

# Nmap Cheatsheet

## Contents

<b>1</b>	<b>Basic Nmap Commands</b>	<b>2</b>
1.1	Service and Version Detection (-sV) . . . . .	2
1.2	Default Script Scan (-sC) . . . . .	3
1.3	OS Detection (-O) . . . . .	4
1.4	Aggressive Scan (-A) . . . . .	5
<b>2</b>	<b>Host Discovery Commands</b>	<b>6</b>
2.1	ARP Scan (-PR) . . . . .	6
2.2	ICMP Scans . . . . .	7
2.2.I	Echo Scan (-PE) . . . . .	7
2.2.II	Timestamp Scan (-PP) . . . . .	8

This document shall contain everything I have learned so far about nmap. I must keep it up-to-date because I feel a bit inundated in this cybersecurity journey to be frank.

# 1 Basic Nmap Commands

## 1.1 Service and Version Detection (-sV)

This is the Service Detection flag (*yes; -sV is a single flag, not a combination of both s AND V*), which will tell you the name and description of the identified services.

### Service and Version Detection

```
$ sudo nmap -sV {target_IP}

Starting Nmap 7.80 ( https://nmap.org ) at 2025-03-28 06:27 GMT
Nmap scan report for 10.10.171.202
Host is up (0.00011s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE      VERSION
7/tcp     open  echo
9/tcp     open  tcpwrapped
13/tcp    open  daytime?
17/tcp    open  qotd?
22/tcp    open  ssh          OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu
Linux; protocol 2.0)
8008/tcp   open  http         lighttpd 1.4.74

2 services unrecognized despite returning data. If you know the
service/version, please submit the following fingerprints at
https://nmap.org/cgi-bin/submit.cgi?new-service :
# [Detailed fingerprint data omitted for brevity]

MAC Address: 02:89:EC:B7:74:EF (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.00 seconds
```

## 1.2 Default Script Scan (-sC)

I sometimes use this one instead of `-sV` because it runs *default scripts*<sup>1</sup>, which can give out additional information depending on the services running on the target. You can see a comparison in outputs between the two flags in the two boxes below.

### Default Script Scan

```
$ sudo nmap -p- -sC {target_IP} # Do you notice how I had to scan all ports, not
just the top 1000 most common?

Starting Nmap 7.80 ( https://nmap.org ) at 2025-03-28 09:22 GMT
Nmap scan report for 10.10.219.233
Host is up (0.00024s latency).
Not shown: 65526 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
|_smtp-commands: debra2.thm.local, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS
|_      , ENHANCEDSTATUSCODES, 8BITMIME, DSN, CHUNKING,
|_ssl-cert: Subject: commonName=debra2.thm.local
|_Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
|_ssl-date: TLS randomness does not represent time
53/tcp    open  domain
|_dns-nsid:
|_  bind.version: 9.18.28-1-deb12u2-Debian
80/tcp    open  http
|_http-title: Welcome to nginx on Debian!
110/tcp   open  pop3
|_pop3-capabilities: PIPELINING SASL UIDL STLS AUTH-RESP-CODE RESP-CODES CAPA TOP
|_ssl-cert: Subject: commonName=debra2.thm.local
|_Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
111/tcp   open  rpcbind
|_rpcinfo:
|_  program version    port/proto  service
|_  100000  2,3,4      111/tcp    rpcbind
|_  100000  2,3,4      111/udp    rpcbind
|_  100000  3,4        111/tcp6   rpcbind
|_  100000  3,4        111/udp6   rpcbind
143/tcp   open  imap
|_imap-capabilities: more LOGIN-REFERRALS have IDLE post-login STARTTLS listed
|_      ENABLE capabilities LOGINDISABLEDA0001 Pre-login SASL-IR OK ID LITERAL+
|_      IMAP4rev1
|_ssl-cert: Subject: commonName=debra2.thm.local
|_Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
993/tcp   open  imaps
|_imap-capabilities: LOGIN-REFERRALS more IDLE capabilities OK post-login ENABLE
|_      listed have Pre-login SASL-IR AUTH=PLAINA0001 ID LITERAL+ IMAP4rev1
|_ssl-cert: Subject: commonName=debra2.thm.local
|_Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
995/tcp   open  pop3s
|_pop3-capabilities: PIPELINING SASL(PLAIN) UIDL USER AUTH-RESP-CODE RESP-CODES
|_      CAPA TOP
|_ssl-cert: Subject: commonName=debra2.thm.local
|_Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
MAC Address: 02:DD:7B:88:3D:75 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 20.77 seconds
```

<sup>1</sup>Default NSE Scripts, [Nmap.org](https://nmap.org)

### Service Version Detection for Comparison

```
$ sudo nmap -sV {target_IP}

Starting Nmap 7.80 ( https://nmap.org ) at 2025-03-28 09:21 GMT
Nmap scan report for 10.10.219.233
Host is up (0.0062s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.18.28-1~deb12u2 (Debian Linux)
80/tcp    open  http         nginx 1.22.1
110/tcp   open  pop3         Dovecot pop3d
111/tcp   open  rpcbind      2-4 (RPC #100000)
143/tcp   open  imap         Dovecot imapd
993/tcp   open  ssl/imap     Dovecot imapd
995/tcp   open  ssl/pop3     Dovecot pop3d
MAC Address: 02:DD:7B:88:3D:75 (Unknown)
Service Info: Host: debra2.thm.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.00 seconds
```

### 1.3 OS Detection (-O)

Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses.

#### OS Detection Scan

```
$ sudo nmap -O {target_IP}

Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-28 18:41 +03
Nmap scan report for # {redacted}
Host is up (0.0044s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    filtered ftp
22/tcp    filtered ssh
23/tcp    filtered telnet
53/tcp    open  domain
80/tcp    open  http
8022/tcp  filtered oa-system
MAC Address: # {redacted}
Device type: general purpose
Running: Linux 3.X|4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
       cpe:/o:linux:linux_kernel:5
OS details: Linux 3.10 - 4.11, Linux 5.10 - 5.13
Network Distance: 1 hop
```

## 1.4 Aggressive Scan (-A)

What if you can have both **-O**, **-sV** and some more in one option? That would be **-A**. This option enables OS detection, version scanning, and traceroute, among other things.

### Aggressive Scan

```
$ sudo nmap -A {target_IP}

Starting Nmap 7.80 ( https://nmap.org ) at 2025-03-28 09:21 GMT
Nmap scan report for 10.10.219.233
Host is up (0.0062s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 b9:bc:8f:01:5f:59:23:d3:3a:a2:2d:04:10:e5:04:2d (ECDSA)
|_  256 c0:11:12:52:14:b3:e2:3d:41:bc:3e:94:bb:73:5f:89 (ED25519)
25/tcp    open  smtp      Postfix smtpd
|_smtp-commands: debra2.thm.local, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
CHUNKING,
| ssl-cert: Subject: commonName=debra2.thm.local
| Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
53/tcp    open  domain    ISC BIND 9.18.28-1~deb12u2 (Debian Linux)
| dns-nsid:
|_  bind.version: 9.18.28-1-deb12u2-Debian
80/tcp    open  http      nginx 1.22.1
|_http-title: Welcome to nginx on Debian!
|_http-server-header: nginx/1.22.1
110/tcp   open  pop3      Dovecot pop3d
|_pop3-capabilities: PIPELINING SASL UIDL STLS AUTH-RESP-CODE RESP-CODES CAPA TOP
| ssl-cert: Subject: commonName=debra2.thm.local
| Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
111/tcp   open  rpcbind   2-4 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4       111/tcp     rpcbind
|   100000  2,3,4       111/udp     rpcbind
|   100000  3,4         111/tcp6    rpcbind
|_  100000  3,4         111/udp6    rpcbind
143/tcp   open  imap      Dovecot imapd
|_imap-capabilities: more LOGIN-REFERRALS have IDLE post-login STARTTLS listed ENABLE capabilities LOGINDISABLEDA0001 Pre-
login SASL-IR OK ID LITERAL+ IMAP4rev1
| ssl-cert: Subject: commonName=debra2.thm.local
| Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
993/tcp   open  imaps?
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Help, RPCCheck, SSLSessionReq, TerminalServerCookie:
|     * OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN] Dovecot ready.
|   GenericLines, NULL:
|     * OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN] Dovecot (Debian) ready.
|   GetRequest:
|     * OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN] Dovecot ready.
|     BAD Command received in invalid state.
|   HTTPOptions:
|     * OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN] Dovecot ready.
|     BAD Command received in invalid state.
|   RTSPRequest:
|     * OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN] Dovecot ready.
|     BAD Command received in invalid state.
|_   OPTIONS RTSP/1.0
| ssl-cert: Subject: commonName=debra2.thm.local
| Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
995/tcp   open  pop3s?
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, GenericLines, GetRequest, HTTPOptions, Help, NULL, RPCCheck, RTSPRequest,
SSLSessionReq, TerminalServerCookie:
|_   +OK Dovecot (Debian) ready.
| ssl-cert: Subject: commonName=debra2.thm.local
| Not valid before: 2021-08-10T12:10:58
|_Not valid after: 2031-08-08T12:10:58
MAC Address: 02:DD:7B:88:3D:75 (Unknown)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.4
Network Distance: 1 hop
Service Info: Host: debra2.thm.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   6.16 ms  10.10.219.233

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.77 seconds
```

## 2 Host Discovery Commands

### 2.1 ARP Scan (-PR)

The ARP scan, `-PR`, is what you'd typically use if you're already in the network that you want to scan for live systems. The `-sn` flag here is necessary because it prevents nmap from scanning for open ports after the ARP scan.

#### ARP Scan

```
$ sudo nmap -sn -PR 192.168.100.7/24

Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-29 02:07 +03
Nmap scan report for 192.168.100.1
Host is up (0.0016s latency).
MAC Address: # This will show for all the live hosts, for all the
upcoming Discovery Host scans in this document, but I'll redact
them just in case.
Nmap scan report for 192.168.100.9
Host is up (0.092s latency).
Nmap scan report for 192.168.100.12
Host is up (0.054s latency).
Nmap scan report for 192.168.100.13
Host is up (0.11s latency).
Nmap scan report for 192.168.100.15
Host is up (0.082s latency).
Nmap scan report for 192.168.100.16
Host is up (0.10s latency).
Nmap scan report for 192.168.100.25
Host is up (0.053s latency).
Nmap scan report for 192.168.100.28
Host is up (0.061s latency).
Nmap scan report for 192.168.100.7
Host is up.
Nmap done: 256 IP addresses (9 hosts up) scanned in 28.27 seconds
```

Source	Destination	Protocol	Info
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.1? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.2? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.3? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.4? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.5? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.7? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.8? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.9? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.10? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.11? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.1? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.2? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.3? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.4? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.5? Tell 10.10.210.6
02:ba:eb:d6:18:2b	Broadcast	ARP	Who has 10.10.210.7? Tell 10.10.210.6

Figure 1: Wireshark capture of ARP scan packets, a network different from the aforementioned by the way.

## 2.2 ICMP Scans

We'd rarely send these ICMP packets for the discovery of our system's own subnet, ARP Scans would be better in that case since firewalls tend to block ICMP packets a lot.

### 2.2.1 Echo Scan (-PE)

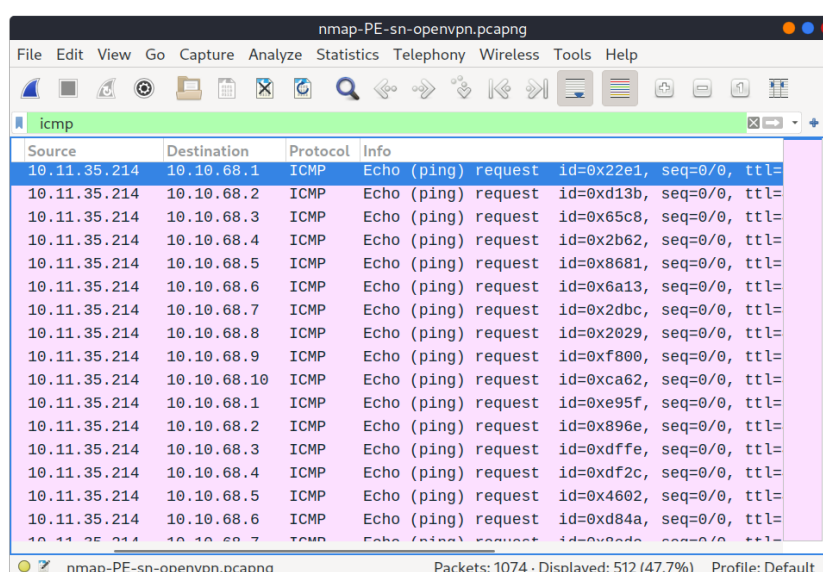
Either way, the -PE scan sends the ICMP Type 8 packets (i.e. same as your ping command).

#### ICMP Echo Scan (-PE)

```
$ sudo nmap -sn -PE 192.168.100.7/24

Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-29 03:06 +03
Nmap scan report for 192.168.100.1
Host is up (0.0039s latency).
Nmap scan report for 192.168.100.12
Host is up (0.33s latency).
Nmap scan report for 192.168.100.13
Host is up (0.74s latency).
Nmap scan report for 192.168.100.15
Host is up (0.072s latency).
Nmap scan report for 192.168.100.16
Host is up (0.32s latency).
Nmap scan report for 192.168.100.25
Host is up (0.11s latency).
Nmap scan report for 192.168.100.28
Host is up (0.73s latency).
Nmap scan report for 192.168.100.7
Host is up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 9.18 seconds
```

Do you notice how it discovered 8 devices, whilst the previous ARP scan discovered 9?



Source	Destination	Protocol	Info
10.11.35.214	10.10.68.1	ICMP	Echo (ping) request id=0xd22e1, seq=0/0, ttl=
10.11.35.214	10.10.68.2	ICMP	Echo (ping) request id=0xd13b, seq=0/0, ttl=
10.11.35.214	10.10.68.3	ICMP	Echo (ping) request id=0x65c8, seq=0/0, ttl=
10.11.35.214	10.10.68.4	ICMP	Echo (ping) request id=0x2b62, seq=0/0, ttl=
10.11.35.214	10.10.68.5	ICMP	Echo (ping) request id=0x8681, seq=0/0, ttl=
10.11.35.214	10.10.68.6	ICMP	Echo (ping) request id=0x6a13, seq=0/0, ttl=
10.11.35.214	10.10.68.7	ICMP	Echo (ping) request id=0x2dbc, seq=0/0, ttl=
10.11.35.214	10.10.68.8	ICMP	Echo (ping) request id=0x2029, seq=0/0, ttl=
10.11.35.214	10.10.68.9	ICMP	Echo (ping) request id=0xf800, seq=0/0, ttl=
10.11.35.214	10.10.68.10	ICMP	Echo (ping) request id=0xca62, seq=0/0, ttl=
10.11.35.214	10.10.68.1	ICMP	Echo (ping) request id=0xe95f, seq=0/0, ttl=
10.11.35.214	10.10.68.2	ICMP	Echo (ping) request id=0x896e, seq=0/0, ttl=
10.11.35.214	10.10.68.3	ICMP	Echo (ping) request id=0xdffe, seq=0/0, ttl=
10.11.35.214	10.10.68.4	ICMP	Echo (ping) request id=0xdf2c, seq=0/0, ttl=
10.11.35.214	10.10.68.5	ICMP	Echo (ping) request id=0x4602, seq=0/0, ttl=
10.11.35.214	10.10.68.6	ICMP	Echo (ping) request id=0xd84a, seq=0/0, ttl=
10.11.35.214	10.10.68.7	ICMP	Echo (ping) request id=0x80dc, seq=0/0, ttl=

Figure 2: Wireshark capture of Echo Request packets, a network different from the aforementioned by the way.

## 2.2.II Timestamp Scan (-PP)

Because ICMP echo requests tend to be blocked, you might also consider ICMP Timestamp (ICMP Type 13) to tell if a system is online.

### ICMP Timestamp Scan (-PP)

```
$ sudo nmap -sn -PP 192.168.100.7/24

Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-29 03:53 +03
Nmap scan report for 192.168.100.1
Host is up (0.0016s latency).
Nmap scan report for 192.168.100.9
Host is up (0.075s latency).
Nmap scan report for 192.168.100.12
Host is up (0.12s latency).
Nmap scan report for 192.168.100.13
Host is up (0.083s latency).
Nmap scan report for 192.168.100.14
Host is up (0.058s latency).
Nmap scan report for 192.168.100.15
Host is up (0.033s latency).
Nmap scan report for 192.168.100.16
Host is up (0.073s latency).
Nmap scan report for 192.168.100.25
Host is up (0.077s latency).
Nmap scan report for 192.168.100.7
Host is up.
Nmap done: 256 IP addresses (9 hosts up) scanned in 14.88 seconds
```

We got "9 hosts up" once again!

Source	Destination	Protocol	Info
10.11.35.214	10.10.68.1	ICMP	Timestamp request id=0xb6bf, seq=0/0, ttl=64
10.11.35.214	10.10.68.2	ICMP	Timestamp request id=0xcad3, seq=0/0, ttl=64
10.11.35.214	10.10.68.3	ICMP	Timestamp request id=0x53ce, seq=0/0, ttl=64
10.11.35.214	10.10.68.4	ICMP	Timestamp request id=0x0149, seq=0/0, ttl=64
10.11.35.214	10.10.68.5	ICMP	Timestamp request id=0x2ead, seq=0/0, ttl=64
10.11.35.214	10.10.68.6	ICMP	Timestamp request id=0x3ce5, seq=0/0, ttl=64
10.11.35.214	10.10.68.7	ICMP	Timestamp request id=0x5de2, seq=0/0, ttl=64
10.11.35.214	10.10.68.8	ICMP	Timestamp request id=0x884d, seq=0/0, ttl=64
10.11.35.214	10.10.68.9	ICMP	Timestamp request id=0xbf35, seq=0/0, ttl=64
10.11.35.214	10.10.68.10	ICMP	Timestamp request id=0x6b44, seq=0/0, ttl=64
10.11.35.214	10.10.68.1	ICMP	Timestamp request id=0x1a28, seq=0/0, ttl=64
10.11.35.214	10.10.68.2	ICMP	Timestamp request id=0x8586, seq=0/0, ttl=64
10.11.35.214	10.10.68.3	ICMP	Timestamp request id=0xacce, seq=0/0, ttl=64
10.11.35.214	10.10.68.4	ICMP	Timestamp request id=0x0cfa, seq=0/0, ttl=64
10.11.35.214	10.10.68.5	ICMP	Timestamp request id=0xa39f, seq=0/0, ttl=64
10.11.35.214	10.10.68.6	ICMP	Timestamp request id=0x2279, seq=0/0, ttl=64
10.11.35.214	10.10.68.7	ICMP	Timestamp request id=0x00cf, seq=0/0, ttl=64

Figure 3: Wireshark capture of Timestamp Request packets, a network different from the aforementioned by the way.