L-023

受動的なOS特定法にみる，通信サービス品質改善の可能性に関する一考察
Consideration concerning possibility of improving communication service quality based on passive fingerprinting

山口 榮作 † 鈴木 常彦 ‡ 長谷川 明生 §
Eisaku Yamaguchi Tsunehiko Suzuki Akiumi Hasegawa

1. まえがき

ネットワークを介してサービスを提供することは，非常に利便性の高い環境を提供すると同時に，常にサービスの品質を維持する上では厄介な問題を抱え持たねばならない．それは，サービスとして提供する情報やシステムの完全性を維持することであったり，そこで受け付ける情報の機密性であったりする．

TCP/IP の規格は RFC で定義されているが，OS におけるプロトコルスタックの実装には差異があり，この差異は OS fingerprint と呼ばれている．あるホストから送信される TCP パケットを解析すると，その OS 固有の実装に依存した情報から，OS を推定することが可能である．

TCP 通信において，パケットから OS fingerprint を得るためには，二通りの方法がある．一つは，QueSO(Que Sistema Operativo)[1],[2] や nmap[3],[2] のように自ら TCP パケットを送信し，相手からの TCP パケットを誘い，観察することで OS fingerprint を得る方法である．これを Active fingerprinting[4] と呼ぶ．もう一つは，相手からの TCP パケットを待ち，到着した TCP パケットをサービスで処理するのと並行して，解析することで OS fingerprint を得る方法である．これを Passive fingerprinting[5] と呼ぶ．Active fingerprinting では，相手がスキャンされていることを知る手がかりを残すことになるのと対照的に，Passive fingerprinting では，相手にスキャンされていると知られることなく，相手の環境の情報を収集することができる．また，多くのホストの中からこちらのホストにアクセスしようとしてくるホストに限定して情報を収集することができる．

サービスを提供する側は，通信相手の状況に応じてサービスの品質管理を行うことができると，より高度な通信制御ができる．単純な例では，MTA(Mail Transfer Agent) の多くは，UNIX 系 OS で構築されていることから，MTA では，昨今の botnet[6] の温床となっている Windows 系機器からの SMTP アクセスには，一時拒否，tarpiting などの spam 対策手順を強化することなどが考えられるし，IDS(Intrusion Detection System)/IDP(Intrusion Prevention System) においては，パケットの特徴と OS との関係から，検知精度を高めている．

本論文では，サーバのみならずネットワーク管理装置でも有効活用が可能である Passive fingerprinting を中心に，OS fingerprint 利用の効果を検討する．

†愛知県立大学 情報化学部 情報システム学科
‡中京大学 情報理工学部 情報システム工学科
§中京大学 生命システム工学部 身体システム工学科

2. OS の検出

Passive fingerprinting は決して新しい技術ではない．古くは tcpdump[7]/snoop のようなパケットキャプチャプログラムと併用することで実現していたものもあるであろうが，昨今では，`p0f`[8], `pf`[9], `Siphon`[10] など幾つかのパッケージが存在している．

本研究では，継続的に開発が続けられており，OS fingerprint の Database の構築プロジェクトなども並存していることなどから，`p0f` を採用し，FreeBSD-6.2-RELEASE 上で OS fingerprint を確認した．

OS fingerprint の取得確認のために，Solaris 9 SPARC, Solaris 2.6 SPARC, NetBSD-4 BETA2, NetBSD-3.1, NetBSD-3.0.1, NetBSD-2.0.2, NetBSD-1.5.3, FreeBSD-6.1, FreeBSD-5.5, OpenBSD-3.0, Windows XP SP2 Professional, Windows XP SP2 Home といった，複数種の OS 環境を使用した．

`p0f` による OS fingerprint の検出確認結果は表 1 のようになった．

表 1: OS fingerprint の検出確認結果

OS	Detected OS	d¶	link
Solaris 9 SPARC	Solaris 9	16	e/m‖
Solaris 2.6 SPARC	Solaris 2.5-7(2)	16	e/m‖
NetBSD-4 BETA2	NetBSD 3.0(DF)	16	e/m‖
NetBSD-3.1	NetBSD 3.0(DF)	16	e/m‖
NetBSD-2.0.2	NetBSD 1.6Z or 2.0 (DF)	13	e/m‖
NetBSD-1.5.3	NetBSD 1.3	13	e/m‖
FreeBSD-6.1	FreeBSD 6.x (1)	0	e/m‖
FreeBSD-5.5	FreeBSD 5.3-5.4	23	e/m‖
FreeBSD-4.11	FreeBSD 4.6-4.9	0	e/m‖
OpenBSD-3.0	OpenBSD 3.0-3.9	16	e/m‖
Windows XP SP2 (P)	Windows 2000 SP4, XP SP1+	16	e/m‖
Windows XP SP2 (H)	Windows 2000 SP2+, XP SP1+ (seldom 98)	15	e/m‖
Windows 2000 SP4	Windows 2000 SP4, XP SP1+	15	e/m‖

概ね種類は合っており，バージョン等についての相違が見られるものの，`p0f` にはある程度の判定能力がある事が裏づけられる．但し，実運用サーバ環境において，SMTP セッションを張ってくる相手の IP アドレスと OS fingerprint とをチェックしてみると，特定 IP アドレスでの OS 種の変動，バージョン等の変動も確認されており，信頼の確度については，より多くのサンプル等による検証が必要である．

但し，SMTP セッションにおいて，意図的に SYN に応答せず再送が発生する状況を作ったところ，表 2 のよ

¶distance
‖ethernet/modem

うに，再送部分を決まったパターンで誤検出することも確認されている．SYN に答えないことによる応答パターンで通信相手を識別する手法 [11] など，不自然な受け答えに対する反応等についても，サービス品質制御の道具として検討した経験から，OS fingerprint の Database や判定手法に関して，再送パケットへの対応など，改善の余地があると言える．

表 2: SYN パケット再送時の OS fingerprint の誤検出

OS	Detected OS(missmatch)
NetBSD-2.0.2	FreeBSD 4.8-5.1 (or MacOS X 10.2-10.3) (up: 0 hrs)
FreeBSD-6.1	UNKNOWN [65535:64:1:48:M1460,S,E:P:?:?]
FreeBSD-5.5	Windows 2003 (2)
FreeBSD-4.11	FreeBSD 4.6-4.8 (RFC1323-)

3. OS fingerprint 偽装手法

p0f のレコードは，fingerprint 検出の際の指標となる “TCP Windows size”, “Initial TTL”, “Don’t fragment bit”, “Overall SYN packet size”, “Option Value”, “Quirks list” の 6 種類のテーブルと，対応する “OS の種類”, “OS の説明” から構成される．したがって，これらの指標を偽装できれば，OS の判定を霍乱することができる．

p0f に対する TCP パケットの偽装試験には，hping[12] を用いた．hping は，IP パケットを RAW フォーマットで送信したり，特定パラメータのみを修正して送信することも可能である．カーネルスペースへの埋め込みや LKM(loadable kernel module) 等の準備は必要なく，管理者権限を持つユーザスペースで動作するプログラムで多くの変更を実現できる．

例えば次のように FreeBSD-4.11 から，destination port = 80, Type Of Service = 0x10, Set don’t fragment flag, Set syn flag といったパラメータを送ることで，送信元 OS は Databese にない未知のものとして判定される．

```
$ hping p0f.example.jp --destport 80 --tos 10
--dontfrag --syn
```

4. 考察

カーネルスペースでなく，ユーザスペースのプログラムで OS fingerprint とは関係ない fingerprint のパケットを排出することは十分に可能であり，単純な Passive fingerprint に頼ったホストを霍乱したり，パラメータをチューニングすることで既存のツールのみで OS を偽装することができる可能性もある．hping は単純なパケット排出プログラムの一例であるが，専用のプログラムの実現は決して難しくはない．

また，timeout による再送を誘発した場合に，SYN パケットの構成が変化することは，p0f と tcpdump により確認でき，より確実に相手の OS を特定するためには，Active fingerprinting 等の併用が望ましいと言える．

Active fingerprinting では，確認者が能動的に検体に対してパケットを送出し，その応答により判断することになる．検体でパケットを受けるのはカーネルレベルとなり，Active fingerprinting に対する偽装や霍乱は，カーネルレベルでの改造を必要とする．

5. おわりに

本研究では，Passive fingerprinting により得られる情報について，一部の検体を通じて解析したものの，統計的な解析には至っていない．既に半年以上に渡って，MTA への SMTP セッションを p0f で蓄積してきており，個々のセッションの OS fingerprinting と spam 判定結果等との突き合わせを考えている．

また，spam 対策における Greylisting が容易に超えられるハードルであるように，Passive OS fingerprint が容易に偽装可能な情報であっても，インターネットで当たり前のように利用されるまでは，攻撃者は偽装をしない可能性もある．Passive OS fingerprint を含めた情報を，今後のネットワーク越しの攻撃対策の材料として検討したいと考えている．

参考文献

[1] Jordi Murgo : Els Apostols, http://web.archive.org/web/19991004032416/http://apostols.org/projectz/queso/ (1998).

[2] Toby Miller : Intrusion Detection Level Analysis of Nmap and Queso, http://www.securityfocus.com/infocus/1225 (2000).

[3] Fyodor : Free Security Scanner For Network Exploration & Security Audits, http://www.insecure.org/nmap/

[4] Ofir Arkin : Identifying ICMP Hackery Tools Used In The Wild Today, December 4, 2000, http://www.sys-security.com/archive/securityfocus/icmptools.html (2000)

[5] Honeynet Project : Passive Fingerprinting, http://project.honeynet.org/papers/finger/

[6] Paul Bächer, Thorsten Holz, Markus Kötter, Georg Wicherski : Tracking Botnets, http://www.honeynet.org/papers/bots/

[7] TCPDUMP public repository, http://www.tcpdump.org/

[8] Michal Zalewski : the new p0f, http://lcamtuf.coredump.cx/p0f.shtml

[9] Daniel Hartmeier, OpenBSD team : The Open BSD Packet Filter, http://www.openbsd.org/faq/pf/

[10] Subterrain Security Group : The Passive Network Mapping Tool, http://siphon.datanerds.net/

[11] 山口榮作, 鈴木常彦 : TCP Handshake 制御を利用した spam 対策システム, 大学情報システム環境研究, **Vol.8**, pp. 60–68 (2005).

[12] Salvatore Sanfilippo : http://www.hping.org/