社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2005－DSM－38 (11)
2005／8／5

ポリシルーティングを用いたネットワークハニーポットの構築

白畑 真[†], 南 政樹[†], 村井 純[†‡]

[†]慶應義塾大学 政策・メディア研究科
[‡]慶應義塾大学 環境情報学部
{true,minami,jun}@sfc.wide.ad.jp

これまで，未使用のIPアドレス空間を用いた未知トラフィック観測手法が提案されてきた．しかし，これらの手法は受動的であるため，未知のワームの観測に適用するには限界があった．そこで，低インタラクション型ハニーポットを利用することで，ワーム等の活動に伴って発生する対話的なプロトコルを用いた攻撃トラフィックを捕捉できた．この際，ハニーポットが対応できるサービスは，その実装により限定される．この問題を解決するため，複数のサーバにおいて，それぞれ異なる実装を同時に稼働させ，ポリシルーティングを用いて，トラフィックを適切に誘導することにより，多種のサービスを同時に運用可能なシステムを構築した．

Development of Network-based Honeypot Using Policy Routing

Shin SHIRAHATA[†], Masaki MINAMI[†], Jun MURAI[†‡]
[†]Graduate School of Media and Governance, Keio University
[‡]Faculty of Environmental Information, Keio University
{true,minami,jun}@sfc.wide.ad.jp

Up to now, there are many unknown traffic monitoring techniques have been proposed. However those methods have operating limit in monitoring unknown worms, because these techniques are passive. Therefore, operators can monitor attacking traffic, which uses interactive protocols by using low interaction honeypot. In this case, services which honeypot can handle are limited by their implementations. To solve the problem, we have developed multi service capable network-based honeypot system, which runs multiple implementations on several servers and using policy routing to direct traffic.

1. 背景

ハニーポットは，攻撃行為，もしくは攻撃試行行為を捕捉するために，脆弱性が存在する，あるいは存在するかのように振る舞うシステムである．ハニーポットは，“高インタラクション型”ハニーポットと“低インタラクション型”ハニーポットの二種類に分類される．高インタラクション型ハニーポットでは，脆弱性のあるシステムを構築し，攻撃者に侵入させる．一方，低インタラクション型ハニーポットでは，脆弱性のあるサービスをエミュレートのみを行うため，実際にホストへの侵入は発生しない．

理論的には，ホストが存在しないIPアドレス空間に対しては，トラフィックが発生しないと考えられる．しかし現実には，ホストが存在しないIPアドレス空間に対しても，(D)DoS攻撃などのソースIPアドレスを詐称したパケットに対して発生するBackscatter[1]や，ワームや攻撃者による攻撃先探索活動に伴うトラフィックが観測されている．

これまで，CAIDAのNetwork Telescope[2]や，The Team Cymru Darknet Project[3]などにおいて，未使用のアドレス空間に対して到達したパケットを観測する研究が行われてきた．この手法は，一般にDarknetと呼ばれ，さまざまなネットワークにおいて運用されている．

典型的なDarknetの構成例を図1に示す．この構成においては，外部のネットワークに対して10.0.0.0/8のような集約された経路をEGPで広報していると仮定している．しかし，実際にIGPに

おいて広報されている経路は，10.1.0.0/16 と 10.2.0.0/16 のみであるため，その他の 10.0.0.0/8 内のアドレスブロック宛のパケットは，最長一致規則により，計測サーバにルーティングされる．

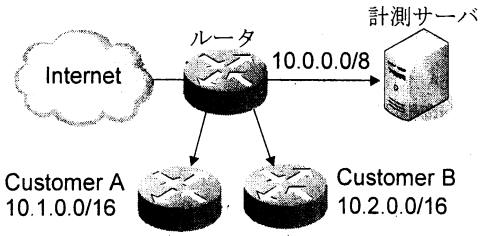

図 1:Darknet の構成例

典型的な Darknet においては，計測サーバにおいて入手できるトラフィックは，セッションを確立することなく受信できるパケットに限定される．このため，TCP のようなセッションベースのプロトコルにおいて，同一のポート番号に対して異なる脆弱性が存在する場合には，攻撃の詳細を調査することが難しい．

さらに，どの脆弱性を狙った攻撃かを判断するには，アプリケーション層のペイロードを分析する必要がある．特に Windows RPC には，過去に MS03-026[5]や MS03-039[6], MS04-011[7]といった複数の脆弱性が存在したため，ポート番号のみでは，攻撃の内容を判断できない．

このような問題を解決するため，TCP セッションをエミュレートするアプローチがある．今回は，未使用の IP アドレス空間を低インタラクション型のハニーポットシステムにルーティングすることで，特定のペイロードを含めた調査を行った．

また，The Internet Motion Sensor[4]では，従来の Darknet に加えて，実質的な低インタラクション型ハニーポットである“lightweight active responder”を用い，TCP コネクションを確立する方法を併用している．

2. 低インタラクション型ハニーポットの概要と課題

低インタラクション型ハニーポットには，アプリケーション層で動作し，特定のサービス実装をエミュレートするタイプと，トランスポート層で動作し，TCP セッションをエミュレートするタイプに分類できる．

アプリケーション層で動作する低インタラクション型ハニーポットには，オープンソースの honeyd[8] や mwcollect[9]，商用製品の KFSensor[10], Specter[11]などがある．一方，トランスポート層で動作するタイプの低インタラクション型ハニーポットには，dumnet[12]がある．

アプリケーション層で動作するハニーポットは，特定のサービス実装をエミュレートするため，アプリケーション層のプロトコルにおける要求に応答できる．例えば mwcollect は，脆弱な Windows RPC サービスをエミュレートし，ワームによる攻撃を捕捉できる．このようにアプリケーション層で動作を行うハニーポットは，特定のサービスに特化した動作ができる．

トランスポート層で動作するハニーポットは，TCP のコネクションをエミュレートする．従って，特定のサービスに依存することなく，全ての TCP ポートに対するアクセスを捕捉できる．例えば，dumnet は，全ての SYN オプションが有効な TCP データグラムに対して，SYN+ACK を返答するため，結果的に全ての TCP ポートが開かれているかのように動作する．図 2 に TCP の Three-way Handshake の流れを示す．このため，dumnet は OS の TCP/IP スタックを利用せず，libpcap を用いて独自に TCP セッションの動作をエミュレートしている．

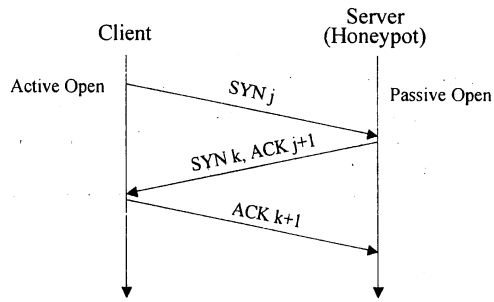

図 2: Three-way Handshake

3. システムの設計

低インタラクション型ハニーポットにおいては，攻撃に利用される可能性のある脆弱性ごとにサービスをエミュレーションすることが理想である．しかし現実には，ハニーポットの実装によりエミュレーションできるサービスが限定されるため，幅広いサービスをエミュレーションできない．

本研究では，これらの異なる特性を持った低インタラクション型ハニーポットを組み合わせることで，“ネットワークハニーポット”を構築した．ネットワークハーポットは，複数のハニーポットから構成され，異なるサービスを同時に提供するシステムである．

アプリケーション層で動作する低インタラクション型ハニーポットは，脆弱性などの特性を持った特定のソフトウェア実装の動作をエミュレートする．従って，エミュレーションの対象ではないソフトウェア実装や，プロトコルには対応できない．

一方，トランスポート層で動作するハニーポットは，アプリケーション層のプロトコルを処理できない．

本研究においては，あるハニーポット実装が，アプリケーション層のサービスをエミュレーションできる場合，このハニーポットを，特定サービスに対する“優先ハニーポット”と呼ぶ．また，トランスポート層で動作するハニーポットのように，特定のサービスに依存せず，TCP コネクションをエミュレートできるハニーポットを”デフォルトハニーポット”と呼ぶ．

また，ハニーポット実装が稼働するプラットホームにも制約が存在する．たとえば，mwcollect と dumnet は UNIX 系 OS で動作するが，KFSensor は Windows で動作するため，異なるホストで稼働させる必要がある．本設計では，単一のハニーポット実装は独立したホストで稼働すると仮定する．

これらの問題を解決するため，ポリシルータを導入し，ハニーポットを構築している IP アドレス空間に対して，宛先ポート番号ベースのポリシルーティングを適用することで，特定のポート番号に対するアクセスを，特定のハニーポットに転送できる．

4. 実装

今回は，"優先ハニーポット”として mwcollect を用い，mwcollect がエミュレーション可能なサービスに対するトラフィックを mwcollect が稼働するホストにルーティングする．その他のエミュレーションできないサービスのトラフィックは "デフォルトハニーポット”である dumnet が稼働するホストにルーティングする．

表 1 に実装環境を示す．

OS		Debian GNU/Linux 3.1 (Sarge), Linux Kernel 2.4.27
ポリシルーティング	パケット判別・マーキング	iptables v1.2.11
	ルーティング	iproute2-ss041019

表 1: 実装環境

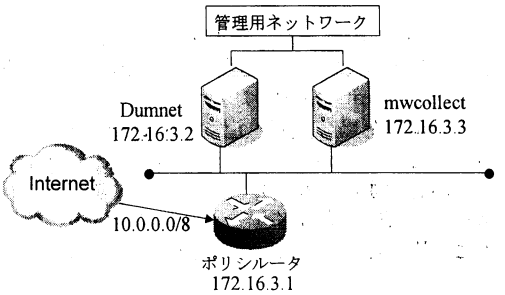

図 3: ネットワーク構成図

また，図 3 にネットワーク構成図を示す．この構成図においては，10.0.0.0/8 をハニーポットで利用するアドレス空間と仮定し，デフォルトハニーポットを 172.16.3.2，優先ハニーポットを 172.16.3.3 としている．

以下に，ポリシルータ上の iptables において，2745/tcp, 2556/tcp, 8866/tcp, 4751/tcp, 6777/tcp, 11117/tcp, 81/tcp, 135/tcp, 445/tcp, 1025/tcp, 42/tcp のパケットに対して，パケットマーキングを行う設定を示す．なお，表記上は改行されているが，実際には改行しない箇所を←記号で示す．

```
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 2745 -j MARK -set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 2556 -j MARK -set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 8866 -j MARK --set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 4751 -j MARK --set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 6777 -j MARK --set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 11117 -j MARK --set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 81 -j MARK --set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 135 -j MARK --set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
```

```
 -m tcp --dport 445 -j MARK --set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 1025 -j MARK --set-mark 10
iptables -t mangle -A POSTROUTING -p tcp ←
 -m tcp --dport 42 -j MARK --set-mark 10
```

また，iproute2 によるルーティングの設定を以下に示す．

```
echo 10 mwcollect ←
 >> /etc/iproute2/rt_tables
ip route add default via 172.16.3.3 ←
  table mwcollect
ip rule add fwmark 10 table mwcollect
ip route flush cache
```

次に，同一のサービスを実装しているハニーポットが複数種類存在する場合の動作について述べる．同一のポート番号に対するハニーポットを複数設定した場合，経路上はすべてが優先ハニーポットとなるため，Equal Cost Multi Path と同様の処理が発生する．本実装においては，優先ハニーポットが複数存在する場合、そのなかからランダムに利用するハニーポットを決定している．

5. 今後の課題

現在のシステムにおいては，ポート番号を元にポリシルーティングを行っている．しかし，UDP を用いたワームや，TCP セッションを確立した直後から攻撃に伴うペイロードを送信するワームなどによるトラフィックを処理するために，コンテンツベースのルーティングを考慮する必要がある．

今回は，mwcollect がハニーポット側で Windows RPC の複数の脆弱性をエミュレートできたが，今後は特定の脆弱性に対する攻撃行為のペイロードを特定のハニーポットにルーティングできることが望ましい．今後は，ハニーポットに対してアプリケーション層を含めたポリシルーティングを適用することで、攻撃の種類に応じたハニーポットへのルーティングを実現していきたい．

6. まとめ

本研究では，ポリシルーティングを用いることにより，トラフィックの内容に応じて，サービスのエミュレーションが可能なハニーポットに誘導した．これにより，特定のハニーポット実装の稼働環境や，ハニーポットが提供できるサービスの種類に依存することなく，多様な攻撃を適切なハニーポットで処理できた．

7. 参考文献

[1] D. Moore, G. Voelker and S. Savage. "Inferring Internet Denial-of-Service Activity". In *Proceedings of the 10th USENIX Security Symposium*, Augest 2001.

[2] D. Moore. "Network Telescopes: Observing Small or Distant Security Events". *Proceedings of the 11th USENIX Security Symposium,* Aug. 2002

[3] Team Cymru, "The Team Cymru Darknet Project", http://www.cymru.com/Darknet/

[4] Michael Bailey, Evan Cooke, Farnam Jahanian, Jose Nazario and David Watson* "The Internet Motion Sensor: A Distributed Blackhole Monitoring System", *In Proceedings of Network and Distributed System Security Symposium, ISOC*, February 2005

[5] Microsoft Corporation, "RPC インターフェイスのバッファ オーバーランによりコードが実行される (823980) (MS03-026)" July 2003, http://www.microsoft.com/japan/technet/security/bulletin/MS03-026.mspx

[6] Microsoft Corporation, "RPCSS サービスのバッファ オーバーランによりコードが実行される (824146) (MS03-039) ",September 2003, http://www.microsoft.com/japan/technet/security/bulletin/MS03-039.mspx

[7] Microsoft Corporation, "Microsoft Windows のセキュリティ修正プログラム (835732) (MS04-011)",http://www.microsoft.com/japan/technet/security/bulletin/MS04-011.mspx

[8] Niels Provos. "Honeyd: A virtual honeypot daemon (extended abstract)". *In 10th DFN-CERT Workshop*, February 2003.

[9] Felix C. Freiling, Thorsten Holz, and Georg Wicherski. "Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks". *AIB Technical Report,* 2005-07, April 2005

[10] KeyFocus, "KFSensor", http://www.keyfocus.net/kfsensor/

[11] NETSEC. "Specter", http://www.specter.com/

[12] Junichi Murakami, "Dumnet" http://tf.rootkit.jp/work/dumnet/