

ネットワークセキュリティ管理のための ネットワーク可視化システムの実装

幅野莞佑 *

黒米祐馬 †

武田圭史 ‡

村井純 ‡

January 13, 2017

1 序論

近年, 技術の進歩によって, ネットワークはより身近なものになっている. サーバを手元に持たなくともその運用に携わるユーザが増加傾向にある. しかし, それに伴ってセキュリティの知識が十分に持っていない人の管理するサーバが存在するようになっていく. サーバのセキュリティを保つには, ファイアウォールの設定が重要である. ファイアウォールは設定ファイルにあるルールに従って通信しようとするパケットを遮断できるが, その挙動を確認する手段などは備わっていない. ファイアウォールの挙動やルール, そしてネットワークの状況などを把握するには管理者がネットワークについてある程度の知識をもっている必要がある.

2 提案手法

本研究ではネットワークの知識を十分に持っていない個人サーバ管理者に対して, ネットワークとファイアウォールの挙動をリアルタイムで確認し, 状況の理解を支援する可視化システムを提案する. 現状, ネットワークやファイアウォールの状況を確認する方法としては設定ファイルやパケット情報, ログ・ファイルなどの文字形式の情報で確認することが挙げられる. その情報を文字ではわかりにくかった通信の状況や, ファイアウォールの挙動を同時に確認することができる.

3 実装

本システムはパケットキャプチャプログラムと, 可視化プログラムから構成される. パケットキャプチャプログラムは監視対象のサーバでパケットを分析し, 可視化プログラムはクライアントとしてパケットの分析結果を取得, 3D 描画する.

3.1 パケットキャプチャプログラム

表 3.1: パケットキャプチャの実装環境

要素	環境
OS	Ubuntu 14.04
言語	C
コンパイラ	gcc version 4.8.4
ライブラリ	libpcap version 1.5.3 mysql version 5.5.53 netstat 1.42[1]
ファイアウォール	iptables version 1.4.21[2]

このプログラムで実装するプロセスを以下に述べる.

1. パケットのデータ収集
2. 収集したパケットの必要なヘッダデータの抽出
3. netstat コマンドによりポートの状況を取得
4. iptables の設定ファイル, ログファイルを取得
5. データを可視化プログラムに取得した各データをソケットで送信

パケットのデータ収集には libpcap ライブラリを用いた. 取得する情報は送信元 IP アドレス, 送信先 IP アドレス, 送信元ポート番号, 送信先ポート番号, プロトコル, TCP フラグ情報の 6 つである. これらの情報をリアルタイムでソケットで送信する. ここで, 送信相手のアドレス毎にデータベースで通信回数などの状況を管理する. また, 通信回数が多くなった場合, 可視化プログラムに負荷をかけないようデータベースにある情報を一定時間ごとに取得して送信するようにする. そして一定時間で通信回数でソートされた IP アドレスのリストを送信する.

iptables は設定ファイルから filter テーブルを参照し, パケットのフィルタ基準を取得する. このとき, 本システムは設定ファイルの変更があれば取得して, 送信する. また, iptables がドロップしたパケット情報などをログファイルから取得することによってパケットをドロップする挙動を可視化することが可能になる.

*慶應義塾大学 総合政策学部

†慶應義塾大学 環境情報学部

‡慶應義塾大学

3.2 可視化プログラム

表 3.2: 可視化プログラムの実装環境

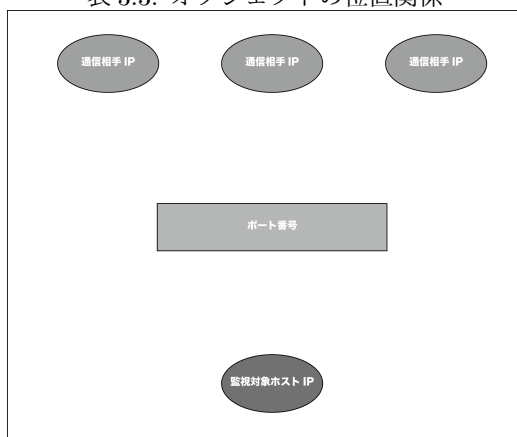
要素	環境
OS	Mac OS X El Capitan 10.11.6
統合開発環境	Unity version 5.4.1[3]
言語	C #
asset	Spicy Pixel Concurrency Kit 1.1.4

このプログラムで実装するプロセスを以下に述べる．

1. パケットキャプチャプログラムからソケットでデータを受信
2. ポートの開放状況を描写
3. 受信したデータを元に通信アニメーションを3D表現
4. ファイアウォールがドロップしたパケットのドロップアニメーションを表現
5. ファイアウォールの設定情報を更新して表示
6. キャプチャ部側に設定したいファイアウォールのフィルタ情報を送信

可視化プログラムはパケットキャプチャプログラムを3Dで表現する．実装にはゲーミングエンジンであるUnityを用いた．表現方法は3D空間にアドレスを表示した通信相手、ホスト、ホストのポートの3種類のオブジェクトを設置して、通信を表現する．そして、ホストと通信相手のオブジェクトを対照的にしてポートオブジェクトを挟むようにする．

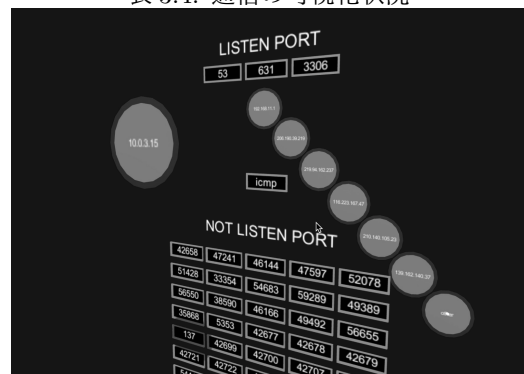
表 3.3: オブジェクトの位置関係



通信の挙動はパケットキャプチャプログラムから受信した通信情報のデータを元に送信元IPに対応したオブジェクトからホストのポートを介して送信先IPに対応したオブ

ジェクトへ移動するようなパケットオブジェクトを表示して、実現する．ここで、リアルタイムでパケットの通信アニメーションを描画しようとする、パケットの多さによって負荷がかかってしまう状況がある．大量のオブジェクトを生成して削除するという事は大きな負荷がかかってしまうため、事前に100個ほどパケットオブジェクトを生成して、その表示、非表示をくりかえすことで負荷を軽減させるようにした．通信の多寡をわかりやすくユーザに提示するため、各IPアドレスごとの通信回数をソートしたリストを一定時間間隔で受け取り、通信回数の多い順に通信相手オブジェクトを高い位置に設置した．また、これらのアニメーションを見ながら、ファイアウォールを設定できるようにiptablesのフィルタリングルールを可視化プログラムからパケットキャプチャプログラムで送信できるようにする．

表 3.4: 通信の可視化状況



4 あとがき

本研究では小室氏のシステム[4]を元にして、実装を行った．それに新たにくわえて、可視化システムを見ながら、ファイアウォールのフィルタリングルール設定を作り設定できるようにした．これによって、ファイアウォールの挙動を確認しながら、ファイアウォールの設定を変更しネットワークの管理をより円滑に容易に行うことができると考えられる．

参考文献

- [1] netstat: <http://www.5group.com/netstats/>
- [2] iptables: <http://www.netfilter.org/projects/iptables/index.html>
- [3] Unity: <http://unity3d.com/jp/unity>
- [4] 小室 研人：設定支援のためのネットワーク可視化システムの開発, 2015.