GateKeeperへのフィルタ管理機能の追加

2008MI234 鈴木 雅喬

指導教員　後藤 邦夫

1　はじめに

近年，web サービスやインターネットを利用したアプリケーションが普及している．これらの普及に伴い，その安全性を保つのが非常に重要となってきている．

本研究は，その対策として「段階的通信制限システムの拡張」[4][2] のゲートキーパー (以下，GK) へ，攻撃ホストの管理機能を実装する．

2　本研究の概要

この節では，本研究で提案するシステムの概要を説明する．本研究では，この GK に対して用いることのできるデータベースとそのデータベースを利用して，GK を操作するコマンドを作成する．

本研究では，操作する従来のコマンド入力の補助とデータベースを利用したコマンドの拡張を目的として，フィルタ管理機能を追加する．フィルタ管理機能とは，今までに GK で利用したフィルタに管理番号を付けてデータベース上で管理し，その情報を利用して GK の操作をする．

データベース上でフィルタデータを管理して，いつ誰がどのようなフィルタを作成したかを管理して，視覚化することにより，GK の利用を円滑にすることができる．言語は C++ プログラミングを用い，postgresql への接続ライブラリとして libpq を使用する．

3　システムの実現

本研究で追加するシステムを実現するために，実装するものを説明する．

- データ蓄積用のデータベースの作成
- データベースへアクセスし，データベースを操作するプログラム
- SSLCommander へのコマンド追加

3.1　データベースの作成

データベースソフトウェアに postgresql[1] を利用する．Commander がフィルタを GK に入力する際に用意する 13 の項目に加えて，データの 1 対 1 の識別記号として用いる管理番号とユーザー名の 15 の項目を扱う．ユーザー名は，SSL 認証を用いた遠隔操作 [4] でフィルタを作成した際の，作成者を保存するためのものである．管理番号には各 PC のユーザー名を先頭に，重複を避けるために後ろにアルファベット一文字，末尾に番号を付与し，それぞれを：で区切るという形をとる．

3.2　プログラムの作成

GK は，SSLCommander によって操作することができる．SSL クライアント認証を用いた通信と外部ホストからの要求でフィルタ操作をクラスである SSL クライアント認証の処理は run メソッド，フィルタを操作するのは accessFilter メソッドに記述されている．

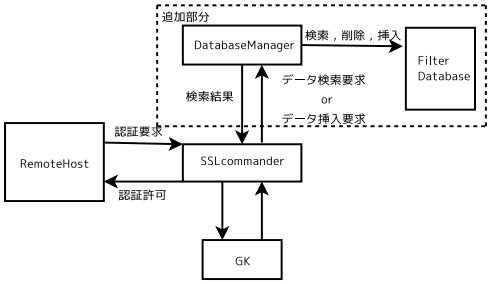

図 1　改修後の SSLCommander

3.3　データベースへの書き込み

ユーザー名は SSL 認証を用いたユーザー名を利用する．管理番号は，そのユーザー名をデータベース上の情報と照合して，識別情報として付与する．日付は，そのフィルタに関するデータが挿入される直前の，GK を起動させている PC のローカルタイムから取得する．

3.4　コマンドの追加機能

既存システムのコマンドに加えて，以下の処理をするコマンドを追加する．

- ADD コマンドを用いた際の，通信制限の決定をデータベースのデータを利用して設定．
- 特定の管理番号のフィルタのデータからフィルタを作成する BAC コマンドを追加．
- クライアント側でデータベースの中身を閲覧できる，SHOW コマンドの追加．
- 特定のフィルタをデータベース上から削除する DDEL コマンドの追加．
- データベースのフィルタに特定条件で検索をかけて，各制限が何回掛けられたかをクライアント側での表示をする COUNT コマンドの追加．

これらの処理を追加する上で、新たに DataBaseManager というクラスを作成する．

データベースの情報から通信制限を決定する際の計算方式には，もっとも厳しい通信制限を探す MOST，今までの累積で計算する TOTAL，一番掛けられた回数の多い制限にする MEDIAN，平均から求める AVERAGE の 4 タイプを用意する．

この 4 タイプの計算を示す文字列を ADD コマンド入力の通信制限の欄で入力することにより，データベースへアクセスし，格納されている情報と計算方式から通信制限を決定することができる．

4 実験

組み込んだプログラムの動作テストの実験に関して述べる．実験環境には Ubuntu10.04(32bit OS) をインストールした PC を使用し，後藤研究室で開発中のネットワークエミュレーター GINE[3] を用いて実験環境を構築した．

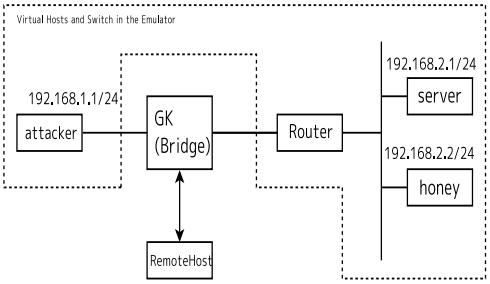

図 2　実験ネットワーク図

今回のテストでは，追加で拡張と実装したコマンドが正しく動作し，通信制限を行えているかを確認する．値の受渡しが正確にされているかを，クライアント側，サーバー側，データベース側で順を追って確認した．

```
ADD|4|IN|192.168.1.1|255.255.255.255|192.168.
2.1|255.255.255.255|ANY|0|0|TOTAL|1.0|0|1
```

上記のコマンドを実行したところ，通信制限は DELAY が設定された．パラメーターから，1.0msec の遅延が 192.168.1.1 から 192.168.2.1 への通信時に発生することとなる．

実際に遅延が発生されているかを，ネットワークテスト時と同じように iperf を利用した TCP 通信で確認する．

```
Attacker : iperf -c 192.168.2.1
------------------------------------------
Client connecting to 192.168.2.1, TCP port 50
01
TCP window size: 16.0 KByte (default)
------------------------------------------
[108] local 192.168.1.1 port 50421 connected
with 192.168.2.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[108]  0.0-10.0 sec  560 KBytes  458 Kbits/s
ec

Server : iperf -s
------------------------------------------
Server listening on TCP port 5001
TCP window size:   85.3 KByte (default)
------------------------------------------
[  9] local 192.168.2.1 port 5001 connected w
ith 192.168.1.1 port 50421
[ ID] Interval       Transfer     Bandwidth
[  9]  0.0-10.0 sec  560 KBytes  458 kbits/s
ec
```

正常通信と比べ，通信速度が落ちているため，正しく通信制限がされていることが分かる．

また，その他 3 タイプの計算方式による ADD コマンド，新しく実装した，BACK，SHOW，COUNT，DDEL コマンドいずれも正常に動作していることを確認した．

5 おわりに

本研究では，GK に攻撃ホストを管理するデータベースの実装とデータベースを利用した通信制限を施すプログラムを追加した．これによって，Commander の機能の拡張に成功した．

今後の課題として，IDS 等の脅威を自動的に検知するシステムを組込み，自動的にフィルタリングルールを作成する機能を追加することでより効果的なシステムを構築することが出来ると考える．

参考文献

[1] Joshua D. Drake: Postgresql :Welcome http://www.postgresql.org/(accessed Apr.2011).

[2] 青山　正樹，小島　正和：段階的通信制限を実現するゲートキーパーの提案と試作，卒業論文，南山大学 数理情報学部 情報通信学科 (2006).

[3] 浅野　洋介:大規模ネットワーク構築のための GINE の管理機能の追加，修士論文，南山大学 大学院 数理情報研究科 数理情報専攻 (2010).

[4] 福井　麻美：通信制限システムにおける TCP セッションの途中切替と安全なリモートアクセス機能の実装，修士論文，南山大学 大学院 数理情報研究科 数理情報専攻 (2010).