Identificação Passiva de Sistemas Operacionais

DANIEL B. CID

Projeto Honeypot BR
Rio de Janeiro(RJ) - Brasil
daniel@underlinux.com.br
http://www.honeypot.com.br
Maio 2003

Resumo : *Muito se tem falado sobre "Passive Fingerprinting" e "Active FingerPrinting", mas nada na língua portuguesa. Esse artigo visa explicar como funciona o "Passive FingerPrinting" (Identificação Passiva de Sistemas Operacionais), que é uma excelente técnica de identificação de sistemas operacionais remotos, completamente indetectável , diferente do "Active Fingerprinting" que "qualquer" IDS detecta.*

Abstract : *This article tries to supply a big lack existent in the our language. A lack for good security documentation. In this article i will introduce the reader to the "OS Passive FingerPrinting" technic. You will learn what it is, how it works and you will see some examples of its implementation using the p0f program.*

Palavras Chaves: detecção passiva de sistemas, honeypots**,**segurança,.

FingerPrinting

O "OS FingerPrint" (Operation System FingerPrint), como o próprio nome já diz, se refere a impressão digital do sistema operacional. Ele se baseia no fato de que todos os sistemas operacionais, assim como as pessoas, têm características únicas, que podem ser usadas para identificá-los. Assim como os seres humanos podem ser identificados pela íris, pelo "polegar" ou pela face, os sistemas operacionais podem ser identificados por algumas peculiaridades presentes no cabeçalho dos pacotes transmitidos por eles. Para detectar essas diferenças, alguns métodos foram criados, como o "Active FingerPrinting", no qual se envia vários pacotes mal formados e se analisa a resposta fornecida pelo sistema. Geralmente, cada sistema operacional responde de uma maneira diferente a esses pacotes. Essa técnica é utilizada pelo (http://www.insecure.org/nmap) NMAP. O problema do "Active FingerPrinting" é que ele pode ser facilmente detectado por algum IDS.

O outro método existente é o "Passive FingerPrinting" que, apesar de ter o mesmo conceito do método ativo (Active FingerPrinting), o implementa de uma maneira diferente. O "Passive" utiliza logs obtidos por um sniffer, como tcpdump, separando as conexões originadas do host desejado e verificando as peculiaridades presentes nos cabeçalhos dos pacotes. O "Passive FingerPrinting" tem como maior vantagem ser indetectável, pois ele se baseia em conexões normais, como um acesso ao HTTP ou SMTP. A sua principal desvantagem é que ele não é 100% confiável.

Cabeçalho TCP/IP

Veremos agora alguns pontos interessantes presentes no cabeçalho TCP/IP que podem nos ajudar na tarefa de detecção das pecualiridades dos sistemas operacionais.

Window Size - Esse campo nos diz a quantidade de dados que o computador pode receber. Esse número varia entre cada sistema operacional.

TCP Options - Esse campo é o mais importante para o "Passive FingerPrinting". Ele pode praticamente sozinho identificar um sistema operacional. Algumas opções utilizadas são:

+ MSS (Maximum Segment Size)
+ TimeStamp
+ wscale (Window Scale)
+ SackOK
+ Nop

Você não precisa saber qual a função dessas opções para entender esse nosso estudo. Apenas precisa saber quais sistemas as usam ou não.

Cabeçalho IP

TOS (Type of Service) - Este é utilizado pelo sistema operacional para identificar os pacotes que requerem algum tratamento especial. A maioria dos sistemas o deixa como “00” (normal). Ele pode vir setado das seguintes maneiras:

+ 1000 - Minimizar o "delay"
+ 0100 - Maximizar o "rendimento".
+ 0010 - Maximizar a confiabilidade.
+ 0001 - Minimizar o custo.
+ 0000 - Uso normal

TTL (Time To Live) - Este nos diz quanto tempo o pacote vai estar circulando. Ele é decrementado em cada hop (router) por qual passa. Ele é muito importante para o nosso estudo porque cada sistema operacional seta o TTL inicial de uma forma diferente.

IP ID - Esse campo serve para identificar cada pacote. Dependendo do sistema operacional ele pode ser randômico ou se incrementar de um em um.

Packet Length - Esse campo nos diz o tamanho total do pacote.

Indiosincrasias de cada sistema operacional

Cada sistema operacional “tem” algumas características únicas que tornam possível nossa detecção. Veremos abaixo algumas peculiaridades presentes no Linux, no OpenBSD e no
Windows 2000. Uma lista completa pode ser vista nesse link: http://opensolutions.com.br/docs/passfing/db-passive.php

Linux

IP TTL (2.2 e 2.4) - Geralmente o valor é 64. Você pode alterá-lo em /proc/sys/net/ipv4/ip_default_ttl.

IP ID - O linux cria um ID randômico a cada seção, e esse se incrementa de um em um durante a conexão.

TCP Options - MSS, sackOK, timestamp, nop e wscale setados.

IP Packet Length - O linux seta o tamanho total do pacote como 60 bytes (único SO que faz isso).

TCP Window size - O linux seta o valor inicial como 5840.

OpenBSD

IP TTL - Geralmente o valor é 64.

IP ID - Completamente randômico.

TCP Options - MSS, sackOK, timestamp, nop (5x) e wscale setados.

IP Packet Length - O OpenBSD seta o tamanho total do pacote como 64 bytes.

TCP Window size - O OpenBSD seta o valor inicial como 16384.

Windows 2000

IP TTL - Geralmente o valor é 128.

IP ID - Incrementa de um em um o tempo todo.

TCP Options - MSS, sackOK e nop (2x) setados.

IP Packet Length - O Windows 2000 seta o tamanho total do pacote como 48 bytes.

TCP Window size - O Windows 2000 seta o valor inicial como 16384.

Analisando alguns pacotes

Vamos analisar alguns pacotes, tirados do tcpdump, para facilitar o entendimento.

```
15:07:14.028664 eth0 < enigma.opensolutions.com.br.48723 >
study.logictree.com.www: S
1659907010:1659907010(0)
win 16384 <mss 1460,nop,nop,sackOK,nop,wscale 0,nop,nop,timestamp 178834575 0>
(DF)
```

Nesse primeiro, está vindo uma conexão da máquina enigma.opensolutions.com.br para a porta 80 da máquina study.logictree.com. Nós queremos descobrir qual o sistema operacional da enigma.
Primeiro vamos separar alguns dados importantes:

TCP Options (setadas) - MSS, nop(5x), sackOK, timestamp e wscale
Window Size - 16384

Só com isso já podemos determinar que na máquina está rodando um OpenBSD, pois este SO é o único que seta o campo TCP Options desse modo (com 5 “nop”, MSS, SackOK, timestamp e wscale setados).

```
15:31:15.510446 0:1:2:4b:86:6b 0:a0:cc:73:b2:7f ip 74: 192.168.1.106.39378 >
192.168.1.2.http: S
```

```
[tcp sum ok] 3803196851:3803196851(0) win 5840  (DF)
(ttl 64, id 26489, len 60)
```

Nesse outro exemplo, está saindo uma conexão da máquina 192.168.1.106 para a 192.168.1.2 na porta 80. Ela tem o campo "Window Size" de tamanho 5840, está com o "MSS, SackOK, timestamp, nop e wscale" setados no campo TCP Options, o tamanho do pacote é 60 bytes (len 60) e o TTL é 64. Essa máquina (192.168.1.106) só pode estar rodando um linux (ver informações sobre o Linux acima).

Utilizando o p0f (passive OS fingerprinting tool)

O p0f (http://www.stearns.org/p0f/) é um programa que utiliza essas técnicas estudadas e tenta descobrir o sistema operacional do sistema desejado. A intalação dele é muito simples assim com o seu uso. Você só precisa baixá -lo e compilá-lo: (http://www.stearns.org/p0f/devel/p0f-1.8.2.2.tar.gz)

```
[daniel@sec /home/daniel]$ tar -zxvf p0f-1.8.2.2.tar.gz
[daniel@sec /home/daniel]$ cd p0f-1.8.2.2
[daniel@sec /home/daniel]$ make
```

Para utilizá-lo você precisa de algum log (tcpdump ou do snort por exemplo). Eu irei pegar o log do Scan 28 (http://www.honeynet.org/scans/scan28/) do Projeto Honeynet como exemplo. Eu peguei o arquivo day1.log.gz , descompactei , e executei o p0f:

```
[dani@sec /home/daniel]$ gunzip day1.log.gz
[dani@sec /home/daniel]$ p0f -s day1.log -f p0f.fp
```

```
4.167.44.129 [21 hops]: Windows XP Pro, Windows 2000 Pro
203.69.233.93 [20 hops]: Linux 2.2.9 - 2.2.18
67.36.28.116 [17 hops]: Windows 2000 (4)
61.219.90.180 [21 hops]: Linux 2.4.2 - 2.4.14 (1)
192.168.100.28 [1 hops]: SunOS 5.8
…
...
```

Essa sintaxe do p0f lista o sistema operacional de todos os Ips encontrados. E por meio dele foi possível descobrir todos os sistemas operacionais e ainda responder a questão 1 (http://www.honeynet.org/scans/scan28/) (qual o sistema operacional do HoneyPot , IP 192.168.100.28) de bandeja (192.168.100.28 [1 hops]: SunOS 5.8). Um outro modo de utilizá-lo é colocando-o para analisar algum dispositivo de rede (eth0 , por exemplo). Nesse modo ele irá tentar detectar o sistema operacional de todos os Ips que passarem por esse dispositivo :

```
[dani@sec /home/daniel]$ p0f –i eth0 –f p0f.fp

p0f: passive os fingerprinting utility, version 1.8
(C) Michal Zalewski <lcamtuf@gis.net>, William Stearns <wstearns@pobox.com>
p0f: file: 'p0f.fp', 92 fprints, iface: 'eth0', rule: 'all'.
Kernel filter, protocol ALL, TURBO mode (671 frames), raw packet socket
192.168.1.163 [1 hops]: Windows NT 4.0 (1) *
192.168.1.33 [1 hops]: Windows 2000 (9)
192.168.1.106 [1 hops]: Linux 2.4.2 - 2.4.14 (1)
…
…
```

Conclusão

Espero que esse texto tenha lhe ajudado a entender um pouco mais sobre "OS Passive fingerprinting" e ensinado um pouquinho mais sobre TCP/IP. Ele não está completo, tendo muito assunto para ser dito ainda. Qualquer problema, dúvida e sugestão, me envie um "mailto:daniel@opensolutions.com.br"