

Universidade Tecnológica Federal do Paraná – Câmpus Pato Branco
DAINF – Departamento Acadêmico de Informática

Análise de Segurança de Conversores Serial-Ethernet e Microcontroladores Tibbo

Willian Americano Lopes
walopes23@gmail.com

Análise de Segurança de Conversores Serial-Ethernet e Microcontroladores Tibbo

Ildomar Gomes de Carvalho Junior
Rafael R. Obelheiro

- Automatizar processos industriais
- Constante monitoramento de dados

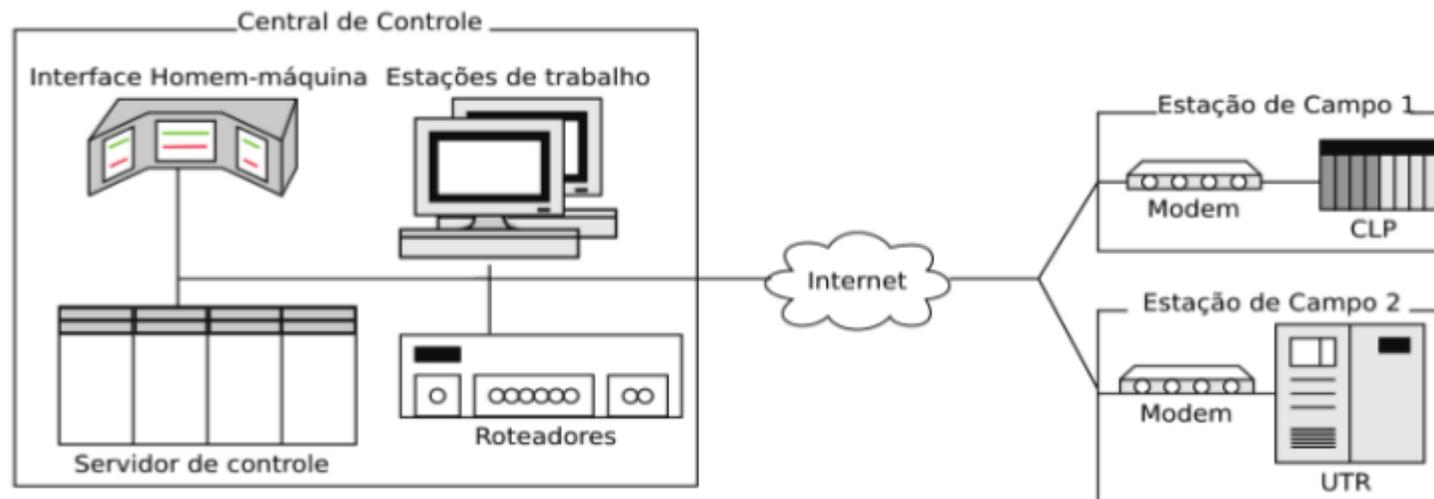


Fig. 1: Exemplo de SCI

TIBBO EM1206



- ASIC T1000 88 MHz
- 512K ou 1024K de Flash
- 2KB EEPROM
- Baud rate máximo de 921.600

Key Features



10/100 Base-T Ethernet port



High-performance CPU (T1000)



Mates with the [Rj203](#) jack/magnetics front end



4 UARTs on 8 I/O lines, flexible mapping options



Optional Wi-Fi connectivity (with the [GA1000](#) add-on)



3.3V power

OBJETIVO

- Buscar vulnerabilidades que violem as propriedades de segurança (integridade, confidencialidade e disponibilidade) [ISO/IEC 2008]
- Injeção de ataques

- Técnica experimental
- Avaliar as propriedades de confiança no funcionamento (*dependability*)
- Segurança: Injeção de ataques

Vantagens:

- Deficiências
- Comportamento

- Avaliar a robustez da pilha TCP/IP
- Tráfego malicioso via rede
- Erros na codificação da pilha TCP/IP
- *Fuzzing*

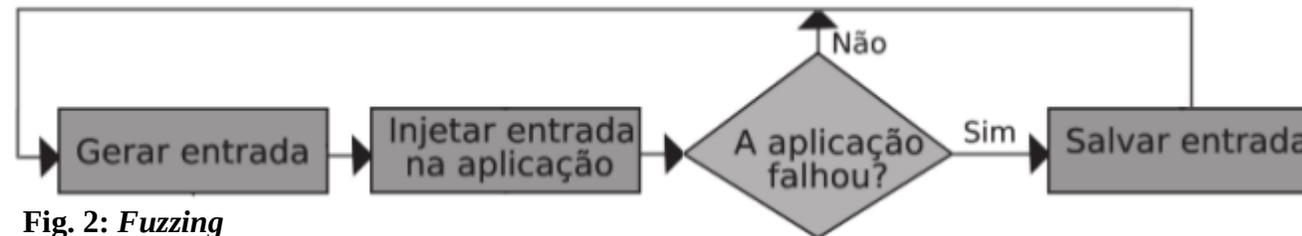


Fig. 2: *Fuzzing*

Metodologia proposta por Pothamsetty e Balinsky (2003), analistas da Cisco.

Reconhecimento

- Identificar remotamente o SO e descobrir serviços de redes ativos

Análise por camadas

- Injeção dos pacotes e pesquisa de vulnerabilidades

Nmap

- Varredura de portas e do SO

```
mandar@mandar: ~  
mandar@mandar:~$ nmap -sV 10.10.6.204  
  
Starting Nmap 5.21 ( http://nmap.org ) at 2013-07-14 14:23 IST  
Nmap scan report for 10.10.6.204  
Host is up (0.00029s latency).  
Not shown: 996 closed ports  
PORT      STATE SERVICE          VERSION  
21/tcp    open  ftp              vsftpd 2.3.5  
22/tcp    open  ssh              OpenSSH 5.9p1 Debian 5ubuntu1 (protocol 2.0)  
23/tcp    open  telnet           Linux telnetd  
902/tcp   open  ssl/vmware-auth VMware Authentication Daemon 1.10 (Uses VNC, SOAP)  
Service Info: OSs: Unix, Linux  
  
Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 0.64 seconds  
mandar@mandar:~$
```

Fig. 3: NMap

Nessus

- Scanner de vulnerabilidade

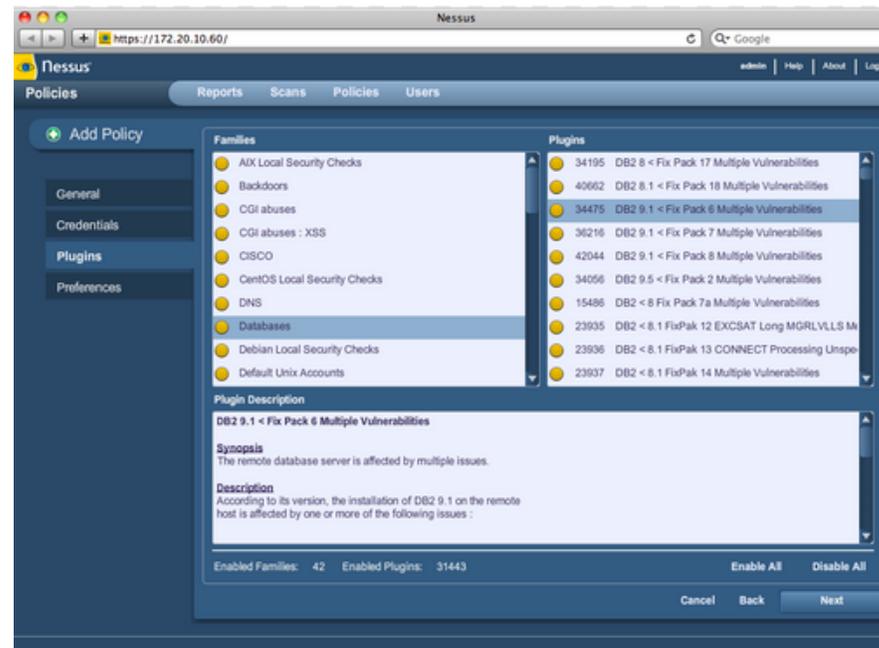


Fig. 4: Teanable Nessus

IP Stack Integrity Checker

- Conjunto de ferramentas para executar o *fuzzing* na pilha TCP/IP
- Gera pacotes pseudo-aleatórios
- ESIC: gera *frames Ethernet*
- ISIC: gera pacotes UDP
- UDPSIC: gera datagramas UDP
- TCPSIC: gera pacotes TCP

Bruteforce Exploit Detector (BED)

- Testes *fuzzing* na aplicação WEB

```
root@kali:~/Script/bedfuzzer# perl bed.pl
BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de )

Usage:
bed.pl -s <plugin> -t <target> -p <port> -o <timeout> [ depends on the plugin ]

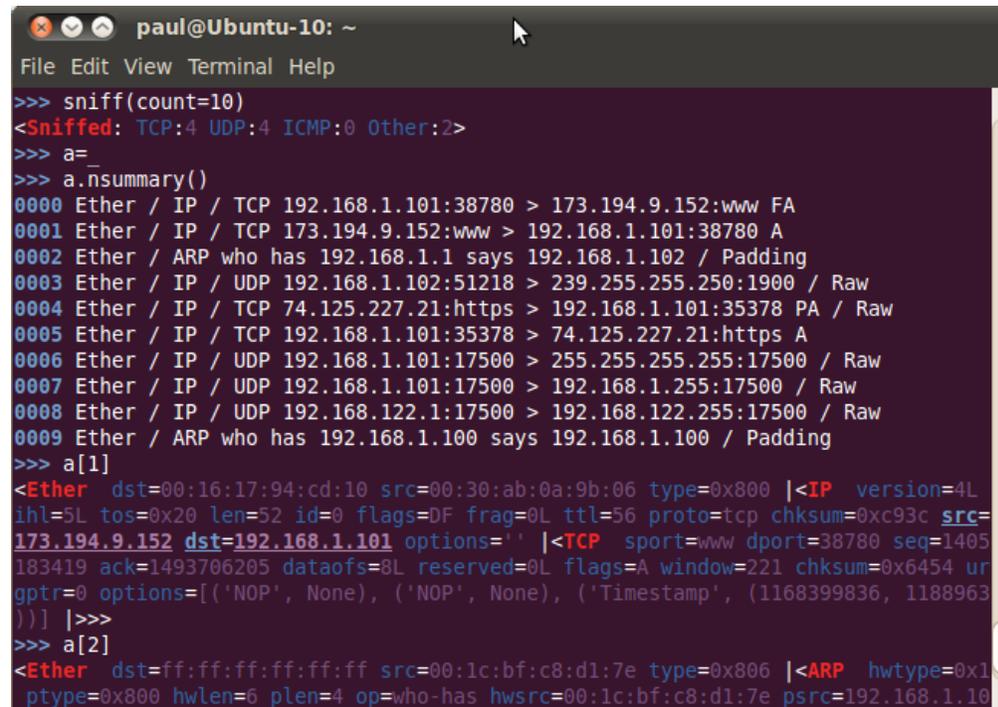
<plugin> = FTP/SMTP/POP/HTTP/IRC/IMAP/PJL/LPD/FINGER/SOCKS4/SOCKS5
<target> = Host to check (default: localhost)
<port>   = Port to connect to (default: standard port)
<timeout> = seconds to wait after each test (default: 2 seconds)
use "bed.pl -s <plugin>" to obtain the parameters you need for the plugin.

Only -s is a mandatory switch.
root@kali:~/Script/bedfuzzer#
```

Fig. 5: BED

Scapy

- *Scripts* para reproduzir as vulnerabilidades encontradas



```
paul@Ubuntu-10: ~
File Edit View Terminal Help
>>> sniff(count=10)
<Sniffed: TCP:4 UDP:4 ICMP:0 Other:2>
>>> a=
>>> a.nsummary()
0000 Ether / IP / TCP 192.168.1.101:38780 > 173.194.9.152:www FA
0001 Ether / IP / TCP 173.194.9.152:www > 192.168.1.101:38780 A
0002 Ether / ARP who has 192.168.1.1 says 192.168.1.102 / Padding
0003 Ether / IP / UDP 192.168.1.102:51218 > 239.255.255.250:1900 / Raw
0004 Ether / IP / TCP 74.125.227.21:https > 192.168.1.101:35378 PA / Raw
0005 Ether / IP / TCP 192.168.1.101:35378 > 74.125.227.21:https A
0006 Ether / IP / UDP 192.168.1.101:17500 > 255.255.255.255:17500 / Raw
0007 Ether / IP / UDP 192.168.1.101:17500 > 192.168.1.255:17500 / Raw
0008 Ether / IP / UDP 192.168.122.1:17500 > 192.168.122.255:17500 / Raw
0009 Ether / ARP who has 192.168.1.100 says 192.168.1.100 / Padding
>>> a[1]
<Ether dst=00:16:17:94:cd:10 src=00:30:ab:0a:9b:06 type=0x800 |<IP version=4L
ihl=5L tos=0x20 len=52 id=0 flags=DF frag=0L ttl=56 proto=tcp checksum=0xc93c src=
173.194.9.152 dst=192.168.1.101 options='' |<TCP sport=www dport=38780 seq=1405
183419 ack=1493706205 dataofs=8L reserved=0L flags=A window=221 checksum=0x6454 ur
gptr=0 options=[('NOP', None), ('NOP', None), ('Timestamp', (1168399836, 1188963
))] |>>>
>>> a[2]
<Ether dst=ff:ff:ff:ff:ff:ff src=00:1c:bf:c8:d1:7e type=0x806 |<ARP hwtype=0x1
ptype=0x800 hwlen=6 plen=4 op=who-has hwsrc=00:1c:bf:c8:d1:7e psrc=192.168.1.10
```

Fig. 6: Scapy

Wireshark

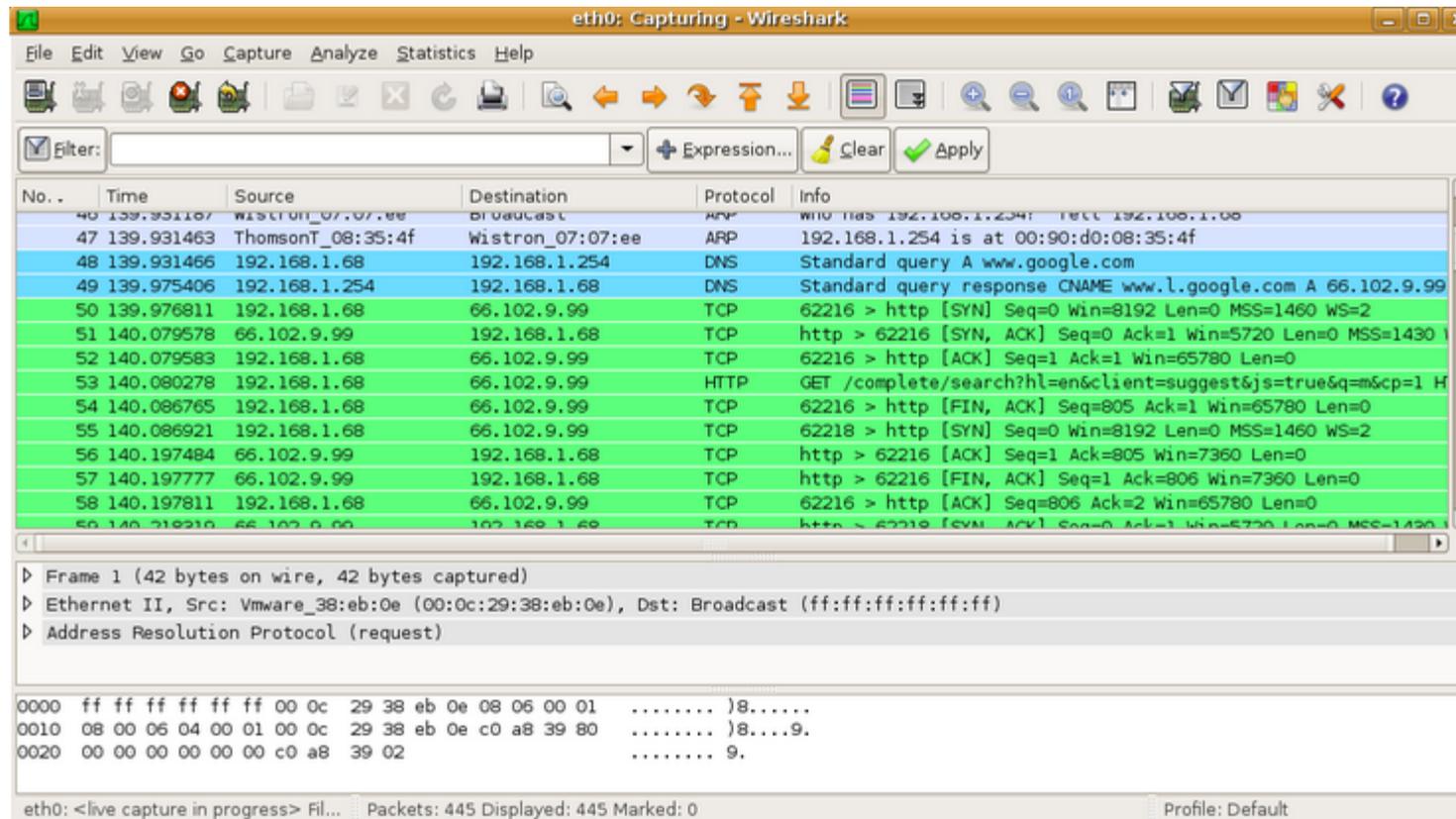


Fig. 7: Wireshark

Reconhecimento

Análise por camadas

- Ethernet
- IP/UDP
- TCP
- Aplicação

- Identificação do SO e da pilha TCP/IP
- SO: ?
SW: NMap
- 3 portas abertas:

Porta 23: Não roda o TELNET → aplicação de gerenciamento do módulo

Porta 80: HTTP e gerenciamento

Porta 1001: conversão serial-*Ethernet*

Testes:

1. Aplicação do *fuzzing* nos campos do cabeçalho *Ethernet*

SW: ESIC

Não demonstrou vulnerabilidade ao *fuzzing*

2. SW: Nessus

Vulnerabilidade: Etherleak

RFC 1042: tamanho mínimo para os quadros *Ethernet*

Dados vem de *buffers*



No.	Time	Source	Destination	Protocol	Info
758	332.036434	Azurewav_2a:5a:dc	TibboTec_50:57:b9	ARP	Who has 192.168.25.5? Tell 192.168.25.4
759	332.042817	TibboTec_50:57:b9	Azurewav_2a:5a:dc	ARP	192.168.25.5 is at 00:24:77:50:57:b9
760	332.036541	Azurewav_2a:5a:dc	TibboTec_50:57:b9	ARP	Who has 192.168.25.5? Tell 192.168.25.4

▶ Frame 759: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
 ▶ Ethernet II, Src: TibboTec_50:57:b9 (00:24:77:50:57:b9), Dst: Azurewav_2a:5a:dc (1c:4b:d6:2a:5a:dc)
 ▶ Address Resolution Protocol (reply)

```

0000  1c 4b d6 2a 5a dc 00 24 77 50 57 b9 08 06 00 01  .K.*Z..$ wPW.....
0010  08 00 06 04 00 02 00 24 77 50 57 b9 c0 a8 19 05  .....$ wPW.....
0020  1c 4b d6 2a 5a dc c0 a8 19 04 61 6c 75 65 3d 22  .K.*Z... ..alue="
0030  74 73 74 22 20 74 61 62 69 6e 64 65          tst" tab inde
  
```

Fig. 8: Requisição ARP

- Utilizados ISIC e UDPSIC, separadamente
- Não foi vulnerável ao *fuzzing*
- Fluxo de dados causou *flooding*

Três testes realizados

1. Utilizando o SW TCPSIC

Vulnerável ao *flooding* mas não ao *fuzzing*

2. Eficiência do gerador de números iniciais de sequência

Aleatoriedade

Nmap: Aleatoriedade fraca

Nessus: Aleatoriedade forte

Análise manual utilizando *script* no Scapy



Envia segmentos SYN para o EM1206 e imprime os números de sequência usado na resposta SYN-ACK

Incrementa o valor em 64000 a cada 500ms

Viola a RFC 6528

3. *Reset Spoofing*



Possível encerrar conexão enviando pacotes RST ilegítimos

Script pelo Scapy

Enviou pacotes com as *flags* SYN e RST para uma conexão estabelecida, utilizando n^a de sequência e ACK errados

TCP

No.	Time	Source	Destination	Protocol	Length	Info
102	5.042847000	192.168.25.5	192.168.25.2	TCP	60	customs > 56988 [ACK] Seq=961344001 Ack=1497917848 Win=5
134	6.470469000	192.168.25.2	192.168.25.5	TCP	58	56988 > customs [PSH, ACK] Seq=1497917848 Ack=961344001
135	6.470938000	192.168.25.5	192.168.25.2	TCP	60	customs > 56988 [ACK] Seq=961344001 Ack=1497917852 Win=5
227	13.244406000	192.168.25.2	192.168.25.5	TCP	54	30000 > customs [SYN, RST] Seq=0 Win=8192 Len=0
274	16.295299000	192.168.25.2	192.168.25.5	TCP	58	56988 > customs [PSH, ACK] Seq=1497917852 Ack=961344001
275	16.295766000	192.168.25.5	192.168.25.2	TCP	60	customs > 56988 [RST] Seq=961344001 Win=14600 Len=0

- ▶ Frame 227: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
- ▶ Ethernet II, Src: 9c:2a:70:89:04:89 (9c:2a:70:89:04:89), Dst: TibboTec_50:9b:7f (00:24:77:50:9b:7f)
- ▶ Internet Protocol Version 4, Src: 192.168.25.2 (192.168.25.2), Dst: 192.168.25.5 (192.168.25.5)
- ▶ Transmission Control Protocol, Src Port: 30000 (30000), Dst Port: customs (1001), Seq: 0, Len: 0

Fig. 9: Reset Spoofing

4 Vulnerabilidades

1. Autocompletar campo senha



Não desabilitar o autocomplemento oferece um risco de perda de confidencialidade

2. Comportamento do servidor diante de ataques *fuzzing*



SW BED

flooding: GET e POST

3. Roubo de *cookies* de uma seção



Conversor dá um *cookie* para identificar que está autenticado

Script pelo Scapy

Enviou pacotes com as *flags* SYN e RST para uma conexão estabelecida, utilizando n^a de sequência e ACK errados

4. Envio não criptografado da senha do conversor



Três maneiras de acessar o EM1206:

- Página WEB embarcada
- DS Manager
- Aplicação que roda na porta 23

Enviam a senha de rede de forma aberta

DS MANAGER → envia a senha por *broadcast*

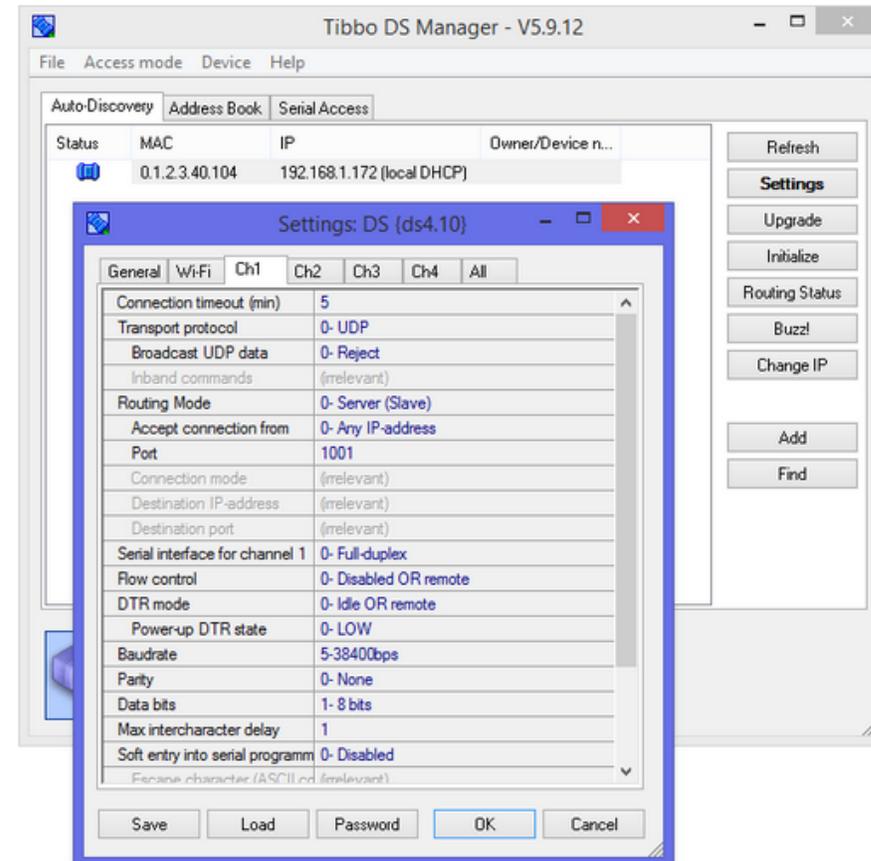


Fig. 10: DS Manager

Tabela 1. Vulnerabilidades encontradas no módulo EM1206 da Tibbo.

Camada	Problemas Encontrados
<i>Ethernet</i>	Etherleak
IP	<i>Flooding</i>
UDP	<i>Flooding</i>
TCP	<i>Flooding</i> , Aleatoriedade Fraca do Número Inicial de Sequência e <i>Reset Spoofing</i>
HTTP	Autocompletar Senha, <i>Flooding</i> , Roubo de <i>Cookie</i> e Transmissão de Senha Não Criptografada

- Nível de segurança do Tibbo EM1206 é baixo
- Sugere metodologias para ataques *fuzzing*
- Sugere comparação com outros dispositivos
- Sugere métricas para medir a robustez de um dispositivo embarcado diante *flooding*

-
- [1] <http://tibbo.com/store/soi/em1206.html?page=products/modules/x20x/em1206>
- [2] Pothamsetty, V and Balinsky, A (2003). **A structured and practical methodology for security evaluation of a IP based stack** (version 0.2).

Perguntas?

Sugestões?

Obrigado!