C10-secu - COURS LINUX - MTN

Guillaume ASTIER

26/02/16

Table des matières

La sécurité

La SSI (Sécurité des Systèmes d'Information), c'est l'ensemble des mesures prises pour assurer la **confidentialité**, l'**intégrité** et l' **accessibilité** des données.

confidentialité : Garantir que les données ne sont accessibles qu'aux personnes autorisées ;

intégrité : Garantir que les données n'ont pas été modifiées par un tiers non autorisé ;

disponibilité ou accessibilité : Garantir l'accès à un service ou à des ressources.

Un sinistre (incendie, dégât des eaux)

Un utilisateur du système dont le comportement favorise le danger ("rm -fr *") ;

Personne ou organisation malveillante (DDoS, installation de malware, virus, rootkit, fuite d'information, etc.).

Articles

Article (attaques IT)


Le service DNS Dyn attaqué : Reddit, Spotify, Twitter... trébuchent

Un fournisseur DNS pour des effets accrus

Figure 1: Accessibilité

Article (attaques IT)

Article (attaques IT)

Figure 3: Confidentialité

Article (APT : Advanced Persistent Threat)

![ref6]

Article (APT : Advanced Persistent Threat)

![ref6]

Article (APT : Advanced Persistent Threat)

![ref6]

Article (APT : Advanced Persistent Threat)

![ref7]

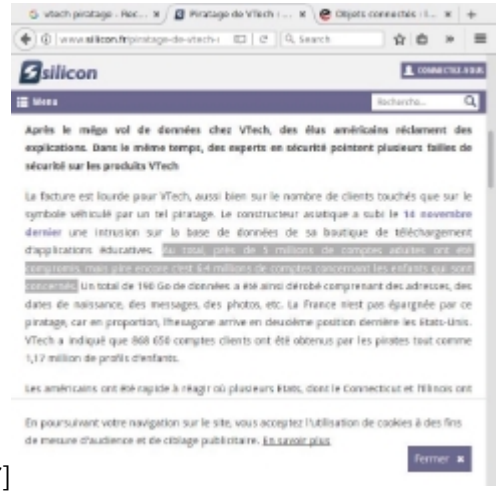
Article (attaques IoT)

![ref7]

Article (attaques IoT)

![ref7]

Article (attaques IoT)



![ref7]

Article (attaques IoT)

![ref7]

Article (attaques IoT)

![ref7]

Corruption des données volontaire/involontaire

Corruption de mémoire (buffer overflow, format string, null pointer dereference, etc.) ;

Mauvaise conception d'une application (PATH, file race condition, etc.) ;

Problème de permissions sur des fichiers "sensibles" (clés ssh, DB, backup système, etc.) ;

Négligence de l'administrateur (confiances ssh, sudo trop permissif, etc.).

Corruption lié au développement

Corruption de mémoire (service réseau, stack IP kernel, etc.) ; Mauvaise conception d'une application (Hearbleed, Shellshock, etc.) ;

Vulnérabilité Web (iSQL, XSS, XSRF, injections, etc.) ;

Brute force (login/password, user enumeration, etc.) ;

Social engineering (fishing, spam, etc.) ;

MITM (empoisonnement de table ARP, DHCP Exhaustion, etc.) ;

(D)DoS (SYN flood, ping of death, etc.).

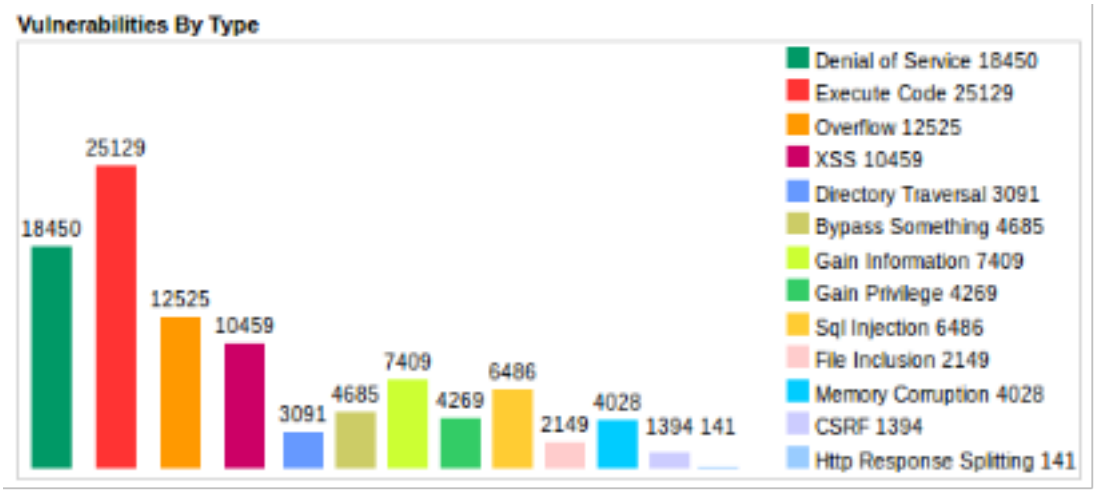


Figure 4: Source: http://www.cvedetails.com

Figure 5: Source: http://www.cvedetails.com

Exemple

Exemple 1 (Buffer overflow)

```
1 void copy(char \*input) { char buf[8];
2 strcpy(buf, input); [...]
3 }
4 int main(int argc, char \*\*argv) {
5 copy(argv[1]);
6 }
```

Exemple 2 (Buffer overflow)

```
1 int main() {
2 char buf[1024]; gets(buf);
3 }
```

Exemple 3 (Integer Overflow) Extrait de OpenSSH 3.3 :

```
1 int nresp;
2 nresp = packet\_get\_int();
3 if (nresp > 0) {
4 response = xmalloc(nresp\*sizeof(char\*)); for (i = 0; i < nresp; i++)
5 response[i] = packet\_get\_string(NULL); }
```

Exemple 3 (Integer Overflow)

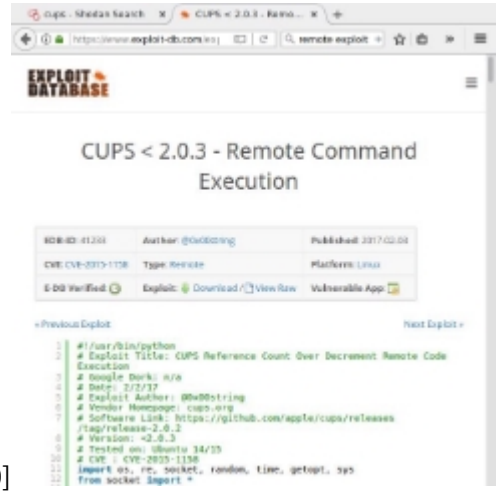
```
1 Si nresp = 1073741824,
2 nresp \* sizeof(char\*) = 4294967296
3 En binaire : 1 00000000 00000000 00000000 00000000 L  opération xmalloc
     (0) est exécutée !
```

Exemple 4 (File race condition) :

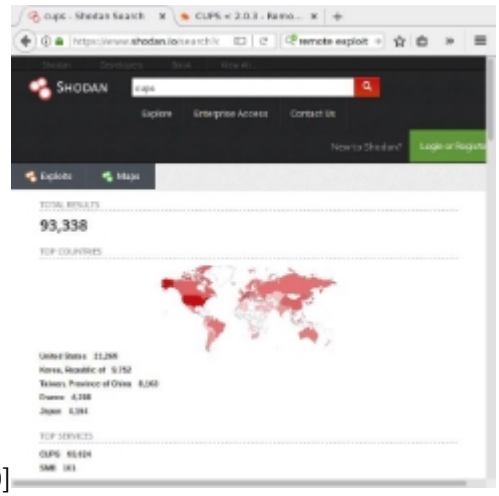
```
$ cat -n /etc/init.d/x11-common [...] SOCKET\_DIR="/tmp/.X11-unix"set
 -e if [ -e $SOCKET\_DIR ] && [ ! -d $SOCKET\_DIR ]; then mv $SOCKET\
_DIR $SOCKET\_DIR.$$ fi mkdir -p $SOCKET\_DIR chown root:root $SOCKET
\_DIR chmod 1777 $SOCKET\_DIR:
```

Script kiddie

Approche “script kiddie”![ref10]



Approche “script kiddie”![ref10]



Surface d’attaque

Le but est de supprimer tous les éléments logiciels superflus pour réduire la surface d’attaque (tout ce qui est inutile aux utilisateurs légitimes peut servir à un attaquant).

Ce principe de minimisation de la surface d’attaque s’applique :

à la couche utilisateur (“userland”), au noyau (“kernel-land”).

Type d'attaque

Userland

Le Userland correspond aux action qui peuvent être faite sur un système déjà en fonctionnement

Désinstaller tous les programmes qui ne sont pas utiles Désactiver les services inutiles

qui sont en écoute sur le réseau qui s'exécutent sous le compte root

Limiter l'ensemble des programmes privilégiés

les programmes setuid (préférer les capacités de fichiers) en arrière plan par un cron

Inventorier les comptes, les groupes (supprimer les obsolescences)

Kernelland

Le Kernel-land correspond aux modification précédent l'installation d'un système

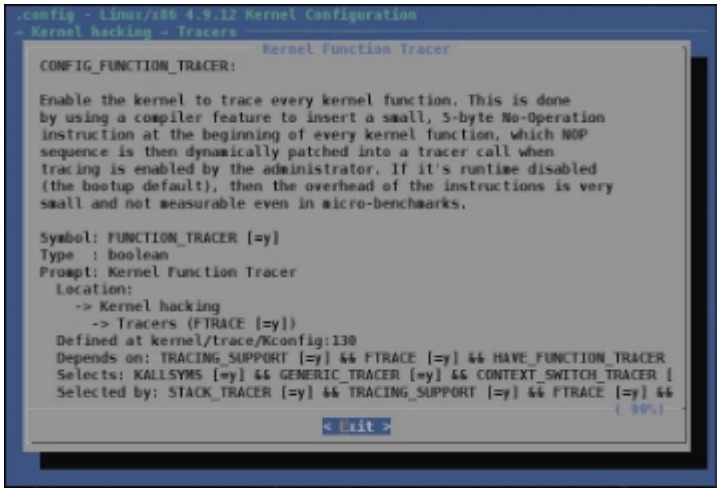
Mettre en œuvre un noyau sur mesure

vis-à-vis des périphériques matériels (support bluetooth, wifi, joystick, etc. ?)

des fonctions logiques dont aura besoin le système (CONFIG_FTRACE ?)

Préférer un noyau non modulaire

ou n'autoriser que le chargement de modules signés



Type d'instances

Poste de travail vs. serveur

Besoin d'un environnement graphique sur un serveur ? Besoin d'un compilateur, déboguer sur un serveur ? Besoin d'un serveur SSH sur un poste de travail ? Noyau sur-mesure pour un poste de travail ?

etc.

Gestion des données

Partitionnement rationnel

Créer une partition distincte pour tout répertoire susceptible d'être rempli à la suite d'actions extérieures (journaux, "spool" d'impression, etc.)

Appliquer des options de montages nécessaires et suffisantes (ro , nodev, noexec et nosuid)

Analyse des donées

Analyse fine des droits et permissions

Supprimer les fichiers orphelins

Ne jamais utiliser "chmod 777 "

Eviter "chmod u+s ", privilégier lesfile capabilities

Cloisonnement

Le principe de cloisonnement permet d'isoler un service du reste de la machine. Limite l'impact d'une attaque réussie.

lxc-create -t debian -n vm1 lxc-start -n vm1 -d lxc-console -n vm1 /etc/init.d/apache2 start

LXC repose sur les fonctionnalités cgroups et namespaces du noyau Linux et permet de faire fonctionner des environnements Linux isolés les uns des autres.

lxc-create -t debian -n vm2

cp malware /var/lib/lxc/vm2/rootfs/tmp/ lxc-execute -n vm2 – strace /tmp/malware

Base firewall

Iptables permet d'interdire/autoriser des communications réseaux. Le filtrage est mis en place sur trois chaînes (INPUT, OUTPUTet FORWARD) selon les critères suivants:

L'interface réseau

Adresse IP source, IP dst

Port(s) source, port(s) destination Protocole (TCP, UDP)

Par exemple, pour n'autoriser que l'accès au service SSH :

- iptables -P INPUT DROP
- iptables -P OUTPUT DROP
- iptables -A INPUT -p tcp –dport 22 -j ACCEPT
- iptables -A OUTPU

Exemple 1 :

Désactivation d'un service réseau Et mise en place d'un pare-feu

Exemple 2 :

Sécurisation d'un service

Et cloisonnement du service

Supprevision

Permet de garder un historique et d'envoyer des alertes (mail, SMS).

Les ressources systèmes (SNMP, Cacti)

Les activités réseaux (mrtg , ntop)

Les journaux systèmes (/var/log/* , syslog , logwatch) On peut utiliser des détecteurs d'intrusion IDS (snort , suricata)

Audit

Auditer régulièrement

Les processus en cours d'exécution (ps, top , etc.) Les ports réseaux ouverts (nmap, netstat)

Les modules kernel chargés (lsmod)

La dureté des mots de passe du système (john) L'intégrité des fichiers systèmes (AIDE, Samhain)

Update

Faire des mises à jour régulièrement

Chiffrer les données (sauvegardes, fichiers sensibles, etc.) Utilisation de contrôle d'accès obligatoires (Tomoyo)

Utiliser un patch de durcissement noyau (PaX, Grsecurity) Penser à la sécurité physique

Appliquer les guides de sécurisation

Guide de sécurisation

L'ANSSI (https://www.ssi.gouv.fr), propose via leur site des guides pour sécuriser des systèmes :

Recommandations de sécurité relatives à un système GNU/Linux

Recommandations pour le déploiement sécurisé du navigateur Google Chrome sous Windows

Guide des bonnes pratiques de l'informatique

Sécurité des mots de passe

Sécuriser son ordiphone

Partir en mission avec son téléphone, sa tablette ou son ordinateur portable

Sécuriser les accès Wi-Fi