

SOME THINGS BEFORE NETWORK ATTACK (A LONG TIME OBSERVATION) 網路攻擊之前的二三事

Canaan Kao, Chuang Wang, I-Ju Liao

canaan@totoro.cs.nthu.edu.tw

canaan_kao@trend.com.tw

AGENDA

- The Motivation
- Port Scan 101
- IDS/IDP-based Port Scan Detection
- Honeypot-based Port Scan Detection
 - Play with SDN switch
- A Long Time Observation
 - A legacy of Anti-Botnet Project

WHO AM I?

- 十幾年前，在讀大學的時候，寫 OpenSource 的網管軟體報告，抽簽抽到 Snort.
- 後來在一家做 IDS/IPS 的公司，寫了十幾年的 code。
- 之後意外地，在我青春的尾巴，執行了教育部的 Anti-botnet 計畫五年，辦了四屆的 Botnet of Taiwan (BoT) 研討會。
 - ~~不要問我今年有沒有 BoT2014?~~
- 去年不小心成為 Anti-Virus 廠商的員工。

WHO AM I?

一些曾經講過的

- 2010 Spam Source Detection at Home
 - <http://www.anti-botnet.edu.tw/content/confs/BoT2010.PPTs/B5.php>
- 2012 The Botnet Traffic Forensics System
 - <http://www.anti-botnet.edu.tw/content/confs/BoT2012.PPTs/B5.php>
- 2013 APT/Malware Traffic Detection
 - <http://www.anti-botnet.edu.tw/content/confs/BoT2013.PPTs/B5.php>

一些縮寫

- **IDS**: 不具備阻擋功能的入侵偵測系統
 - EX: Snort
- **IPS**: 具備阻擋功能的入侵偵測系統
 - EX: Snort-inline
- **FW**: FireWall 防火牆
 - EX: NetFilter / iptables
- **LAN**: 以 FW 為界的內網
- **WAN**: 以 FW 為界的外網
- **SDN**: Software-defined Network

THE MOTIVATION

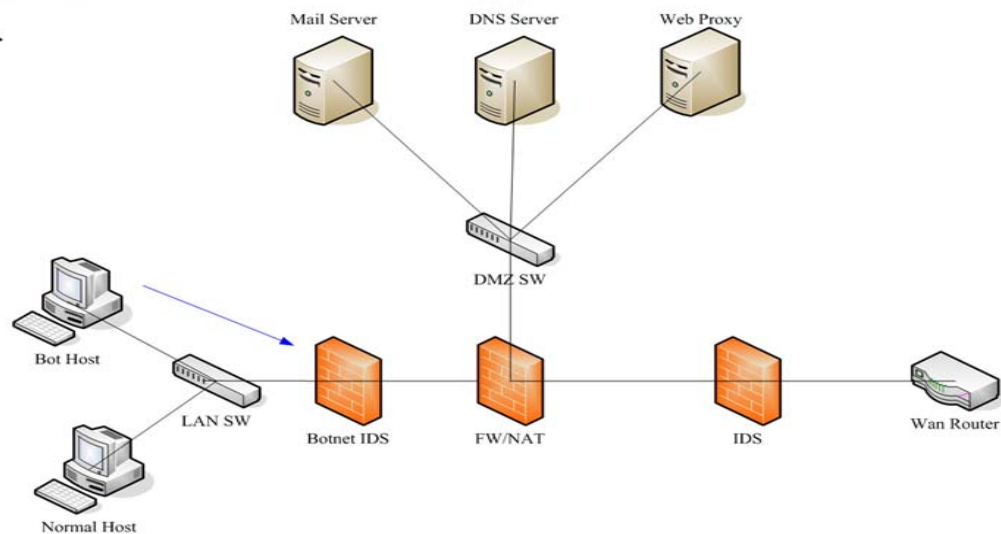
關於內賊(BOT)的偵測位置



1. Botnet Detection

Botnet IDS的防禦位置

- ❖ Botnet IDS 與一般的IDS要分工合作（如下），防禦外侮與保護 servers，不是 Botnet IDS 的責任。
- ❖ Botnet IDS 應該要
- ❖ 得到第一手的
- ❖ 使用者網路行為
- ❖ 才能精準判斷
- ❖ 內賊是誰!!



2012/09/07

The Botnet Traffic Forensics System

24

THE MOTIVATION

關於偵測的時機



0. 一個關於 *Detection Rate* 的故事

- ❖ 在講故事之前，先把一些“詞”先定義一下：
 - 這裡所謂的“偵測”，
 - 都是所謂的“事後偵測”，
 - 指的都是偵測**被入侵之後**該 **Bot/Victim** 的網路行為，進而找出未被發現的 **Bot/Victim**。

2013/09/13

The Current Methodologies for
APT/Malware Traffic Detection

4

THE MOTIVATION

只能事後偵測嗎？

- 假設 Malware 透過行動載具或是其他方式已經進入到內網，我們有什麼方式可以**察覺**或是阻止內網的設備 受到攻擊/感染？
- 或是我們只能做尋找哪些主機已經變成 bot 的事後偵測？
- 如果攻擊的**封包完全不經過** GW / FW / IDS / IPS，那我們還能偵測得到嗎？

THE MOTIVATION

IDS/IPS 產業公開的秘密

- 針對網路上的攻擊，基本上是廠商必須先拿到攻擊樣本或是惡意程式，其所屬的 IDS 或是 IPS 才會有偵測率。
- 所以如果遇到 0day，或是新式攻擊，被攻擊成功的機會就很大。
- 因此，針對 Botnet / APT，做事後的偵測是比較有把握的。
- But....

THE MOTIVATION

看個新聞 (智慧家電越來越多了)

日本智慧冰箱會傳照片

2013年12月05日  讚  10  +1  0



夏普員工日前展示用平板電腦，監控家電的家庭能源管理系統。路透

全球

【國際中心／綜合外電報導】美國消費性電子展 (Consumer Electronics Show, CES) 下月將在拉斯維加斯登場，日本家電業者無不卯足全力，準備推出各種智慧家電，像會傳送訊息的冰箱、聲控洗衣機等，希望扳回在電視、音響等客廳家電輸給南韓同業的劣勢。

路透昨指，Panasonic、東芝 (TOSHIBA) 正積極研發智慧家電，包括會傳送晚餐菜色照片的冰箱，以及聲控洗衣機等。這些家電能透過雲端彼此對話，現階段號稱可省下三成

- 來源：蘋果日報

THE MOTIVATION

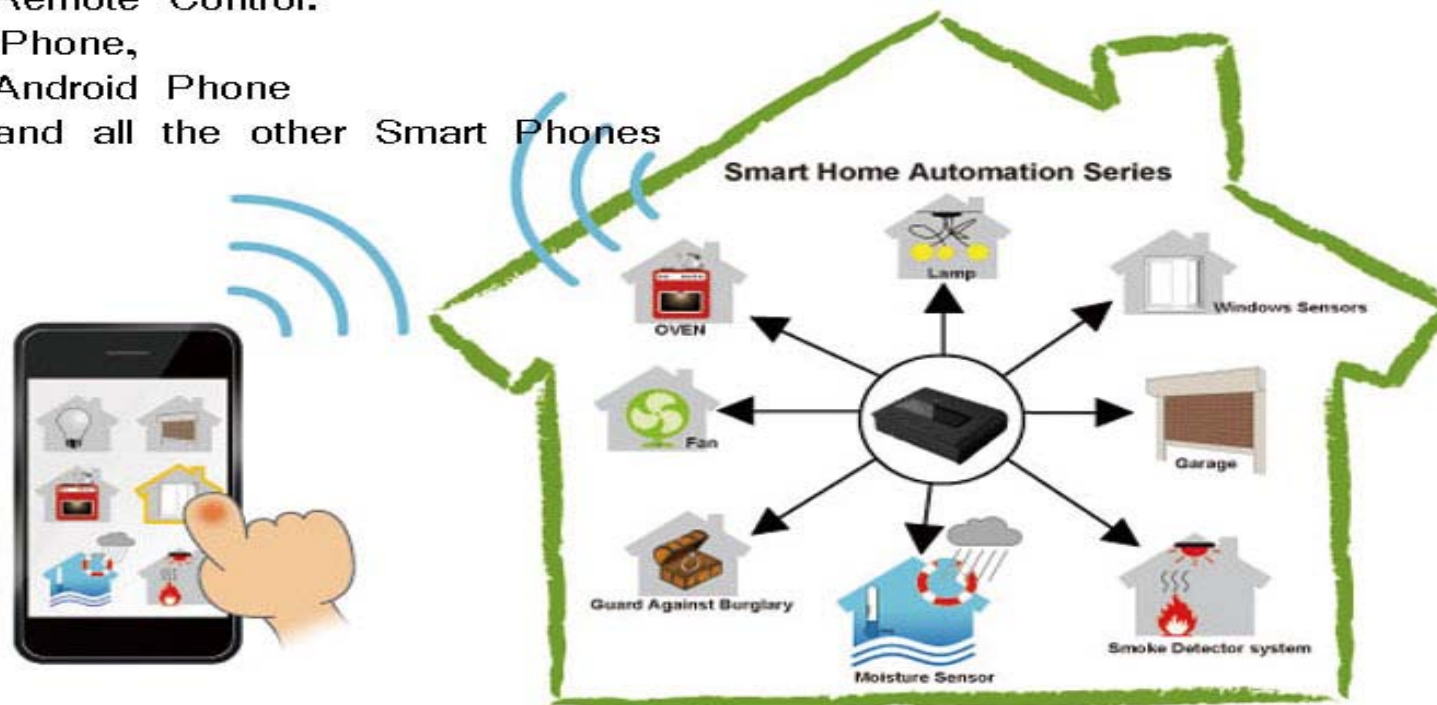
以後家庭生活都可以透過網路控制

Remote Control:

iPhone,

Android Phone

and all the other Smart Phones



http://server1.she777.com/images/www.joybien.com/images/HOME/SmartHome_760x500.jpg

THE MOTIVATION

一個問題

- 假設有一個攻擊 **智慧冰箱** 的 Malware，且這個 Malware 已經殖入你的行動裝置。而你回家的時候，它也跟你一起回家。
- 它要怎麼知道你家 **有可以攻擊的** 智慧冰箱呢？

THE MOTIVATION

- 最簡單的 probe 方式就是 port scan 。
- IDS / IPS / FW 應該要有反應？不是嗎？
 - 等一下會解釋為什麼它們可能不會叫。

PORT SCAN 101

- 基本上 Port Scan 可以分成兩種：
 - Vertical Scans
 - Single Host Target
 - Nmap 預設是這種
 - Horizontal Scans
 - Single Service Port Target
 - aka Port Sweep Scan
 - Bot/Malware 比較常用這種

PORT SCAN 101

- Port Scan 最主要想知道兩件事
 - 1. 目標機器有沒有開？
 - 發 TCP Syn 無回？
 - 2. 如果有開(有回)，那 Service 有沒有開？
 - 回 SYN+ACK
 - 回 RST+ACK

PORT SCAN 101

- 不過 Port Scan 人人會，巧妙各有不同。
 - Nmap
 - Bot/Malware
 - Bot/Malware 的掃法和你想的不太一樣
 - Internet Scan
 - 這陣子很流行

Port Scan 101

Nmap (1K ports/30 seconds)

File: "V:\0_LIB.Traffic.App\NMap\NMap_..." Pockets: 2383 Dis... Profile: Default

No.	Time	Source	Destination	Protocol	Length	Info
2348	2011-11-04 16:19:16.738115	192.168.0.122	192.168.168.100	SSHV2	406	Client: Key Exchange Init
2349	2011-11-04 16:19:16.776901	192.168.168.100	192.168.0.122	TCP	60	ssh > joost [ACK] Seq=4193338699 Ack=860417494 Win=6912 Len=0
2350	2011-11-04 16:19:16.777044	192.168.0.122	192.168.168.100	TCP	81	omscontact > ddm-rdb [PSH, ACK] Seq=3481229630 Ack=4193059891 Win=6570
2351	2011-11-04 16:19:16.777107	192.168.0.122	192.168.168.100	TCP	54	ddm-rdb > omscontact [ACK] Seq=4193059891 Ack=3481229657 Win=9088 Len=0
2352	2011-11-04 16:19:16.777405	192.168.168.100	192.168.0.122	TCP	60	ddm-rdb > omscontact [ACK] Seq=4193059891 Ack=3481229657 Win=9088 Len=0
2353	2011-11-04 16:19:16.777406	192.168.168.100	192.168.0.122	TCP	60	ddm-rdb > omscontact [ACK] Seq=4193059891 Ack=3481229658 Win=9088 Len=0
2354	2011-11-04 16:19:16.828946	192.168.0.122	192.168.168.100	SSHV2	198	Client: Diffie-Hellman Key Exchange Init
2355	2011-11-04 16:19:16.829316	192.168.168.100	192.168.0.122	TCP	60	ssh > joost [ACK] Seq=4193338699 Ack=860417638 Win=8000 Len=0
2356	2011-11-04 16:19:16.881697	192.168.168.100	192.168.0.122	SSHV2	774	Server: New Keys
2357	2011-11-04 16:19:16.924258	192.168.0.122	192.168.168.100	TCP	62	mbl-battd > 18128 [SYN] Seq=3604849022 Win=8192 [TCP CHECKSUM INCORRECT]
2358	2011-11-04 16:19:16.924776	192.168.168.100	192.168.0.122	TCP	60	18128 > mbl-battd [RST, ACK] Seq=0 Ack=3604849023 Win=0 Len=0
2359	2011-11-04 16:19:16.934991	192.168.0.122	192.168.168.100	TCP	54	altcp > ssh [FIN, ACK] Seq=3738093856 Ack=4194991248 Win=65700 [TCP CHECKSUM INCORRECT]
2360	2011-11-04 16:19:16.935009	192.168.0.122	192.168.168.100	TCP	54	joost > ssh [FIN, ACK] Seq=860417638 Ack=4193339419 Win=65700 [TCP CHECKSUM INCORRECT]
2361	2011-11-04 16:19:16.937409	192.168.168.100	192.168.0.122	TCP	60	ssh > altcp [FIN, ACK] Seq=4194991248 Ack=3738093857 Win=8000 Len=0
2362	2011-11-04 16:19:16.937465	192.168.0.122	192.168.168.100	TCP	54	altcp > ssh [FIN, ACK] Seq=3738093857 Ack=4194991249 Win=65700 [TCP CHECKSUM INCORRECT]
2363	2011-11-04 16:19:16.939036	192.168.168.100	192.168.0.122	TCP	60	ssh > joost [FIN, ACK] Seq=4193339419 Ack=860417639 Win=8000 Len=0
2364	2011-11-04 16:19:16.939095	192.168.0.122	192.168.168.100	TCP	54	joost > ssh [ACK] Seq=860417639 Ack=4193339420 Win=65700 [TCP CHECKSUM INCORRECT]
2365	2011-11-04 16:19:19.055674	Toshiba_7a:45:28	Broadcast	ARP	60	Who has 192.168.168.100? Tell 192.168.0.149
2366	2011-11-04 16:19:26.934562	192.168.0.122	192.168.168.100	TCP	66	ddgn > 12286 [SYN] Seq=3668896131 Win=8192 [TCP CHECKSUM INCORRECT]
2367	2011-11-04 16:19:26.935058	192.168.168.100	192.168.0.122	TCP	60	12286 > ddgn [RST, ACK] Seq=0 Ack=3668896132 Win=0 Len=0
2368	2011-11-04 16:19:27.434143	192.168.0.122	192.168.168.100	TCP	66	ddgn > 12286 [SYN] Seq=3668896131 Win=8192 [TCP CHECKSUM INCORRECT]
2369	2011-11-04 16:19:27.434430	192.168.168.100	192.168.0.122	TCP	60	12286 > ddgn [RST, ACK] Seq=0 Ack=3668896132 Win=0 Len=0
2370	2011-11-04 16:19:27.934143	192.168.0.122	192.168.168.100	TCP	62	ddgn > 12286 [SYN] Seq=3668896131 Win=8192 [TCP CHECKSUM INCORRECT]
2371	2011-11-04 16:19:27.934547	192.168.168.100	192.168.0.122	TCP	60	12286 > ddgn [RST, ACK] Seq=0 Ack=3668896132 Win=0 Len=0
2372	2011-11-04 16:19:27.949319	Micro-St_38:64:27	Broadcast	ARP	60	Who has 192.168.168.100? Tell 192.168.20.105
2373	2011-11-04 16:19:31.865626	Toshiba_47:be:23	Broadcast	ARP	60	Who has 192.168.168.100? Tell 192.168.0.130
2374	2011-11-04 16:19:37.934556	192.168.0.122	192.168.168.100	UDP	56	Source port: 64284 Destination port: 10763
2375	2011-11-04 16:19:37.934935	192.168.168.100	192.168.0.122	ICMP	84	Destination unreachable (Port unreachable)
2376	2011-11-04 16:19:37.935406	192.168.0.122	192.168.168.100	UDP	56	Source port: 64285 Destination port: 27851

```

0000  c6 53 a4 fe 62 c3 f0 de f1 70 25 8e 08 00 45 00  .S..b...p%...E.
0010  00 2c fb 24 00 00 39 06 5c 78 c0 a8 00 7a c0 a8  ..,.$..9.\x...z.
0020  a8 64 e8 7a 00 00 8b 6e 6c f7 f4 00 00 00 00 02  .d.z..nl .....
0030  08 00 16 90 00 00 02 04 05 b4                .....
  
```

PORT SCAN 101

BOT/MALWARE-PERL-BOT(ESKENT)

```
317 my $funcarg = $_[1];
318 if (my $pid = fork) {
319     waitpid($pid, 0);
320 } else {
321     if (fork) {
322         exit;
323     } else {
324         if ($funcarg =~ /^portscan (.*)/) {
325             my $hostip="$1";
326             my @portas=("21","22","23","25","53","80","110","143");
327             my (@aberta, %porta_banner);
328             foreach my $porta (@portas) {
329                 my $scansock = IO::Socket::INET->new(PeerAddr => $hostip, PeerPort => $porta, Protc
330                 if ($scansock) {
331                     push (@aberta, $porta);
332                     $scansock->close;
333                 }
334             }
335             if (@aberta) {
336                 sendraw($IRC_cur_socket, "PRIVMSG $printl :Portas abertas: @aberta");
337             } else {
338                 sendraw($IRC_cur_socket, "PRIVMSG $printl :Nenhuma porta aberta foi encontrada.");
```

PORT SCAN 101

BOT/MALWARE-ILEGALBRAIN_PERLBOT

```
151 "\001BitchX-1.1-final+ by panasync - FreeBSD 5.3-RELEASE : Keep it to yourself!\001",
152 "\001bitchx-1.0c18 :tunnelvision/1.2\001","\001PnP 4.22 - http://www.pair.com/\001",
153 "\001BitchX-1.0c17/FreeBSD 4.10-RELEASE:(c)rackrock/bX [3.0.109] : Keep it to yourself!\001",
154 "\001P&P 4.22.2 (in development) + X Z P Bots, Sound, NickServ, ChanServ, Extras\001",
155 "\001HydraIRC v0.3.148 (18/Jan/2005) by Dominic Clifton aka Hydra - #HydraIRC on EFNet\001",
156 "\001lirssi v0.8.10 - running on Linux i586\001","\001lirssi v0.8.10 - running on FreeBSD i386\001",
157 "\001ircII 20050423+ScrollZ 1.9.5 (19.12.2004)+Cdcc v1.6mods v1.0 by acidflash - Almost there\00",
158 "\001ircII 20050423+ScrollZ 1.9.5 (19.12.2004)+Cdcc v1.8+OperMods v1.0 by acidflash - Almost the
159
160 # Default quick scan ports
161 my @portas=("21","22","23","25","53","80","110","113","143","3306","4000","5900","6667","6668","666
162
163 # xeQt
164
165 #my $nick = "Power";
166 my $nick = $nickname[rand scalar @nickname];
167 my $realname = $xname[rand scalar @xname];
168 my $ircname = $xident[rand scalar @xident];
169 my $porta = $rports[rand scalar @rports];
170 my $xproc = $fakeps[rand scalar @fakeps];
```

PORT SCAN 101

INTERNET SCAN

Why scan the Internet (defensive)

- How many systems are vulnerable to Heartbleed?
- How many systems can be used for NTP amplification?
- How many systems vulnerable to D-Link router vulnerability/
- Survey all SSL certificates in use



scan the subnet, the
slash 0 subnet

PORT SCAN 101

INTERNET SCAN

Why scan the Internet (offensive)

- Uh, it's the deepnet
- Pick a random port, run masscan with “—banners”, and you find something hackable within minutes



bug. There's also
an offensive

PORT SCAN 101

MASSCAN -P80 140.114.71.0/24 --RATE=10000

masscan2.pcap [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save Filter TCP/UDP TCP-RST

Time	Source	Destination	Protocol	Length	Info
4	2014-08-20 09:46:02.776916	172.16.1.133	140.114.71.185	TCP	54 59162 > http [SYN] Seq=1745398879 win=1024 Len=0
5	2014-08-20 09:46:02.776964	172.16.1.133	140.114.71.158	TCP	54 59162 > http [SYN] Seq=3005133713 win=1024 Len=0
6	2014-08-20 09:46:02.776989	172.16.1.133	140.114.71.1	TCP	54 59162 > http [SYN] Seq=3312673648 win=1024 Len=0
7	2014-08-20 09:46:02.777011	172.16.1.133	140.114.71.212	TCP	54 59162 > http [SYN] Seq=4081583491 win=1024 Len=0
8	2014-08-20 09:46:02.777033	172.16.1.133	140.114.71.201	TCP	54 59162 > http [SYN] Seq=1847405815 win=1024 Len=0
9	2014-08-20 09:46:02.777054	172.16.1.133	140.114.71.220	TCP	54 59162 > http [SYN] Seq=3206470874 win=1024 Len=0
10	2014-08-20 09:46:02.777076	172.16.1.133	140.114.71.7	TCP	54 59162 > http [SYN] Seq=573344946 win=1024 Len=0
11	2014-08-20 09:46:02.777097	172.16.1.133	140.114.71.213	TCP	54 59162 > http [SYN] Seq=1440332228 win=1024 Len=0
12	2014-08-20 09:46:02.777120	172.16.1.133	140.114.71.54	TCP	54 59162 > http [SYN] Seq=4019563718 win=1024 Len=0
13	2014-08-20 09:46:02.777141	172.16.1.133	140.114.71.253	TCP	54 59162 > http [SYN] Seq=279083756 win=1024 Len=0
14	2014-08-20 09:46:02.777163	172.16.1.133	140.114.71.252	TCP	54 59162 > http [SYN] Seq=3872041853 win=1024 Len=0
15	2014-08-20 09:46:02.777185	172.16.1.133	140.114.71.204	TCP	54 59162 > http [SYN] Seq=3919498759 win=1024 Len=0
16	2014-08-20 09:46:02.779480	172.16.1.133	140.114.71.69	TCP	54 59162 > http [SYN] Seq=4056579132 win=1024 Len=0
17	2014-08-20 09:46:02.779533	172.16.1.133	140.114.71.38	TCP	54 59162 > http [SYN] Seq=2683566549 win=1024 Len=0
18	2014-08-20 09:46:02.779556	172.16.1.133	140.114.71.219	TCP	54 59162 > http [SYN] Seq=3847272336 win=1024 Len=0
19	2014-08-20 09:46:02.779578	172.16.1.133	140.114.71.198	TCP	54 59162 > http [SYN] Seq=950804 win=1024 Len=0
20	2014-08-20 09:46:02.779601	172.16.1.133	140.114.71.83	TCP	54 59162 > http [SYN] Seq=4023903264 win=1024 Len=0
21	2014-08-20 09:46:02.779624	172.16.1.133	140.114.71.14	TCP	54 59162 > http [SYN] Seq=2963046316 win=1024 Len=0
22	2014-08-20 09:46:02.779646	172.16.1.133	140.114.71.237	TCP	54 59162 > http [SYN] Seq=1294159946 win=1024 Len=0
23	2014-08-20 09:46:02.779668	172.16.1.133	140.114.71.97	TCP	54 59162 > http [SYN] Seq=4251947299 win=1024 Len=0
24	2014-08-20 09:46:02.779689	172.16.1.133	140.114.71.34	TCP	54 59162 > http [SYN] Seq=3422659798 win=1024 Len=0

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: Intel_b2:6b:ec (00:d0:b7:b2:6b:ec), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

```
0000 ff ff ff ff ff ff 00 d0 b7 b2 6b ec 08 06 00 01 ..... .k.....
0010 08 00 06 04 00 01 00 d0 b7 b2 6b ec ac 10 01 85 ..... .k.....
0020 00 00 00 00 00 00 ac 10 01 02 00 00 00 00 00 .....
0030 00 00 00 00 00 00 00 00 00 00 00 00 ..... ..
```

File: "C:\Users\Administrator\Desktop\... Profile: Default

PORT SCAN 101

INTERNET SCAN (一些相關單位)

Internet-wide scanning

- › Internet Mapping Project, Bell Labs / Lumeta, 1998+
- › IPv4 Census 2003-2006
- › EFF SSL Observatory 2014
- › Internet Census 2012 (the botnet)
- › Shodan
- › RIPE Atlas (slightly different)
- › Critical.IO, 2012-2013
- › University of Michigan
- › Shadowserver
- › ErrataSec (R. Graham / masscan)
- › Rapid7, Project Sonar

RAPID7

Source: us-14-Schloesser-Internet-Scanning-Current-State-And-Lessons-Learned.pdf

PORT SCAN 101

INTERNET SCAN

- 以前，我們會想，我們把重要的 Service 放在 Internet Scanning 掃不到的地方，不就好了？
 - 例如：放在 LAN 端，有 FW 保護，不開 Virtual Server 或是 Port Mapping，只對內服務，這樣不就沒事了？

IDS/IDP-BASED PORT SCAN DETECTION

- Snort v2.9.2 的 default setting 是這樣

```
# Portscan detection. For more information, see README.sfportscan  
# preprocessor sfportscan: proto { all } memcap { 10000000 } sense_level { low }
```

- 預設是 **disabled**
- Detection Level: **low**
 - For getting few **false positives**.
 - Time window is **60** seconds.

IDS/IDP-BASED PORT SCAN DETECTION

- 基本上是計算單位時間內發現的 port scan 事件次數。
 - 是一個 threshold。
 - 只要低於 threshold 就可以繞過。
 - False Positive?
 - 某些正常連線看起來會像 port scan 的行為。
 - 那基準值/參考值是什麼？

IDS/IDP-BASED PORT SCAN DETECTION

- 如果你今天買了台具備偵測 Port Scan 能力的 IDS / IPS / FW，你會怎麼驗？
 - 大家都愛 Nmap 😊
 - 有人會養個 bot 掃掃看嗎？
 - 所以針對 bot / malware 所發出的 port scan，如果你買的那個資安設備不會叫，是可以了解/諒解的。

IDS/IDP-BASED PORT SCAN DETECTION

- 如果今天 port scan 的 packets，不經過 IDS / IPS / FW呢？
 - 法外之地？
 - LAN <-> LAN traffic
 - 如果Traffic有流經FW的 LAN Ports，之前的資安設備會假設這個方向的 traffic 應該不會有攻擊，所以 **通常不檢查**，採用硬體交換居多。
 - Wireless LAN (WLAN) <-> LAN traffic

IDS/IDP-BASED PORT SCAN DETECTION

對於 PORT SCAN 可能不會叫的原因

- 偵測的功能沒開？
 - 大家可以回去檢查一下 Home GW 的預設值
- 偵測的方式對不上
 - 清朝的劍與明朝的官
- Threshold 被繞過
- Traffic 沒經過

HONEYPOT-BASED PORT SCAN DETECTION

- 因為 LAN <-> LAN之間的 Attack 不會被 FW / IDS 看到，所以為了偵測 LAN <-> LAN 之間的 Attack，我們使用了 HoneyPot。

HONEYPOT-BASED PORT SCAN DETECTION

WHAT IS HONEYPOT?

- 就我個人的定義：
 - 所有可以用來誘使壞人或是惡意程式展露其行為或意圖的系統
 - 所以它可以是
 - 一台 Server
 - 一個 VM
 - 一個 Web Client
 -

HONEYPOT-BASED PORT SCAN DETECTION

- 簡單地說，這個方法就是用一個**影武者**設備(H)，放在需要被保護的主機(S)的旁邊，**H的IP也設在S的附近**。
- H完全不開 services，或是只開少量的 services，**外界完全不知道H的存在**，所以H只要收到來自不明主機(A)的**一個** TCP SYN for a closed port，就可以大膽判定A是 Scanner。

HONEYPOT-BASED PORT SCAN DETECTION

再看一次 SWEEP SCAN

masscan2.pcap [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save Filter TCP/UDP TCP-RST

Time	Source	Destination	Protocol	Length	Info
4	2014-08-20 09:46:02.776916	172.16.1.133	140.114.71.185	TCP	54 59162 > http [SYN] Seq=1745398879 win=1024 Len=0
5	2014-08-20 09:46:02.776964	172.16.1.133	140.114.71.158	TCP	54 59162 > http [SYN] Seq=3005133713 win=1024 Len=0
6	2014-08-20 09:46:02.776989	172.16.1.133	140.114.71.1	TCP	54 59162 > http [SYN] Seq=3312673648 win=1024 Len=0
7	2014-08-20 09:46:02.777011	172.16.1.133	140.114.71.212	TCP	54 59162 > http [SYN] Seq=4081583491 win=1024 Len=0
8	2014-08-20 09:46:02.777033	172.16.1.133	140.114.71.201	TCP	54 59162 > http [SYN] Seq=1847405815 win=1024 Len=0
9	2014-08-20 09:46:02.777054	172.16.1.133	140.114.71.220	TCP	54 59162 > http [SYN] Seq=3206470874 win=1024 Len=0
10	2014-08-20 09:46:02.777076	172.16.1.133	140.114.71.7	TCP	54 59162 > http [SYN] Seq=573344946 win=1024 Len=0
11	2014-08-20 09:46:02.777097	172.16.1.133	140.114.71.213	TCP	54 59162 > http [SYN] Seq=1440332228 win=1024 Len=0
12	2014-08-20 09:46:02.777120	172.16.1.133	140.114.71.54	TCP	54 59162 > http [SYN] Seq=4019563718 win=1024 Len=0
13	2014-08-20 09:46:02.777141	172.16.1.133	140.114.71.253	TCP	54 59162 > http [SYN] Seq=279083756 win=1024 Len=0
14	2014-08-20 09:46:02.777163	172.16.1.133	140.114.71.252	TCP	54 59162 > http [SYN] Seq=3872041853 win=1024 Len=0
15	2014-08-20 09:46:02.777185	172.16.1.133	140.114.71.204	TCP	54 59162 > http [SYN] Seq=3919498759 win=1024 Len=0
16	2014-08-20 09:46:02.779480	172.16.1.133	140.114.71.69	TCP	54 59162 > http [SYN] Seq=4056579132 win=1024 Len=0
17	2014-08-20 09:46:02.779533	172.16.1.133	140.114.71.38	TCP	54 59162 > http [SYN] Seq=2683566549 win=1024 Len=0
18	2014-08-20 09:46:02.779556	172.16.1.133	140.114.71.219	TCP	54 59162 > http [SYN] Seq=3847272336 win=1024 Len=0
19	2014-08-20 09:46:02.779578	172.16.1.133	140.114.71.198	TCP	54 59162 > http [SYN] Seq=950804 win=1024 Len=0
20	2014-08-20 09:46:02.779601	172.16.1.133	140.114.71.83	TCP	54 59162 > http [SYN] Seq=4023903264 win=1024 Len=0
21	2014-08-20 09:46:02.779624	172.16.1.133	140.114.71.14	TCP	54 59162 > http [SYN] Seq=2963046316 win=1024 Len=0
22	2014-08-20 09:46:02.779646	172.16.1.133	140.114.71.237	TCP	54 59162 > http [SYN] Seq=1294159946 win=1024 Len=0
23	2014-08-20 09:46:02.779668	172.16.1.133	140.114.71.97	TCP	54 59162 > http [SYN] Seq=4251947299 win=1024 Len=0
24	2014-08-20 09:46:02.779689	172.16.1.133	140.114.71.34	TCP	54 59162 > http [SYN] Seq=3422659798 win=1024 Len=0

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)

- Ethernet II, Src: Intel_b2:6b:ec (00:d0:b7:b2:6b:ec), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Address Resolution Protocol (request)

```
0000 ff ff ff ff ff ff 00 d0 b7 b2 6b ec 08 06 00 01 ..... .k.....
0010 08 00 06 04 00 01 00 d0 b7 b2 6b ec ac 10 01 85 ..... .k.....
0020 00 00 00 00 00 00 ac 10 01 02 00 00 00 00 00 .....
0030 00 00 00 00 00 00 00 00 00 00 00 ..... ..
```

File: "C:\Users\Administrator\Desktop\... Profile: Default

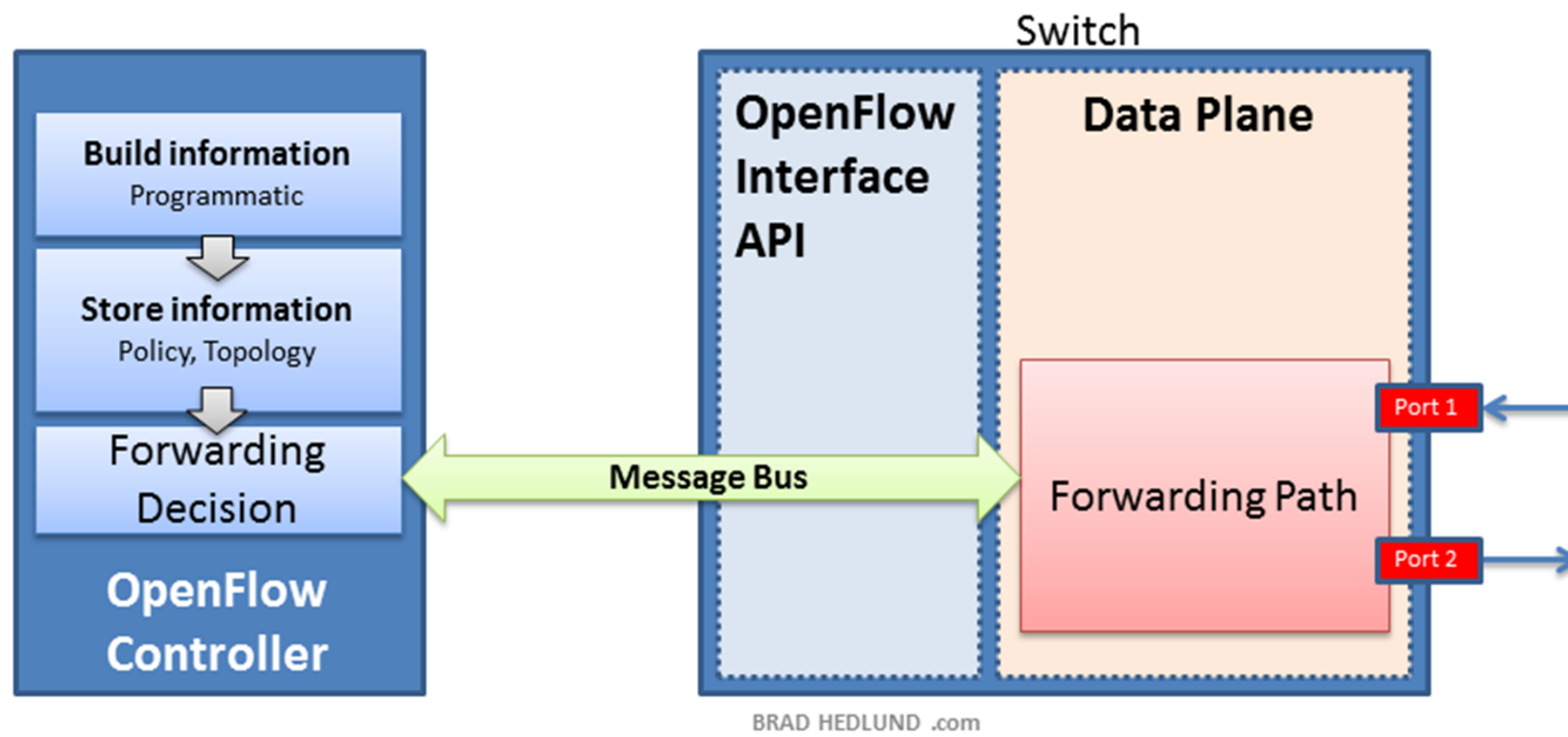
HONEYPOT-BASED PORT SCAN DETECTION

- 這個方法有個好處
 - 不管 scan 是 from WAN 或 from LAN，都可以偵測。
 - NO False-Positive 😊
 - 對付 Malware with BYOD/IoT 也行。
 - 掃再慢都抓得到 😊
- 這個方法的缺點
 - 萬一 A 沒掃到 H 呢？
 - 偵測到有人在掃，下一步呢？
 - FW 可以馬上擋，H 呢？

HONEYPOT-BASED PORT SCAN DETECTION

話說當今世上有個神器，叫 **SDN SWITCH**

Externally controlled Switch



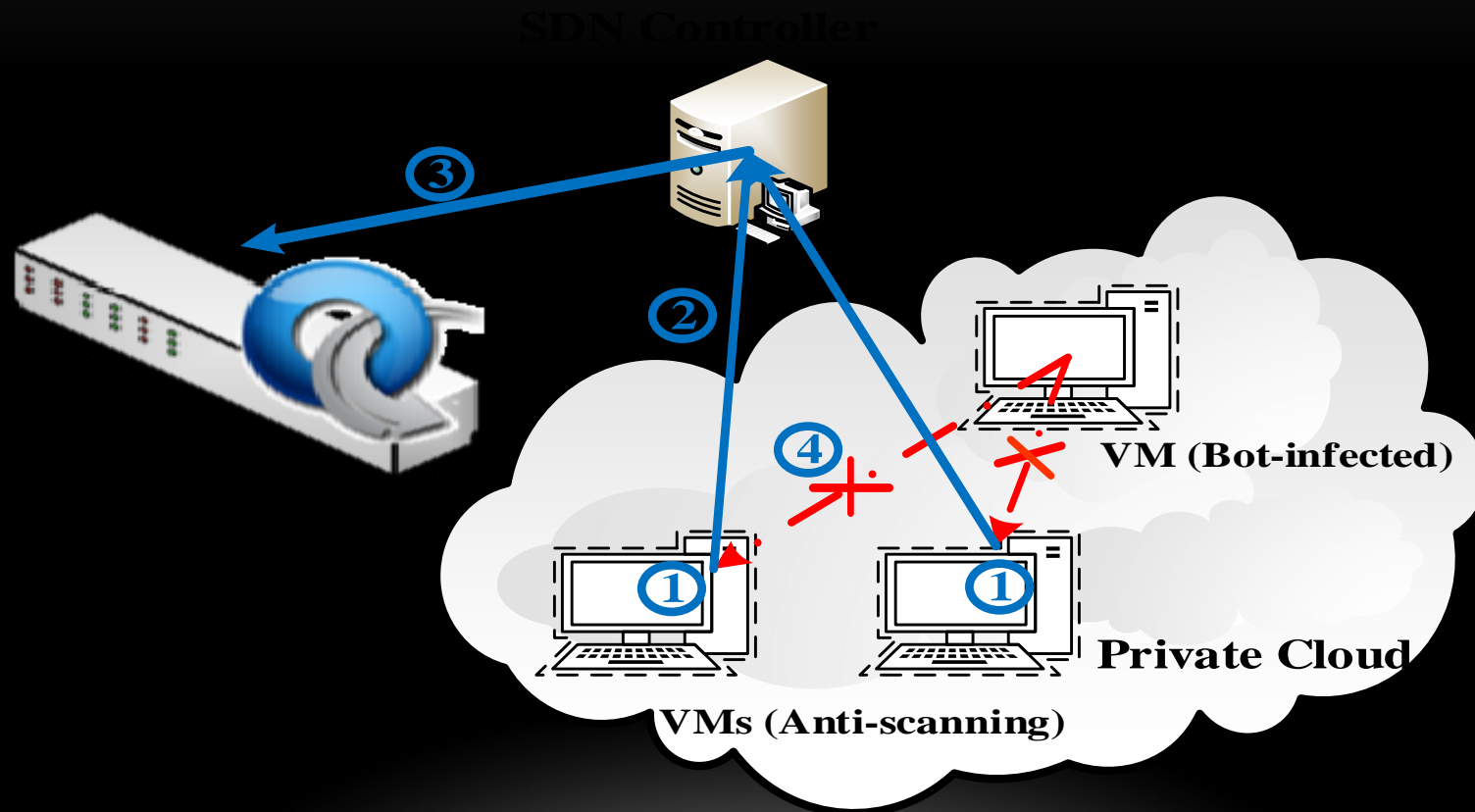
<http://bradhedlund.s3.amazonaws.com/2011/openflow-scale/openflow-switch.png>

HONEYPOT-BASED PORT SCAN DETECTION 實驗

- 為了簡化環境，我們把
 - Bot-infected host
 - Honeypot (Anti-Scanning)
 - 都接在同一台 SDN switch上。

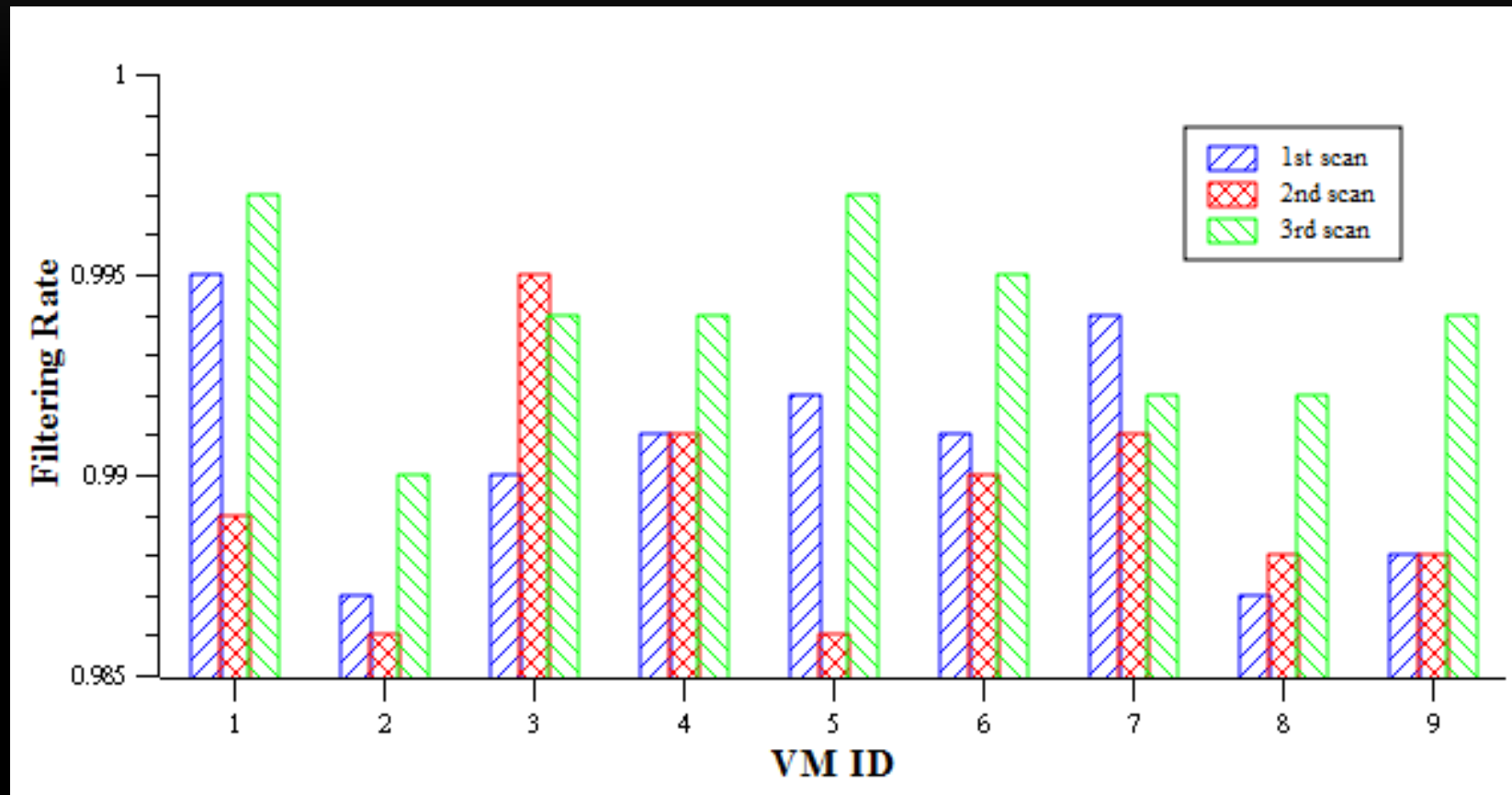
HONEYPOT-BASED PORT SCAN DETECTION

SDN SW + HONEYPOT



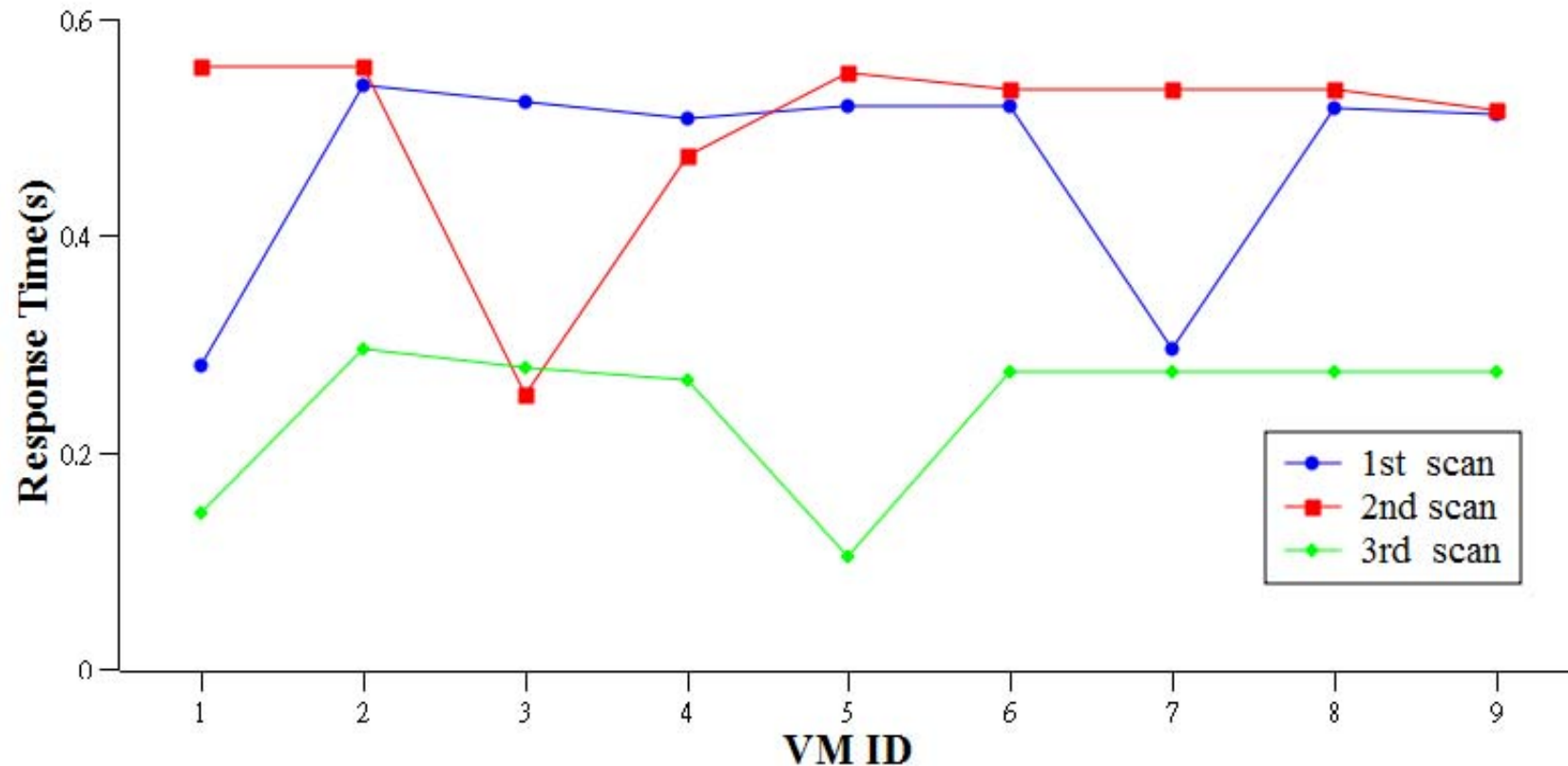
HONEYPOT-BASED PORT SCAN DETECTION

SDN SW + HONEYPOT (THE SCAN BLOCKING RATE)



HONEYPOT-BASED PORT SCAN DETECTION

SDN SW + HONEYPOT (THE RESPONSE TIME)



ABOUT LAN PORT SCAN DETECTION

小結

- 這樣看來針對 LAN <-> LAN 的 Scan，用
 - SDN SW + Honeypot 或許是一招
 - 0.6s 的反應時間
 - 98.5% 的阻擋率
 - 0% FP rate
- But, 我們還有更好的方法 😊
 - Maybe HITCON 2015?

同場加映

- A Long Time Observation
 - A **legacy** of Anti-Botnet Project (2009-2013)
 - <http://www.anti-botnet.edu.tw/>

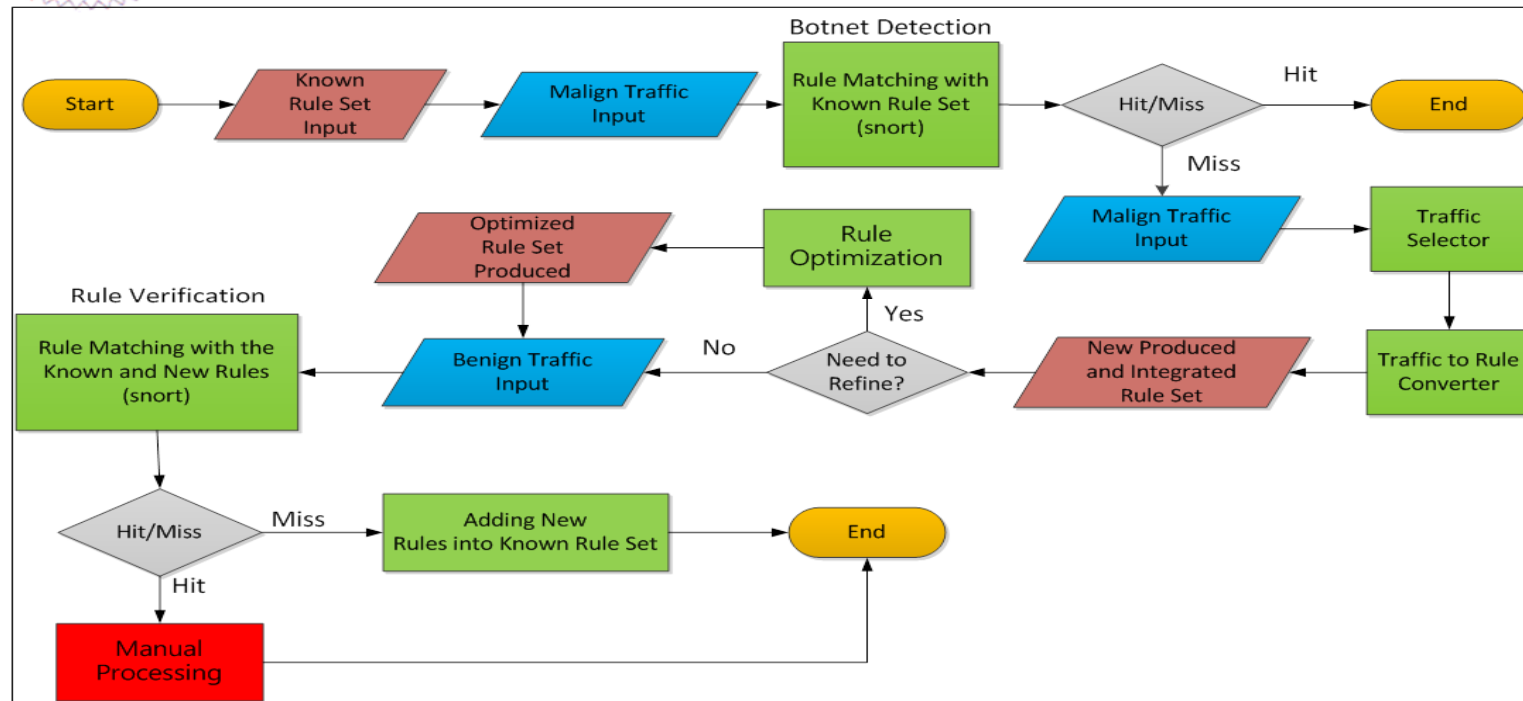


ABOUT ANTI-BOTNET PROJECT

THE FLOW OF AUTO-BOTNET-RULE GEN



0. About Anti-Botnet Project



2012/09/07

The Botnet Traffic Forensics System

16

ABOUT ANTI-BOTNET PROJECT BOTNET DETECTION RULE SERVICE

The screenshot shows the website for the Anti-Botnet Project. The main header features the project's logo and the title '教育學術網路系統安全與惡意程式偵測技術研發建置計畫'. A sidebar on the left contains navigation links for home, news, and various reports. The main content area is titled 'Rule 下載' and displays a table with the following data:

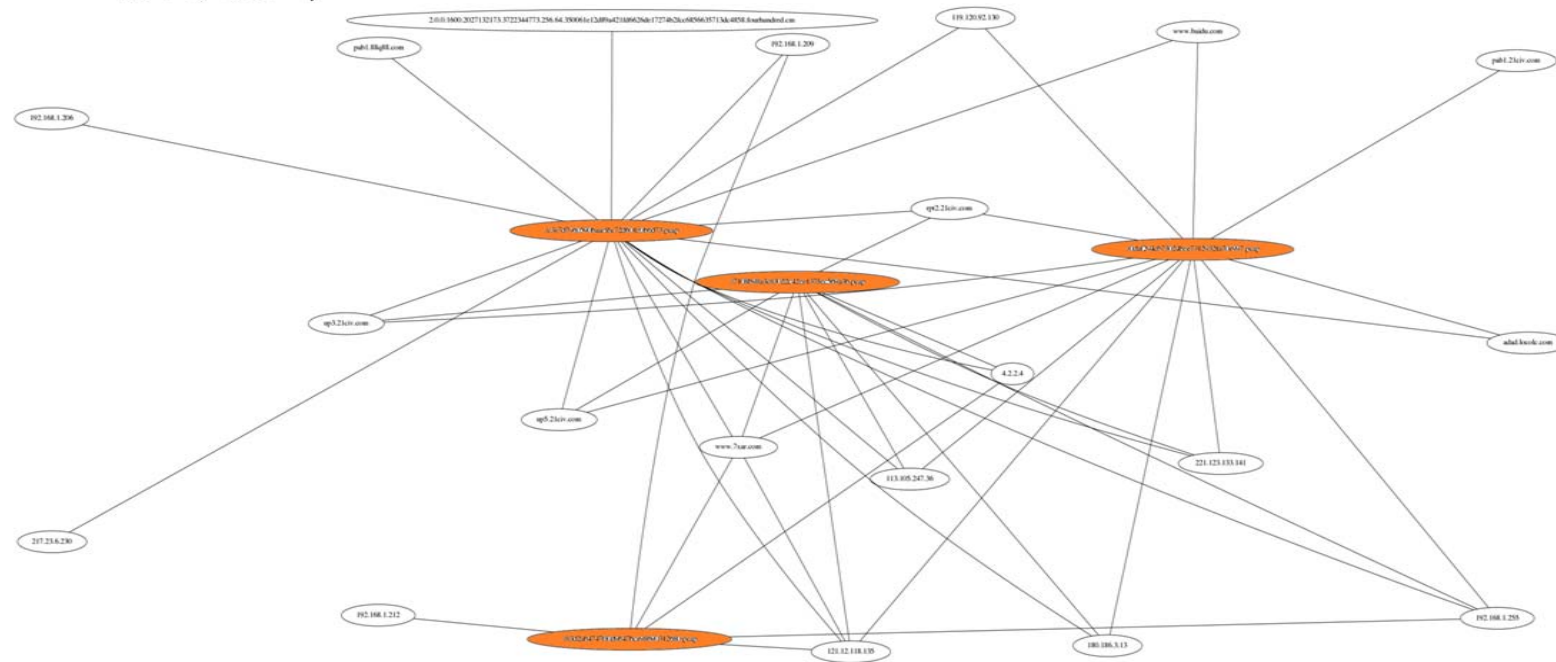
文件說明	版本	產出時間	與上個版本差異	下載	下載次數
rules v155	155	2013/12/30	下載	下載	1735
rules v154	154	2013/12/23	請下載最新版本	請下載最新版本	61
rules v153	153	2013/12/16	請下載最新版本	請下載最新版本	252
rules v152	152	2013/12/09	請下載最新版本	請下載最新版本	96
rules v151	151	2013/12/02	請下載最新版本	請下載最新版本	78
rules v150	150	2013/11/25	請下載最新版本	請下載最新版本	73
rules v149	149	2013/11/18	請下載最新版本	請下載最新版本	69
rules v148	148	2013/11/11	請下載最新版本	請下載最新版本	75
rules v147	147	2013/11/04	請下載最新版本	請下載最新版本	89
rules v146	146	2013/10/28	請下載最新版本	請下載最新版本	50
rules v145	145	2013/10/21	請下載最新版本	請下載最新版本	86

ABOUT ANTI-BOTNET PROJECT THE FORENSIC OF BOT NETWORK TRAFFIC



2. Known and Unknown: The Family

❖ 相同族群



2012/09/07

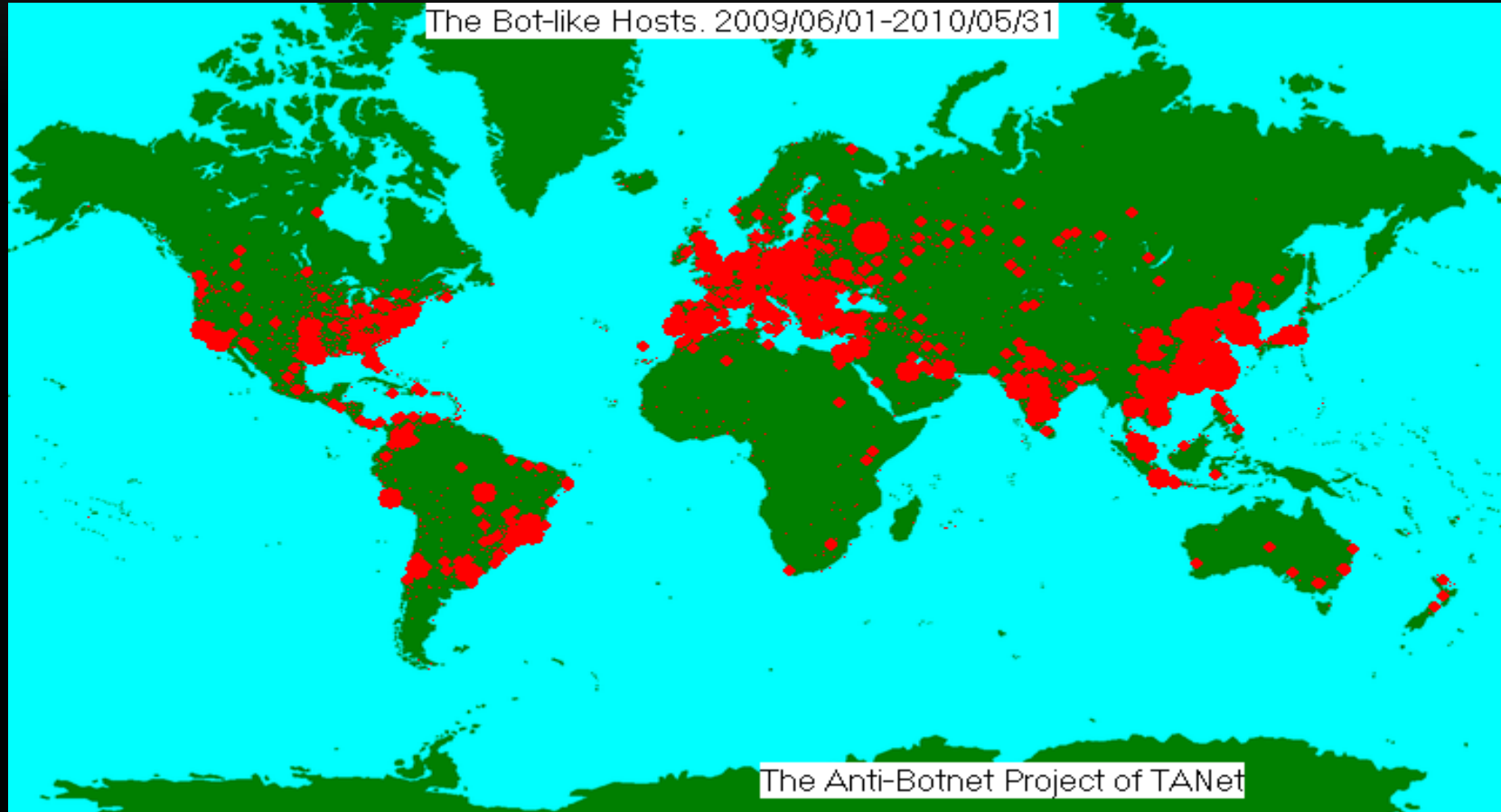
The Botnet Traffic Forensic System

42

A LONG TIME OBSERVATION (A LEGACY OF ANTI-BOTNET PROJECT)

- 這故事是這樣，在2009年的時候，因為 Anti-Botnet Project 的需要，我放了一個 HoneyPot-based port scan detector (影武者)....
- 因為會掃到影武者的，基本上都可以假設居心不良，所以我們用這個來產生 bot-like host distribution map。

A LONG TIME OBSERVATION (WHERE IS TAIWAN?)



A LONG TIME OBSERVATION (A LEGACY OF ANTI-BOTNET PROJECT)

- Anti-Botnet Project 在 2013 年結束，但是一些相關的設施並沒有完全撤除，我在前一些日子發現我有保留它自 2009 年以來的 log....
- 大數據分析？不，只是久數據。

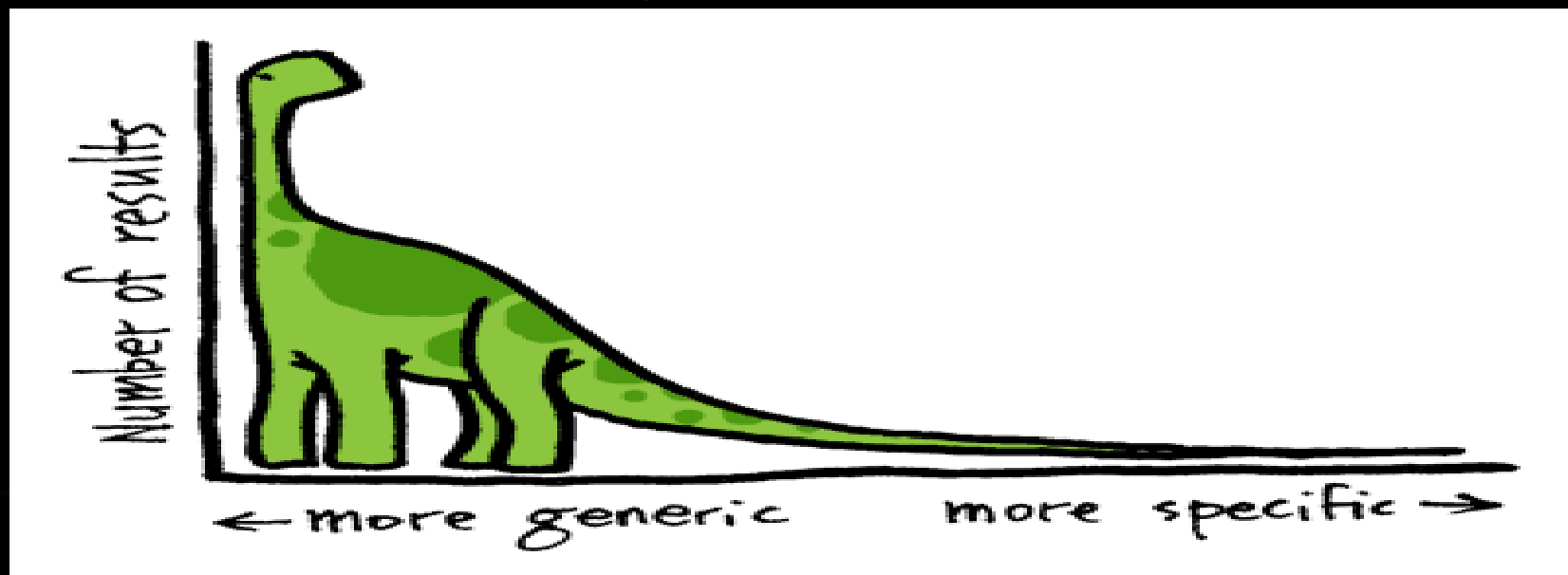
A LONG TIME OBSERVATION (大約有 1.2M 筆 LOGS)

```
1215854 201407200513,2014072005,20140720,201407,2014,46.170.251.18,36401,TCP,80
1215855 201407200516,2014072005,20140720,201407,2014,198.13.98.118,6000,TCP,1433
1215856 201407200517,2014072005,20140720,201407,2014,198.13.98.118,6000,TCP,1433
1215857 201407200517,2014072005,20140720,201407,2014,93.174.93.51,40065,TCP,13621
1215858 201407200520,2014072005,20140720,201407,2014,222.186.56.85,6000,TCP,3389
1215859 201407200528,2014072005,20140720,201407,2014,116.10.191.173,6000,TCP,22
1215860 201407200529,2014072005,20140720,201407,2014,85.120.232.38,23450,TCP,5555
1215861 201407200533,2014072005,20140720,201407,2014,203.171.234.229,6000,TCP,1433
1215862 201407200534,2014072005,20140720,201407,2014,116.10.191.173,6000,TCP,22
1215863 201407200536,2014072005,20140720,201407,2014,61.147.103.138,6000,TCP,1433
1215864 201407200545,2014072005,20140720,201407,2014,58.213.120.44,6000,TCP,3306
1215865 201407200545,2014072005,20140720,201407,2014,82.221.109.194,44734,TCP,80
1215866 201407200546,2014072005,20140720,201407,2014,82.221.109.194,44734,TCP,8000
1215867 201407200548,2014072005,20140720,201407,2014,219.217.246.24,12369,TCP,8000
1215868 201407200558,2014072005,20140720,201407,2014,122.0.114.49,6000,TCP,1433
1215869 201407200604,2014072006,20140720,201407,2014,116.10.191.212,6000,TCP,22
1215870 201407200606,2014072006,20140720,201407,2014,218.5.76.221,6000,TCP,9999
1215871 201407200606,2014072006,20140720,201407,2014,23.95.28.228,6000,TCP,3306
1215872 201407200614,2014072006,20140720,201407,2014,192.227.245.112,6000,TCP,1433
1215873 201407200618,2014072006,20140720,201407,2014,46.146.243.72,3790,TCP,4899
1215874 201407200618,2014072006,20140720,201407,2014,46.146.243.72,3790,TCP,4899
```

A LONG TIME OBSERVATION

這些年的一些統計

- 有 177,084 個 IP 掃到我們
- 有 23,273 個 TCP ports 被掃到 (Total:1,036,624 hits)
- 有 1,479 個 UDP ports 被掃到 (Total:179,250 hits)



A LONG TIME OBSERVATION

比較熱門的 TCP PORTS

Rank	Port #	Hit #	Ratio
1	1433, MS SQL Server ?	201395	19.43%
2	445, SMB?	185436	17.89%
3	9415, PPLive open proxy ?	64894	6.26%
4	1080, Socks Proxy or Back Door?	43778	4.22%
5	80	31769	3.06%
6	22	28751	2.77%
7	135, Remote Procedure Call (RPC)?	24874	2.40%
8	3306, MySql?	24745	2.39%
9	3389, Windows RDP?	22039	2.13%
10	8080	19998	1.93%

A LONG TIME OBSERVATION

比較熱門的 UDP PORTS

Rank	Port #	Hit #	Ratio
1	29285 (2009~2012)	132937	74.16%
2	22722	7701	4.30%
3	137, netbios-ns?	4818	2.69%
4	161, snmp?	4767	2.66%
5	5060, SIP?	4669	2.60%
6	53	4091	2.28%
7	5724, Operations Manager - SDK Service?	2969	1.66%
8	7793	2041	1.14%
9	19, chargen?	1113	0.62%
10	33348	618	0.34%

A LONG TIME OBSERVATION

當 RDP 發生問題 MS12-020 時(2012/03/13)

- TCP 3389 在整個統計區間的 ratio 是 2.13% .
 - 但是在 2012/03 那個月卻是 4%.
 - 再往前看 2012/02 那個月已先漲到 3.4%
 - 再往前看 2012/01 那個月就回到基本盤 2.0%
- 壞人在 MS12-020 發佈前就已經先拿出來打？
 - 如果我們夠 lucky 就可以先被打到 ☺

A LONG TIME OBSERVATION (A LEGACY OF ANTI-BOTNET PROJECT)

- 以上就是 Anti-Botnet 計畫的遺產，
- 讓我在計畫結束一年後，還有些東西可以跟大家分享 😊

ADAPT TO THE NEW ERA OF SECURITY THREATS.

- *It was the **best of times**, it was the worst of times.*
- 在 WAN 有 Internet Scanning，在 LAN 有 BYOD / IoT issues。
- 設備需要**升級**，人腦也需要升級。
 - 我有一個**夢**，當我的**冰箱**有對外的**TCP 6667**連線時，FW 能警告我，那有多好 😊
 - 人補**腦**需要更久的時間。

SUMMARY

- 我們不是只能做事後處理，而是我們面對各種攻擊時，IDS / IPS / FW 因為各種先天不良、後天失調的結果**可能不會叫**。(不是叫大家不要買啦)
- 除了在 FW 的 WAN 的端，**LAN** 也是戰場了。
- **SDN** switch 在 LAN <-> LAN Attack 的保護上應該幫得上忙。
- 如果我們夠**幸運**的話，HoneyPot-based port scan detector 也可以當成 **0-day** 的**預警**。(久數據的妙用?)
- 在未來，除了**智慧家電**外，應該也會有**智慧網安**的設備出現。
- 所以，在網路攻擊發生之前，我們還是可以做點事 😊

Q&A

ABOUT BOT2014



BoT2014?

❖ **In case I don't see you, good afternoon,
good evening, and good night.**



2013/09/13

The Current Methodologies for
APT/Malware Traffic Detection

43

ONE MORE THING

- 大約今年的十月中
- 相關 Honeypot-based PortScanDetector 會放在
- <https://github.com/canaankao/PortScanDetector>

REFERENCE

- 長尾分布的圖是引用自
 - http://1.bp.blogspot.com/_UfxPP3QC4us/SbVjYX-Qbil/AAAAAAAAAY0/v1a7zLipdfQ/s400/long-tail.png