Oignon (e→i) : Physique → Réseau → Proto → Host → App
CIA : Confidentiality - Integrity - Availability
Kill Chain (RUAG) : Reconnaissance - Exploit - Post Exploit

RootKit
Movement latéraux : Machine → Machine meme privilège

Exemple d'intrusion logiciel
- **Heartbleed**(buffer overflow)
- **Log4Shell**
Log4 = lib de log JAVA tu pouvais balancer un URL à toi sur les log ce qui permettais une injection de code JAVA

Attaque Web
Injection SQL : ' OR 1=1
XSRF : XSS stocker sur un forume

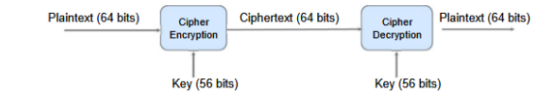
OWASP :
1) Broken Access controle (Peu accédé à des ressources non authorisé)
2) Cryptographic Failures
3) Injection
4) Insecure Design
5) Security Misconfiguration
6) Vulnerable / Outdated Component
7) Identification & Auth Failures
8) Software & Data intergrity Failures
9) Security logging & Monitoring Failures
10) Server-Side Request forgery

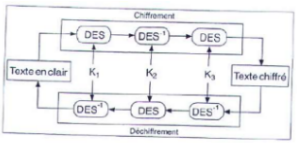
Outils :

DES : Data Encryption Standard
Faibles : Recherche exhaustive de Clès 56 bits de clé = 2^56 = 7 * 10^16
Cpocabana (2006-2008) 48 millard de dec par seconde → ~= 10 jours
Triple-DES
3Key-TDES : 3 key de 56 bits | aussi secu que **2Key-TDES** : 2 key de 56 bits | plus courant (surtout en embarqué, HW / SW)

AES (128 bits) successe TDES (trop lent sur SW)

« Block cipher »
- Dédié pour les implémentations hardware
- Blocs de 64 bits
- Clés de 56 bits (parfois 64 bits et parités)





Crypto
Verman cipher :
Enc : C = {M | Message de x bits} xor {K |clé de x bits}
Dec : M = C xor K
Bute : Confidentialité - Authenticité - Intégrité
Cipher : Algo de chiffr/déchiffr, en général avec clé serète
PK : Algo de chiffr/déchiffr à clé publique
NIST : National Institute of Standard and Technology
Block cipher : encrypt par bloc de x bits | Haut débit
IDEA-NXT
Stream cipher : encrypt bit by bit | très haut debit
Chacha20

Cloud computing

Not to be confused with Cloud Computing (horse).

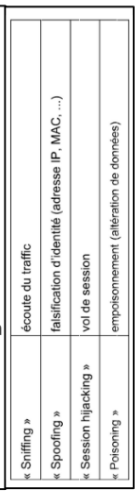
Exemple de Virus

	LM hash	NTLM hash
Windows 9x	X	
Windows ME	X	
Windows NT	retro	X
Windows 2000	retro	X
Windows server 2000	retro	X
Windows XP	retro	X
Windows server 2003	retro	X
Windows Vista		X
Windows server 2008		X
Windows 7, 8, 10, 11		X

Mot de passe dans les os
Windows :
c:\Windows\system32\config\SAM (128 bits par Syskey)
LM : max 14 caractères, pas vraiment un hash (dont care about case)
NTLM : MD4

Linux :
/etc/shadow (128 bits)
stocke :
Username:$**X**$HASH:LastPasswordChange:MinDayBeforChange:MaxDay...
X:
1 = MD5
2a et 2y = Blowfish
5 = SHA-256
6 = SHA-512
DES ou MD5

Attaque Reseau
IMCP : Protocole de Message de Contrôle Internet
Scanning : trouver les machine vivante, trouver les port ouvert, trouver les services qui tourne
Port scaning : requet TCP(UDP) (la machin renvoie par defaut un message "destination port unreacable" (ICMP)).
TCP logic : si hit → server send SYN-ACK message
si miss → message RST-ACK (close)
si ça hit mais on fait rien, TCP envoyer un ACK au server et crée un log, mais si on envoie un RST ça close la connection est c'est all good
DOS(Denial of Service)
SYN Flooding : bombarder des packet SYN pour tuer la RAM
Smurf : ping tous le monde sauf que l'on dit que le ping vien de la victime, du coup il se prend toute les reponces d'un coup
DDOS : tu connais chill



Virus
Govware = defence
Cyber war = Attaque (Stuxnet, Regin, Gauss)

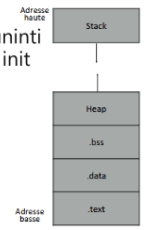
Furtivité :
Se cache, EX dans la routine de l'OS

Polymorphisme :
Virus change à chaque infection

Email Forger
Protocol : SMTP (envoie d'email), POP et IMAP (get email to read)
Aucun des 3 protocol checke la confidentialité
En gros, tu peux just dire que tu es X et y a rien pour bloquer de base
En regle general, faut pas faire confiance mais le haut du text-source peut être pris au serieux (par ce que c'est les info reçus par le dernier Server)

John Draper (Capt. Crunch) : sifflet : 60's.
Steve Jobs et Steve Wozniak : bluebox (sifflet en mieux) : 70's.
Kévin Mitnick : pen. centraux AT&T, interception détournement : 3 mois : 1981.
Idem : introduction dans ARPAnet, défense américaine, 6 mois Prison: 1983.
1982 : Elk Cloner : premier virus dans la nature, infection secteur boot, poème chaque 50 boots.
1986 : Brain : premier virus sur MS-DOS, infection serveur boot, pub.
1988 : 1er ver, Robert Tappan Morris 1994 : 1er casse, Vladimir Levin, 10.7 Mio$.
1994 : Kévin traqué par Shimomura : arrêté, 5 ans prison.
1999 : Jonathan James, NASA cassage mdp, pas prison (17ans)
1999 : ver Melissa, propag. via Outlook, saturation, insertion code viral dans
doc. Word.
2000 : iloveyou : mail, ver, pièce avec extension masquée (VBS).
2000 : QAZ : code source WinME volé => backdoor déguisée en NotePad.
2000 : mafiaboy DoS, attrapé sur IRC.
2002 : BugBear : keylogger / propa. via mail.
2003 : Blaster : dénis de service sur Microsoft / mal prog., redémarrage PC.
2003 : SQL Slammer : DoS.

Stack
EIP : Addr de retour de la fonction
EBP : début d'encadrement du stack
ESP : stack pointer
.bss = global data uninti
.data = global data init
.text = exe code



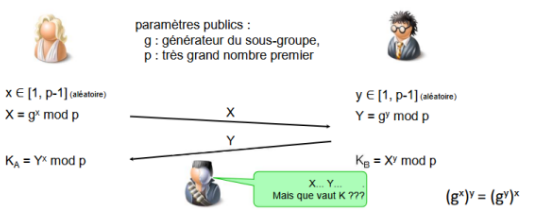
	Temps préparation	Temps cassage	Taille mémoire	Probabilité succès	Sel
Dictionnaire	0	?	Faible (dictionnaire)	?	Sans importance
Heuristique	0	?	Faible (dictionnaire)	?	Sans importance
Force brute	0	O(N)	0	100%	Sans importance
Pré-calculation complète	O(N)	0	O(N)	100%	Plus difficile
Hellman	Long	faible	variable	variable 50-95%	Plus difficile
Rainbow tables	Long	faible	variable	variable 50-95%	Plus difficile

Linux folder
/bin : exe de base (ls, cat, etc...)
/usr/bin : exe de plus (grep,...)
/sbin : exe admin de base
/lib : lib utiliser par /bin et /sbin
/etc : ficher de config
/dev : périphériques
/tmp : fichier temps
/etc/resolv.conf : DNS cache

Mot de passe
M. Hellman :
Password → hash → fnc qui crée un psw depuis le hash (fnc de reduction) → new password → {repeate}
(pas ouf parceque possiblement des doublons)

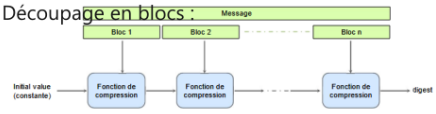
Rainbow tables (de Phillipe Oechslin) :
meme chose que Hellman mais la fnc change à chaque étape pour eviter les doublons

Diffie-Hellman
Protocole :

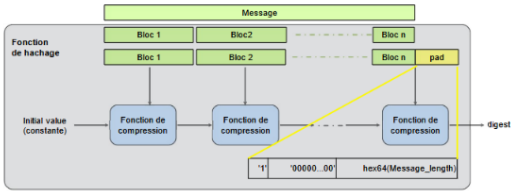


Fonction de hashage
MD (Message Digest) : par Ronald Rivest, 128 bits, cassé (de MD2-MD5)
SHA-(0/1) (Secure Hash Algorithm) : par NSA/NIST, 160 bits, cassé
SHA-x : 224 - 512 bits, Recommandé

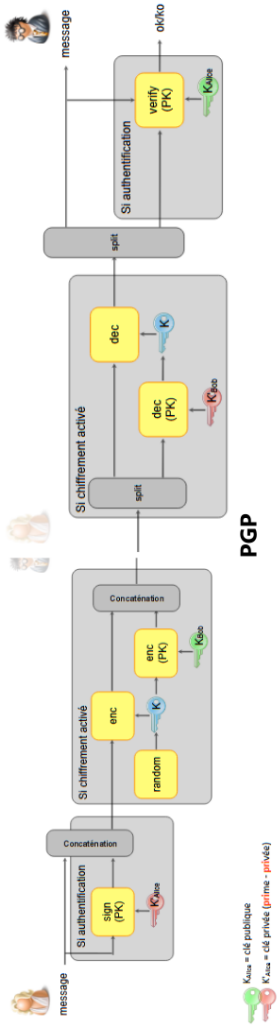
Découpage en blocs :



Padding :
1) 1 bit (à 1) est ajouter à la fin du message
2) puis on ajoute autant de 0 qu'il faut pour que le message inférieure de 64 bits à un multiple de 512
3) les bits restant sont remplie avec 64 bits représentant la longeur du message de base modulo 2^64



- Il y a TOUJOURS un «padding» !
  Même si le message tombe parfaitement sur des blocs complets.
- On voit que pad nécessite au minimum 66 bits.
  Si pas assez d'espace, il est nécessaire d'ajouter un bloc pour insérer le «padding».



RSA
- Chaque user à une (Ki)private key et une (Ku)public key
- (Ki) et (Ku) sont dépent mathématiquement
- Pour envoyer un message à MARGIT, on enc le message avec la (Ki) de MARGIT
- Key de 512 jusqu'à 4096 bits possibles

ℓ : la taille du modulus RSA

p, q : deux nombres premiers aléatoires de $\ell/2$ bits

$N = p \cdot q$ (modulus)

$\varphi(N) = (p-1) \cdot (q-1)$

$e \cdot d = 1 \pmod{\varphi(N)}$

e : exposant de chiffrement (encryption)

d : exposant de déchiffrement (decryption)

Chiffrement

$c = m^e \pmod{N}$

Déchiffrement

$m = (c^e) mod(N)$

Clé symétrique (bits)	56	64	73	88	109	128	...
Clé asymétrique (bits)	510	725	1024	2048	4096	6974	...

Authenticité des données
MAC (Message Authentication Code) : code de taille fixe. En gros, en passe notre message dans une FNC MAC (Enc symetrique avec donc clé secret), de ça en sort un "tag" qui seras envoyer avec le message, si le receveur utilise le FNC MAC avec la meme clé sur le message reçus et que ça donne le tag → happy
Construction naïve (vulnérable) : concat(clé & message) → hash → tag

HMAC:
Co(x,y) = Concaténation(x,y)
ipad et opad sont des constants
ipad-xor = {secrete key} xor ipad
opad-xor = {secrete key} xor opad
Sécurisé, FONCTION :
m = Co(hash(Co(ipad-xor, message)), opad-xor)
return hash(m);

Signature :
c'est un RSA inversé, tous le monde peux dec mon message mais je suis le seul à pouvoire l'enc + pas besoin de clef secret a partager

Authentification :
Jeton actif : one time password, c'est un secret partagé
Challenge/reponse : je te demande d'enc un {nonce} , si t'as le meme resulta que moi, nice
nonce = nombre arbitraire destiné à être utilisé une seule fois
Clé publiquer : Signature

PKI (public key infrastructure) :
tous les infras qui concerne les clé public
- TLS (SSL) : authentification du serveur (Web, SMTP/StartTLS, IMAPS, etc.)
- EMAIL : S/MIME, PGP, GPG

CA (Certificate Authority) : Émettre, renouveler, maintenir les certificats et CRLs
RA (Registration Authority) : Gérer les demandes : stockage, contrôle des données soumises, communication avec l'entité, vérification du respect de lapolitique de certification
VA (Validation Authority) : Service de contrôle de certificats (signature, date, validité, etc.)
CPS (Certification Practice Statement) : pratiques utilisées par une CA pour émettre les certificats.
CP (Certificate Policy) : Règles sous lesquelles le certificat a été émis
Point critique du stystème : génération des clés et ou on les stockes

Vulnérabilité bas niveau
Memory overflows:
- stack overflow : écrasement/lecture des variables locales (Heartbleed)
- stack smashing : écrasement du retour de fonction
- heap overflow : écrasement des variables allouées dynamiquement

Buffer overflow : dépassement de tampon => exécution de code non-autorise / dénis de service => valider les entrées, vérifier la longueur, utiliser Java/Perl.
Interger overflow : débordement d'entier => toujours identifier les bornes.

Attaque
Stack overflow : les variables de la stack sont écrasées.
- Shellcode : suite d'instructions destinées à être injectées (aussi petit que possible, et exécutable).

Stack off-by-one : dépassement d'un seul caractère.
Heap overflow : manipulation des variables sur le heap.
Integer overflow : les entiers dépassent la taille possible (erreur dans le calcul de la taille du buffer, p.ex.).
- Head wrap-around attacks : indication d'une valeur trop grande pour l'allocation de la mémoire.
- Negative-size bug : la fonction interprète un entier signé comme un non signé (par exemple, -5 caractères). -

Format string bug : un printf interprète incorrectement le format de la chaîne.

DEFENCE
SSL: Secure Socket Layer
TLS: Transport Layer Security
Objectif : Outils pour établir des communication sécurisé
Protocol utiliser pour la securité sur internet, de ça on peut crée de nouveau protocol
HTTP(port 80) → HTTPS (port 443)
ESMTP (Send mail), **POP3**(Get mail), **IMAP**(manage mail)
PGP (Pretty Good Privacy) : premier logiciel qui permet à tous le monde de securisé ses doc et mails
marche du coup avec des certif si jamais
PGP = GPG (pour gnu)

Firewall : toujour pratique
Topologie sandwich (c'est plus ou moin une topo DMZ)

Outile reseau :
Sing (ping avancé) :
sing -(echo/mask) 192.168.0.255
ICMPscan : scan all hosts across a number of subnets
NMAP : scanner de ports libre
- sV : Probe open ports to determine services/version info
- O : enables OS detection
- v : affichage verbale
- A : detect tous

OUTIL DOC
Normal Command Linux :
grep : search a specific patterns with regex
- i : ignore case | -r : recursively | -w: full word only
[^a-Z] = PAS [a-Z]
useradd (add usr) | usermod(modify) | rm(supp file)
wine (use windows exe on linux)

Hash Cat :
hashcat -m <hash_type> -a <attack_mode> -o <file_format> hashfile

m) **0** (MD5) | **900** (MD4) | **1000** (NTLM)| **3000**(LM)

a)
0 (Dico) | **1** (Multi Dico) | **3** (Brute-force) | **6,7** (Hybrid att, word list + mask ou inverse)