通信制限システムの再設計と安全な遠隔操作

2006MI128 小田嶋 晃
指導教員　後藤 邦夫

1　はじめに

近年，インターネットでは，DoS 攻撃 (Denial of Service attack) や迷惑メール等の迷惑行為が増加している．後藤研究室では，この問題の対策手段として「段階的通信制限システム」(以下，既存手法，または GK とする [2]) を開発してきた．本研究では，後藤研究室で開発中の GINE[1] を利用することで既存手法の再設計をし，SSL(TLS) クライアント認証 [3] を利用することで安全な遠隔操作を可能とすることが目的である．

2　システムの再設計

本節では，本研究で提案するシステム (以下，GK3) の概要を既存手法とあわせて GK と GK3 の基本的な構成，システムの拡張点，通信の流れを述べていく．

表 1 に既存手法と本研究のシステムの違いを示す．

表 1　GK 再設計のポイント

旧 GK	GK3
GINE3 とは別	ほとんど GINE3 のクラス
TCP 途中切替機能無し	途中切替追加
リモートアクセス認証	SSL クライアント認証

既存手法では一部 GINE ライブラリを利用していたが，そのプログラムはほとんどが，GK 用に作成したクラスによるもので構成されていた．そのため，GINE で利用されている部品を再利用し，安定性をはかることが必要である．また，既存手法では TCP の NAT 途中切替えが未実装であったが，本研究で NAT クラスにおいて TCP 途中切替えを実装した．そして，クライアント認証の部分で用いられているパスワード認証では次の 3 点の問題が上げられる．

パスワードを知っていればパスワード認証を突破可能であること，誰かがパスワード認証を突破していればルールの変更が誰でも可能であること，GK 側がクライアントに対するアクセスの制限機構を持たないことである．これらの問題点を解決するために，SSL クライアント認証を追加する．SSL クライアント認証ではクライアントの鍵やクライアント証明書等が必要であり，セキュリティをパスワード認証より向上できる．そして GK の Commander との通信の問題点を改善し，安全性を高め，安全な遠隔操作を実現する．GK3 の構成を図 1 に示す．

フレームが NIC0 に届くと, PFPacketIn に入り, FrameQueue を通り GKBridge に渡される．GKBridge で

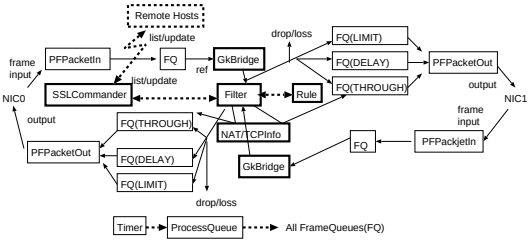

図 1　GK3 の構成

Filter のルールと参照し，ルールにマッチするかどうか検索する．NAT のルールで適合した時は，NAT クラスが呼び出される．リモートホストは SSLCommander を通じて, ルールのリストを変更したり, アップデートすることが可能である．そして各 Queue は PFPacketOut にフレームを渡し，NIC1 へフレームを送信する．逆方向の通信に対しても，同様の手法で扱われる．

3　システムの実現

本節では，本研究でのおもな変更点の詳細を述べる．

3.1　システムの再設計

GK3 で GINE クラスライブラリの利用方法を述べる．FrameQueue は，リモートホストにより設定されたルールに則り，Through，Delay，Limit，Drop，Loss，NAT 等通信制限を加える．PFPacketIn は，パケットの入口でパケットを送り，GKBridge(GKForwarder) へと送る．PFPacketOut は，FrameQueue がルールに従い設定した通信制限を受け，パケットを相手ホストへと送信する．

3.2　TCP NAT 実装

図 2 はフレームの流れを示す．

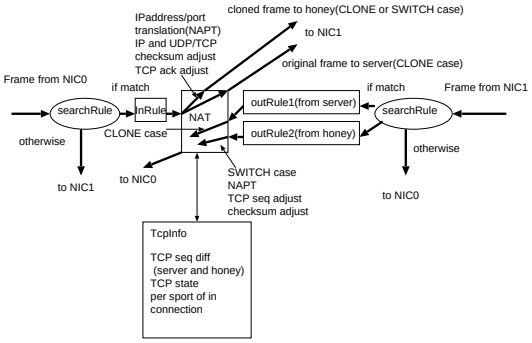

図 2　アドレス変換処理におけるフレームの流れ

NAT でルールがマッチしたときには inRule と outRule1 と outRule2 が作成される．outRule が 2 つ必要な理由は，Server と Honey の両方との通信のシーケン

ス番号等を記憶する必要があるからである．そして各ルールを TcpInfo で記憶する．CLONE のルールの時には，コピーしたフレームを Honey へ送信し，オリジナルのフレームを Server へ送信する．

3.3 安全な遠隔操作

安全な遠隔操作を実現するために，SSL クライアント認証を導入する．私設 CA を作成し，サーバ/クライアントそれぞれが鍵を作成し，CA 署名済み証明書を作成する．クライアント証明書が正規のものであるかを確認する方法は，`SSL_get_verify_result` 関数を利用する．CommonName の確認方法は，テキスト形式で CommonName が記述されているファイルを読み込み，`X509_NAME_get_text_by_NID` 関数で `peer_CN` を取得し，その名前が CommonName とマッチするかを確かめる手法である．

4 実験結果

本節では，SSL クライアント認証を含めた GK の通信実験に関して述べる．

4.1 実験環境の構成

本研究では，Ubuntu 8.10(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 GINE を用いて実験環境を構築した．

4.2 NAT 処理の実験

図 3 は，NAT 処理の実験環境を示す．

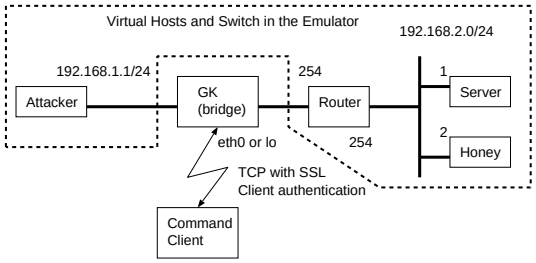

図 3　1PC での NAT 処理実験環境

後藤研究室が開発中のネットワークエミュレータである GINE3 を利用した実験ネットワークを構築したものである．このネットワークではホスト PC1 台で実験することが可能である．

Attacker, Server, Honey, Router は Network Namespace を利用した仮想端末で実装している．OS では GK がブリッジとして動作し，Attacker と Router をつないでいる．リモートホストが別の外部ホストであれば実際の運用と同じ状況であるが，本研究では，GK を動かすホストと，コマンドクライアントを同一のホストで実験した．

4.3 通信実験

TCP 途中切替えの実験について述べる．この実験のとき通信が開始される前にあらかじめ Remote Host から TCPCLONE のルールを設定してある．Router(192.168.2.254) にて tcpdump でパケットを dump した結果を示す．

通信開始時

```
IP 192.168.1.1.41307 > 192.168.2.2.60000:
S 2181408229:2181408229(0)
IP 192.168.2.2.60000 > 192.168.1.1.41307:
S 2185966955:2185966955(0) ack 2181408230
IP 192.168.1.1.41307 > 192.168.2.1.60000:
S 2181408229:2181408229(0)
IP 192.168.2.1.60000 > 192.168.1.1.41307:
S 2177928605:2177928605(0) ack 2181408230
```

Attacker から Server へアクセスを開始した時の各コネクションの作成はこのようになる．Attacker から Server と Honey 両方へ通信経路確立のための Syn が送信された．そして CLONEtoSWITCH のルールを加えたときに発生した通信を以下に示す．

Close Connection(Server)

```
IP 192.168.1.1.41307 > 192.168.2.1.60000:
FP
IP 192.168.2.1.60000 > 192.168.1.1.41307:
F 37:37(0) ack
```

Server への通信経路に対し，Flag の Fin が送信された．Honey との通信も切れたときの DUMP 結果を示す．

Close Connection(Honey)

```
IP 192.168.1.1.41307 > 192.168.2.2.60000:
F 49:49(0) ack 49
IP 192.168.2.2.60000 > 192.168.1.1.41307:
F 49:49(0) ack 50
IP 192.168.2.1.60000 > 192.168.1.1.41307:
FP 31:37(6) ack 38
```

Honey との通信経路にも Flag の Fin が送信された．この他に，CLONE，SWITCH，THROUGH の実験が TCP/UDP 両プロトコルにおいて成功している．

5 おわりに

本研究では，GK を GINE ライブラリを用いて再設計した．結果として，既存手法では未実装であった TCP の NAT 通信を実装し，SSL クライアント認証でセキュリティを強化することが可能となった．

参考文献

[1] Ihara, A., Murase, S. and Goto, K.: IPv4/v6 Network Emulator using Divert Socket, *Proc. of 18th International Conference on Systems Engineering(ISE2006)*, Conventry, UK, pp. 159–156 (September .2006).

[2] 中西忠夫，坂口由佳：段階的通信制限システムの拡張 (卒業論文)，南山大学数理情報学部 情報通信学科 (2009).

[3] Viega, J., Messier, M. and Chandra, P.: OpenSSL-暗号化・PKI・SSL/TLS ライブラリの詳細，オーム社 (August 2004).