一方向ハッシュ関数値（スタンプ）を、データを送る企業の Bespa（送信 Bespa）に登録する。

Data flow Control between systems in different companies with mediator.

Norikatsu Mikami,Kaori Nakagawa,Kiyohiko Kajihara
NTT East Corp,Research and Development Center
NTT West Corp,Research and Development Center

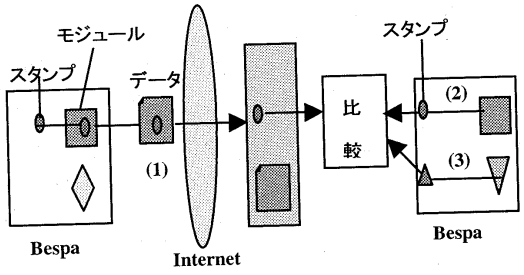

図2：利用制限のための手順

4. 本方式に対する攻撃の対処方法

本章では本方法に対して想定される攻撃の種類について検討し、その対処方法を述べる。

Bespa への攻撃として社外から(a) 盗聴、(b) 不正データの送り込み、(c) Bespa の改竄が考えられる。また、社内からは(d) 盗聴、(e) 不正データの送り込み、(f) Bespa の改竄、(g) 暗号鍵の略奪の危険性がある。

4.1. データ攻撃への対処方法

(a)、(d)は Bespa のデータの暗号化で対処済み。(b)は暗号鍵が不明なため、正常データを作成できず、対処済み。(e)はスタンプが知られていることを想定すべきだが、暗号鍵が適切に管理されているなら暗号鍵が不明なため、正常データを作成できず、対処済み。(c)は Bespa が実行されている計算機へのリモートログインを禁止するように FW を設定することで対処可能。Bespa が受け付けるデータにより Bespa の設定を変更することはできないため、特殊なデータを用いた攻撃は想定外。(バグは考慮外) (f)は共用計算機での Bespa の運用を想定し、以下で更に詳細に検討する。

-(f-1)Bespa の設定変更

適正なモジュール以外に設定しても、そのモジュールのスタンプが異なるためエラーとなり、指定したモジュールは動作しない。よって対処済み。

-(f-2)Bespa のモジュール入替え

設定を変えずに、指定されたモジュールを入れ替えても、やはりそのモジュールのスタンプが異なるため、エラーとなりモジュールは動作しない。よって対処済み。

-(f-3) Bespa のスタンプを確認するコードを Bespa に埋め込むことや、Bespa に対する証明書を作成し、Bespa がそれを開始時に確認する等、技術的には改造の発見が可能である。

(g)は暗号鍵を指定するファイルを暗号化しておき、Bespa プログラムがそのファイルを参照する際に復号化すること等で対処可能である。それゆえ、(f)(g)はプログラムにより保護しても、Bespa の管理者をだまし、モジュールを入れ替えさせる、暗号鍵を聞き出す等が可能である。Bespa が動作する計算機のアカウント管理、設定変更の安全性の向上などが不可欠だが、技術的な対処をする価値は低いと判断する。

4.2. スタンプ攻撃への対処方法

4.1.項目以外の攻撃への対処方法を示す。

(1)送られてくるデータに付属するスタンプの変更

送られてくるデータに付属するスタンプを変更でき、(f-2)のモジュールの入替えができればデータを不正利用できる。スタンプを変更できなくするために、スタンプもデータ同様に暗号化の対象部分とする。

(2)スタンプ確認コードの変更

これは(f-3) で示したように、Bespa の安全な運用体制により対処するのが妥当である。

5. おわりに

本稿では、企業間でデータを送受信する場合に、受け取る企業側の社内システムにデータを自動登録できる環境における登録の制限方式を提案し、その安全性について検討した。今後は総合的な運用の検討をふまえて開発を行う。

参考文献

[1] 梶原,小山：ビジネスイベントによるシステム連携機構 Bespa の提案, NTT R&D, Vol46 No.6, pp571,1997