段階的通信制限システムの拡張

2005MT077 中西 忠夫　　2005MT093 坂口 由佳
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) や迷惑メールなどの迷惑行為が増加している．特に，DoS 攻撃などの不正アクセスについては，掲示板や検索サイトなどがターゲットとして狙われ，その被害について報道されることが多い [3].

本研究ではこの対策手段の一つである「セキュリティのための段階的通信制限システムの評価と改良」(以下，既存手法とする)[2] のシステムを拡張することに重点を置いて進めていく．なお，待ち行列での処理には GINE 論文の Queue 処理の手法 [5] を用いる．また，既存手法のシステムをゲートキーパー (以下，GK) と呼ぶ [1].

拡張部分は，GK にハニーポット (Honeypot)[4] を組み込むことと，フィルタリングルールの設定時の認証である．一般的に言うハニーポットとは，脆弱性を持ったサーバなどをわざとインターネットにさらしておき，攻撃者の手法や侵入者の行動を研究するというものであり，アプリケーションの通信は始めからハニーポットが受ける．しかし，本研究では，正常な通信は普通に処理し，怪しい通信と判断した場合は途中でハニーポットへ切替える．

よって，ハニーポットを GK に組み込むことで侵入者の行動を分析することができ，攻撃の手法を得たり，侵入者を監視しそれらを研究することでよりセキュリティを高めることができる．また，ルールの設定時に認証を加えることでさらに強力なシステムになるであろうと考えた．

なお，実験は共同で行い，中西は主にルールのエントリとフィルタ操作，坂口は主にプログラム作成を担当した．

2　システムの概要

この節では，本研究で提案するシステム (以下，GK3) の概要を既存手法と比較して，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．

2.1　既存手法の構成と処理の流れ

GK は主に，Receiver，Queue，Sender，RealTimeClockTimer(以下，RTCTimer とする)，Filter，FilterManager，RuleUpdator，Commander から成り立っており，**図 1** の NAT を除いた部分を指す．

Receiver は制限対象リストの Filter をチェックし，受信フレームをどの経路に振り分けるか判断する．THROUGH の場合は素通し，DELAY の場合は遅延，THROTTLE(**図 1** では LIMIT を指す) の場合はスループット制限，LOSS の場合は任意の確率でフレームを破棄し，フレーム損失を起こす．各 Queue に入れられたフレームは, 障害を発生させる処理を行った後, RTCTimer と連動した Sender によって送信される．

Commander は外部アプリケーションとの通信，外部アプリケーションからの要求に応じて，フィルタリングルールの追加や削除，表示などの処理を行ったり，外部アプリケーションからの依頼を設定する時の設定内容の確認通知や，依頼された処理が成功したか失敗したかの結果の返信を行う．

また，Commander が外部アプリケーションから受け付けた要求を FilterManager に渡して処理を行う．

2.2　提案するシステムの概要

本研究では，GK にハニーポットを組み込み，通信の始めからと途中からの切替えを行う機能を加え，攻撃のログが取れるようにする．さらに，フィルタリングルールを設定する際に，パスワードによるユーザ認証機能を追加した．また，ルール入力の際の打ち間違いを防ぐため，THROTTLE を LIMIT へ変更した．

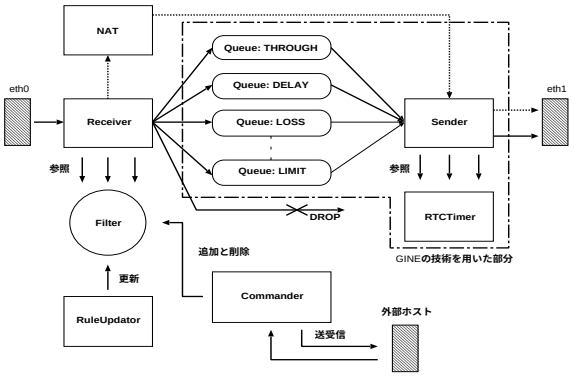

図 1　GK3 の構成

図 1 は GK3 の構成図である．通信を切替えるにあたり，アドレス変換を行わなければならない．その際に必要なハニーポットのアドレスや，切替えの命令は外部アプリケーションから受け付け，アドレス変換を行う．よって，このために既存手法から NAT を追加する．

主な拡張点は，以下の 2 点である．

- アドレス変換
- コマンド受付・返信のスレッド，Filter のルール処理機能の改良

通信を切替えるにあたり，アドレス変換を行わなければならない．その際に必要なハニーポットのアドレスや，切替えの命令は外部アプリケーションから受け付け，アドレス変換を行う．よって，コマンド受付・返信のスレッドを，外部アプリケーションから切替えパターンの入力，変換するアドレスの入力に対応できるように

改良した．

2.3 切替え時の通信の流れ

図 2 は保護ホストとハニーポット両方と通信を行う場合の流れを示したものである．

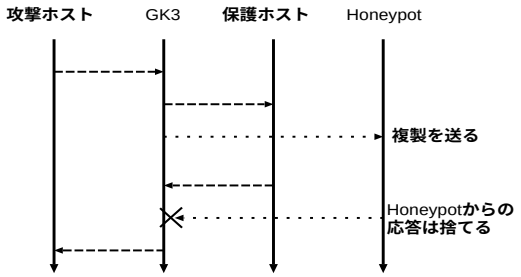

図 2　保護ホストとハニーポット両方との通信の流れ

通信の動きは以下の通りである．

1. 攻撃ホストがパケットを矢印の方向へ送る
2. GK3 が経路情報を参照
3. 保護ホストとハニーポットにパケットが送られる
4. 保護されるホストとハニーポットが GK3 にパケットを送り返す
5. 攻撃ホストにパケットが送られる

3 においてハニーポットへ送られるパケットは，GK3 においてコピーしアドレス変換を行ったパケットである．また，5 において攻撃ホストにはハニーポットが設置されていることは分からないため，ハニーポットからのパケットが攻撃ホストに送られてしまってはならない．よって，4 のハニーポットからのパケットは GK3 が吸収し，保護ホストからのパケットのみを攻撃ホストに送る必要がある．

また，この他の通信として，ハニーポットのみに送られる場合と，保護ホストのみに送られる場合がある．

2.4 TCP を用いた通信の切替え

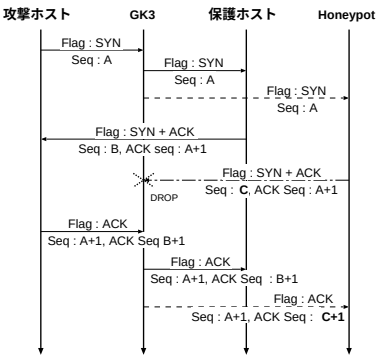

図 3　3 ウェイ・ハンドシェーク

図 3 は，GK3 でハニーポットとも 3 ウェイ・ハンドシェークを可能とする処理内容を表したものである．この時のルールは，保護ホストとハニーポットの両方と送受信を行うルール (以下，ルール A) と，ハニーポットからの返信を削除するルール (以下，ルール B) を用いる．

通信の動きは，以下の通りである．

1. 攻撃ホストが接続要求のパケットを送る．
2. GK3 でルール A の処理を行い，保護ホストとハニーポットにパケットを送る．
3. 保護ホストの確認応答と接続要求のパケットは，そのまま攻撃ホストへ送る．
4. ハニーポットの確認応答と接続要求のパケットは，ルール B によって削除し，シークエンス番号 (Seq : C) を GK3 で値を記憶．
5. 攻撃ホストの確認応答のパケット (保護ホスト宛) は，そのまま保護ホストへ送る．
6. 攻撃ホストの確認応答のパケット (ハニーポット宛) は，GK3 で記憶した値 (C) を用いて ACK 番号を変更．

GK3 では，ハニーポットからのパケットの値を保存することは出来ているが，その後の攻撃ホストからのパケットの値を変更する部分が出来ていない．なぜならば，Receiver が通信の方向別 (仮に保護ホストやハニーポットからの通信を in，攻撃ホスト側からの通信を out とする) にあり，それに伴ってフィルタリングルールが設定されている．そのため，in 方向に設定したルール B と out 方向に設定したルール A は，お互いを参照することが出来ない．したがって，今後 in 方向と out 方向のフィルタリングルールをペアにして，それらのルールがお互いに他を参照できるようにする必要がある．

TCP の通信では，3 ウェイ・ハンドシェーク以外に接続の切断や再送制御の特別な処理がある．GK3 では値の変更が行えず，これらに対応した処理を行えていない．

3 システムの実現

本研究で提案するシステムを実現するためには，アドレス変換の実装，ルール設定時のパスワード認証の成立が必要となる．

3.1 通信経路の切替え方法

既存手法では，コマンド受付・返信スレッドの処理を行う Commander は外部アプリケーションからフィルタリングルールの追加や削除，表示などの処理を行う．本研究では，さらに Commander で経路を切り替えるために必要なコマンドの種類を追加する．

さらに，ルールを追加・挿入する際に，変換するアドレスを入力する欄が加わった．通信経路の切替え指示は，以下の 3 点である．

- N_CLONE：保護ホストとハニーポットの両方と送受信を行う．
- N_DROP：N_CLONE のとき，ハニーポットからの返信を削除する．
- N_SWITCH：ハニーポットのみと送受信を行う．

N_CLONE の場合，まずフレームをコピーし，その

フレームに対してアドレスを変換する．この変換したフレームは，フィルタリングを行わずにハニーポットへ送信される．また，N_DROP を使用して，ハニーポットからの返信を破棄する．

N_SWITCH の場合，フレームはコピーせず，そのままアドレス変換を行う．その変換したフレームは，フィルタリングを行わずに送信する．ただし，ハニーポットのみと送受信を行うためハニーポットのアドレスを外部へ知られてはならないので，ハニーポットから GK3 へ送られるフレームの送信元のアドレスを変えることも必要となる．

3.2 データの変更点

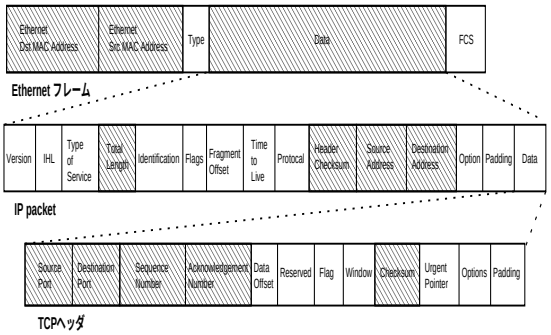

図 4　Ethernet フレームと IP/TCP パケット構造

図 4 は，フレーム，パケット，TCP/UDP のデータ構造の主要情報で，アドレス変換で変更となる箇所 (色付き) を表したものである．

Ethernet フレームでは，送信元 MAC アドレスや宛先 MAC アドレス，データの変更が必要となる．しかし，GK3 では送信元 MAC アドレスと宛先 MAC アドレスの変更を行わない．既存手法は，実験環境でルータを挟まない構成であった．もし多数のハニーポットを使用したい場合，各 PC の MAC アドレスを調べ記録しなければならず，記録ミスなどの人的ミスを起こし兼ねないと考えた．そこで，GK3 と保護ホストやハニーポットの間にルータを挟む実験環境とし，GK3 では送信元 MAC アドレスや宛先 MAC アドレスの変更を行わない．

IP ヘッダは，データ長やチェックサムの再計算，宛先 IP アドレスや送信元 IP アドレスの変更が必要となる．

TCP/UDP ヘッダは，ヘッダの情報や応用層のデータ，チェックサムの再計算が必要となる．

ICMP ヘッダではチェックサム計算に IP アドレスをカバーしないため，疑似ヘッダを含まない．ただし，ICMPv6 では TCP，UDP と同様である．GK3 では，ICMPv6 を扱わないため，ICMP ヘッダの変更は必要ない．

3.3 データ変更処理の動き

プログラムでデータの変更をどのように処理するか，以下に示す．

1. 変更する IP アドレスの情報を得る．
   (通信開始)
2. ルール判定をし，N_CLONE であればコピーしたフレームのデータ，N_SWITCH であればフレームのデータを，アドレス変換スレッドに送る．
3. IPv4 であるか判断し，IPv4 でなければエラー文を表示する．
4. 変更する送信先のアドレスがあるか，変更する送信元のアドレスがあるかどうか確認する．
5. 変更箇所があれば，アドレスを変更する．
6. TCP や UDP のチェックサムの再計算を行う．
7. 最後に，IP パケットのチェックサムの再計算を行う．
8. 変更したフレームは，フィルタリングを通さず，THROUGH と同様の Queue を用いて送信先へ送信する．

3.4 ルール設定時の認証

パスワードはあらかじめ GK3 を扱う者が知っているものとし，プログラム内でのみ変更可能である．ただし，外部からのパスワードを変更する指示は出来ない．フィルタリングルールを依頼する際の流れは以下の通りである．

1. GK3 へ接続し，パスワードを入力する．
2. 再度 GK3 へ接続し，フィルタリングルールを依頼する．

パスワードを入力後パスワードが合っている場合，接続者と GK3 へ認証が成功されたことを表示する．パスワードが間違っている場合は，接続者と GK3 へエラー文を表示させる．

4 実験結果

この節では実際に GK3 を動かして実験を行い，その結果を述べる．

4.1 実験ネットワークの構成

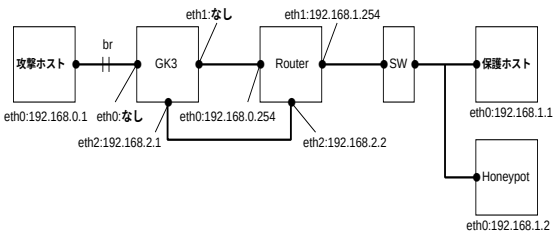

図 5　実験環境

図 5 は実験環境について示した図である．実験環境は Vine Linux 4.2 をインストールした PC を 4 台用意し，それぞれを攻撃ホスト，ルータ (Router)，保護ホスト，ハニーポットと見立て，さらに GK3 をインストールした PC(Vine Linux 4.2) を用意し，LAN ケーブルを繋いで構築した．攻撃ホストと保護ホストの NIC は 100M

である．

4.2 パスワード認証

ルール設定を依頼すると，パスワードが必要であると表示され接続が強制的に遮断された．その後，設定されているパスワードを入力をし，再度依頼すると正しい結果が得られた．よって，パスワードの入力を行わないとルール設定が出来ないことの確認ができた．

4.3 GK3 の動作確認

変換するアドレスの入力と，通信経路の切替えが正しく反映できているかの確認を行う．確認を行うのは，通信の始めからルールを設定する場合と，通信の途中からルールを設定する場合である．また，ICMP では Linux コマンドの ping，UDP・TCP では iperf を用いた．

始めから切替えの場合

N_SWITCH のルールを設定して確認を行った．ICMP・UDP・TCP の全てにおいて，攻撃ホスト側の表示が保護ホストと通信を行っているとなっていたが，tcpdump などで調べるとハニーポットとの通信となっていた．これは，宛先と送信元が変換されていたので，N_SWITCH のルールが正しく設定されていることが言えた結果である．

途中から切替えの場合

通信の途中から切替えは，N_CLONE・N_DROP の 2 つのルールから N_SWITCH にルール変更することである．攻撃ホストとの通信を途切れないようにするために，N_CLONE，N_DROP の順番にルールを削除し，その後 N_SWITCH のルールを設定していく．

ICMP・UDP とも以下のような結果となった．

- 攻撃ホストでは保護ホストと通信をしている表示となっており，通信が途切れることはなかった．
- tcpdump の表示より
  - 最初のルール設定では保護ホストとハニーポット両方との通信となった．
  - N_CLONE のルールの削除，そして 1 つ目の N_SWITCH のルールが追加された時は保護ホストのみとの通信となった．
  - 2 つ目の N_SWITCH のルールが設定されたらハニーポットのみとの通信となった．

これは N_CLONE，N_DROP，N_SWITCH のルールが正しく動作していること言え，さらに通信が途切れなかったことにより，ルールの変更の順序も正しかったことが言える．以上より，通信の途中で，保護ホストからハニーポットに切替えることに成功した．

4.4 応答時間と速さの検証

表 1 は ICMP の応答時間，そして UDP・TCP それぞれの速さの測定値を示したものである．ルールを設定していない場合を THROUGH，始めからの切替えの場合を N_SWITCH，途中からの切替えの場合を N_CLONE としている．

結果として，全ての場合においてルールを何も設定していない場合と，多少の誤差があるが値が近い．したがって，切替えを行っても影響がないことが言えた．

表 1　ICMP の応答時間，UDP・TCP の速さの測定値

	THROUGH	N_SWITCH	N_CLONE
ICMP(msec)	0.83	0.81	0.95
UDP(Mbps)	93.14	95.04	94.98
TCP(Mbps)	91.58	92.98	-

5 おわりに

本研究は，GK にハニーポットを組み込み，外部アプリケーションから切替えの要求に対応することが可能となった．また，パスワードを用いたルール設定の認証を行う．

そして，切替えの動作確認を行い，通信の始めからの切替えと途中からの切替えの 2 パターンの通信に成功した．よって，攻撃者からハニーポットの存在を知られず，攻撃を分析することが可能となった．

これらより，保護したいホストを守りながら攻撃の手法を得ることができ，ルール設定時の安全性が高まったと考えられる．

今後の課題として，以下の 4 つが挙げられる．

- TCP を用いた通信の切替えの成立
- 新たな攻撃を用いた性能評価
- 実ネットワークでの実験
- フィルタリングルール設定時の認証の改良

参考文献

[1] Aoyama, M., Kojima, M., Goto, K.: Design and Implementation of a Traffic Limiter for Network Security, *Proc. of 16th International Conference on Systems Science, Vol.II*, Poland, pp. 213–220 (2007).

[2] 福井麻美，末吉昭仁：セキュリティのための段階的通信制限システムの評価と改良，卒業論文，南山大学 数理情報学部 情報通信学科 (2008).

[3] 警察庁セキュリティポータルサイト@police：我が国におけるインターネット治安情勢の分析について (平成 20 年度第 1/四半期) (2008). (http://www.cyberpolice.go.jp/detect/pdf/080723.pdf).

[4] Spitzner, L.: ハニーポット-ネットワーク・セキュリティのおとりシステム-, 慶応義塾大学出版会 (2004).

[5] Sugiyama, Y. and Goto, K.(Eds. Zhang, S. e. a.: Design and Implementation of a Network Emulator using Virtual Network Stack, *Proc. of the Seventh International Symposium on Operations Research and Its Applications (ISORA2008), Lecture Notes in Operations Research, Vol.8*, World Publishing Corporaiton, pp. 351–358 (2008).