Sécurité informatique
INF36207

Hacking, cracking et exploitations. Exploitation de failles logicielles. Injection de données malicieuses, injections SQL, HTML, XSS et autres attaques spécifiques au langage de programmation/logiciel. Comment solidifier le code.

Martin Arsenault, ing., MBA,MGP

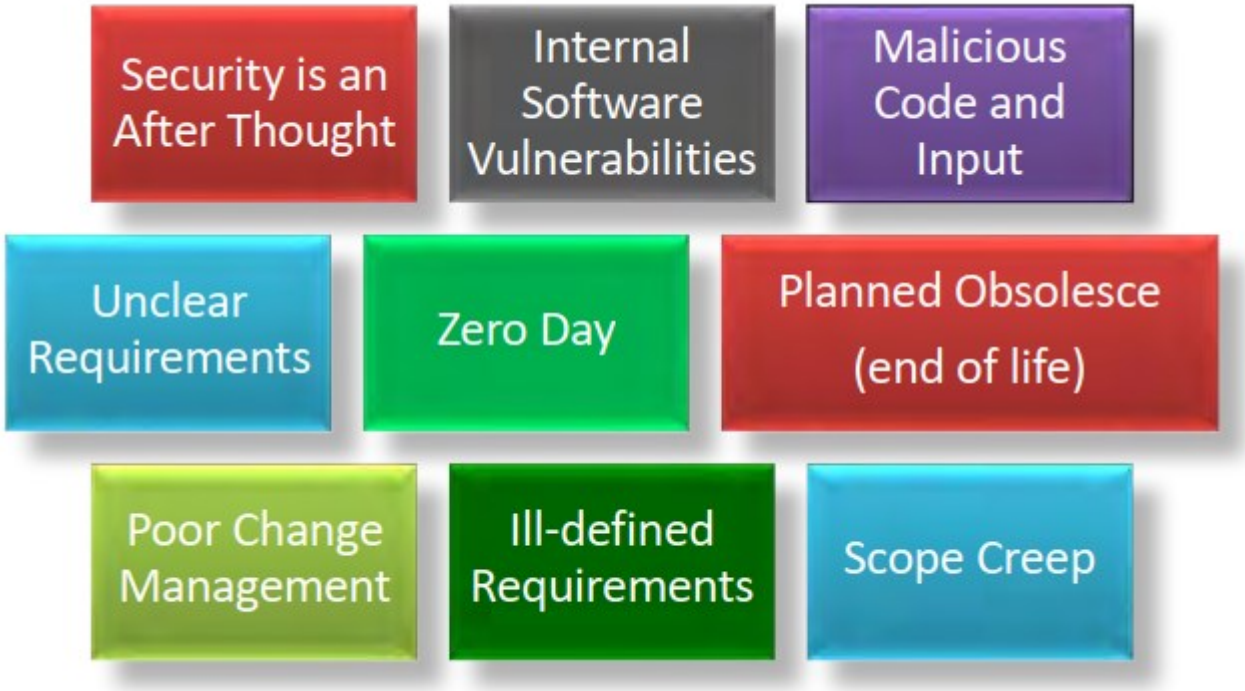
Hiver 2023

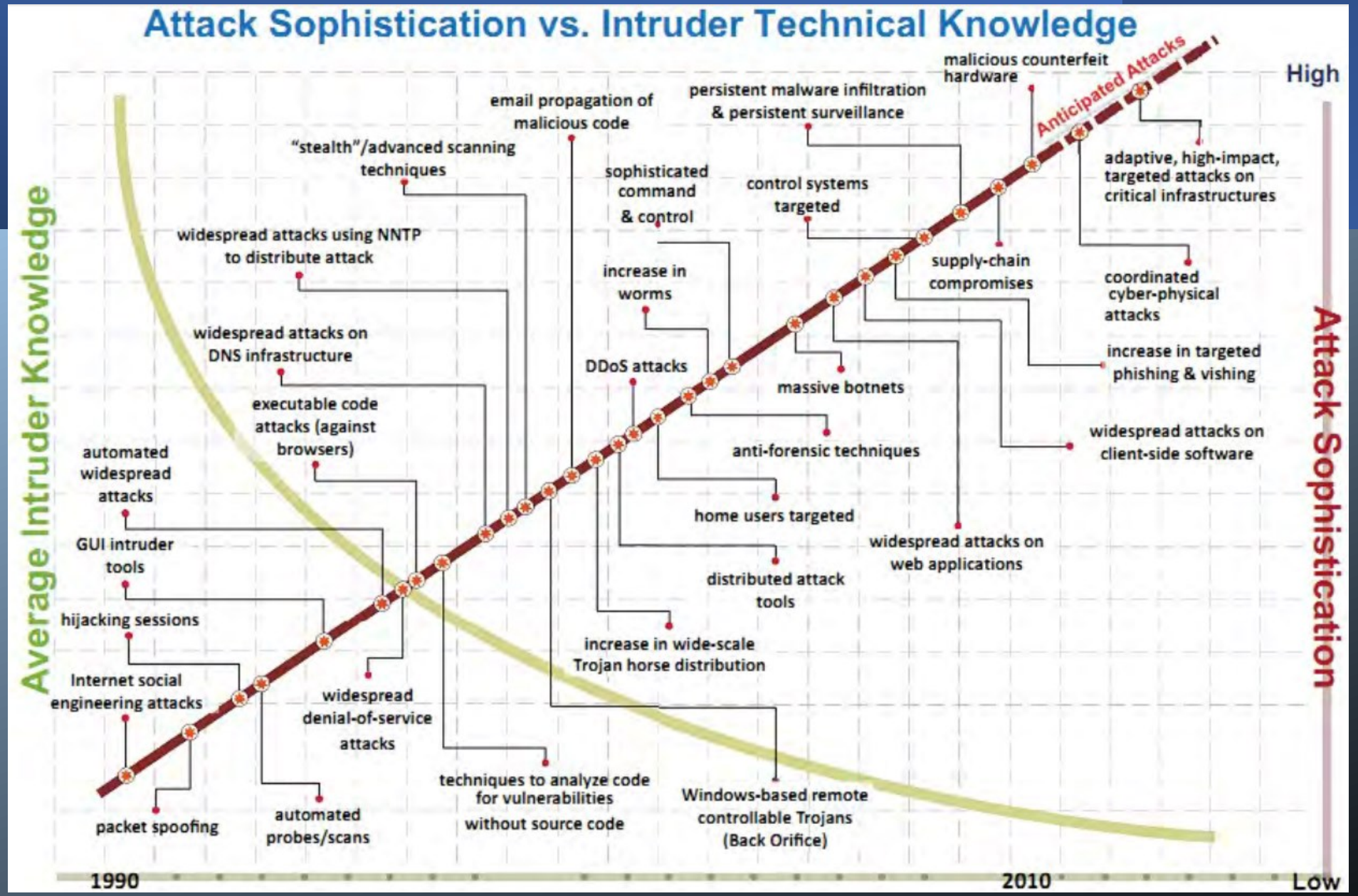
Horaire de ce soir

- Correction de l’examen
- Théorie du cours #13

Faiblesse en matière de sécurité à la base du code source

- Vecteur et surface d'attaque
  - Comment un agent malicieux peut vous attaquer ?
- Analyse de la surface d'attaque
  - Identifiez et réduisez ce qui est disponible pour les utilisateurs non fiables
- Modélisation des menaces
  - Méthodologie structurée pour découvrir comment un compromise pourrait se produire





Fonctionnalités VS Sécurité

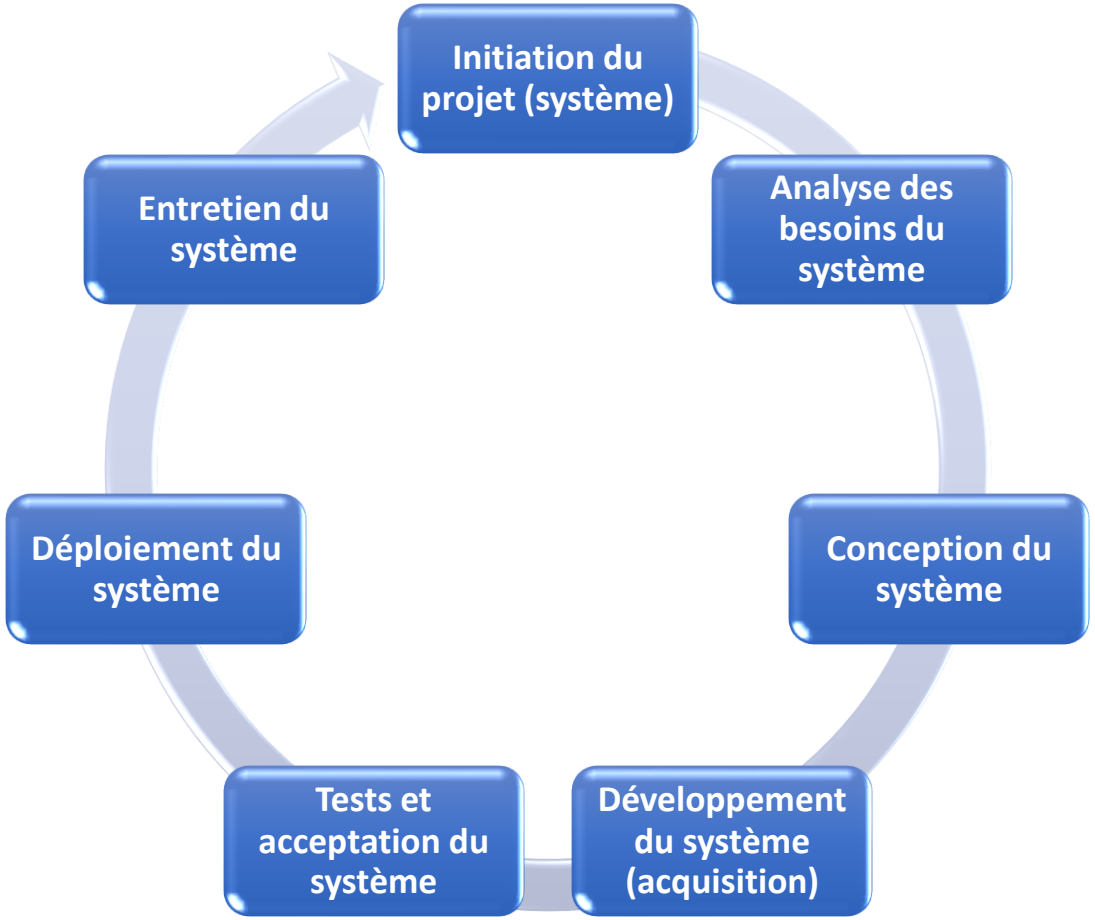
- Différents environnements requiert différents mécanismes de sécurité à mettre en place;
- Le cycle de vie de développement de système informatique (SDLC - Software Development Life Cycle) est un processus utilisé pour développer et maintenir des systèmes informatiques de manière efficace.
- Il s'agit d'un processus continu qui peut être mis en place par les organisation qui développement des systèmes informatiques (d'information).
- Le processus peut être adapté spécifiquement au projet selon les besoins.
- À terme, le cycle de vie assure une conception, une mise en œuvre et une maintenance continue et efficace des systèmes.

SDLC - Software Development Life Cycle

ISO/IEC 15288:2008 - Systems and software engineering (System life cycle processes)

ISO/IEC 12207:2008 - Systems and software engineering (Software life cycle processe

SDLC - Software Development Life Cycle



Security by Design

Sécurisé dès la conception

SDLC - Software Development Life Cycle

- Initiation du projet (système)
  - Un besoin ou une opportunité est défini.
  - Une proposition de concept est faite.
  - Une première étude de faisabilité est menée.
  - Une charte de projet (si nécessaire) est formulée.
- Analyse des exigences du système
  - Analyser les besoins des utilisateurs et développer les exigences des utilisateurs.
  - Créer un document d'exigences fonctionnelles détaillé.
  - Décomposez le système, le processus ou le problème en unités ou modules discrets et utilisez des diagrammes et d'autres outils visuels pour analyser la situation ou le besoin.
  - Toute exigence de sécurité doit être définie.

SDLC - Software Development Life Cycle

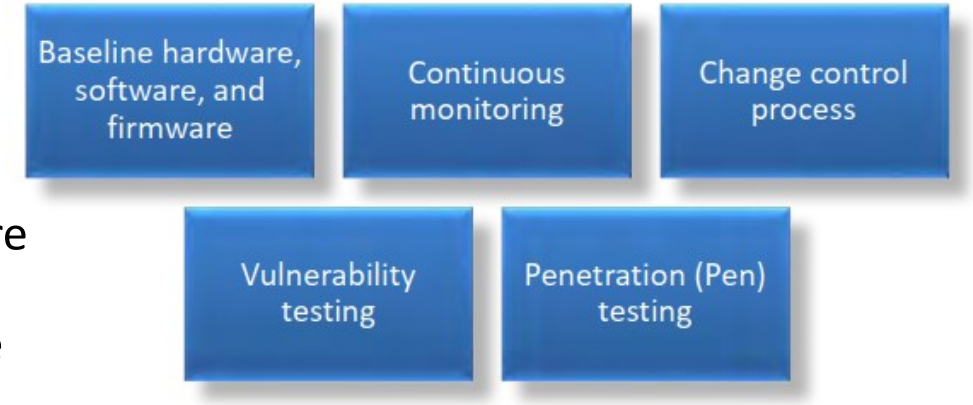
- Conception du système
  - Cette phase transforme les exigences en un document de conception.
  - Les fonctions et les opérations du système ou du logiciel en cours de conception sont décrites en détail.
  - Une analyse des risques doit être effectuée entre les phases d'exigences du système et de conception du système.
  - Une revue de conception finale doit être effectuée pour s'assurer que la conception tient compte de l'aspect pratique, de l'efficacité, du coût, de la flexibilité et de la sécurité.
- Développement du système (acquisition)
  - Cette phase implique la transformation des documents de conception détaillée en un produit fini ou une solution.
  - Les tests manuels et automatisés au niveau d'une unité ou d'un module sont effectués tout au long de cette phase par les développeurs du système ou du logiciel.
  - Les considérations de sécurité sont prises en compte lors des tests.
  - Un produit tiers peut être utilisé en tant que système ou solution logicielle s'il correspond le mieux aux besoins de l'utilisateur et s'il est plus pratique du point de vue du budget et/ou des ressources.
  - Cependant, toutes les phases suivantes doivent être suivies, que la solution ait été développée en interne ou achetée.

SDLC - Software Development Life Cycle

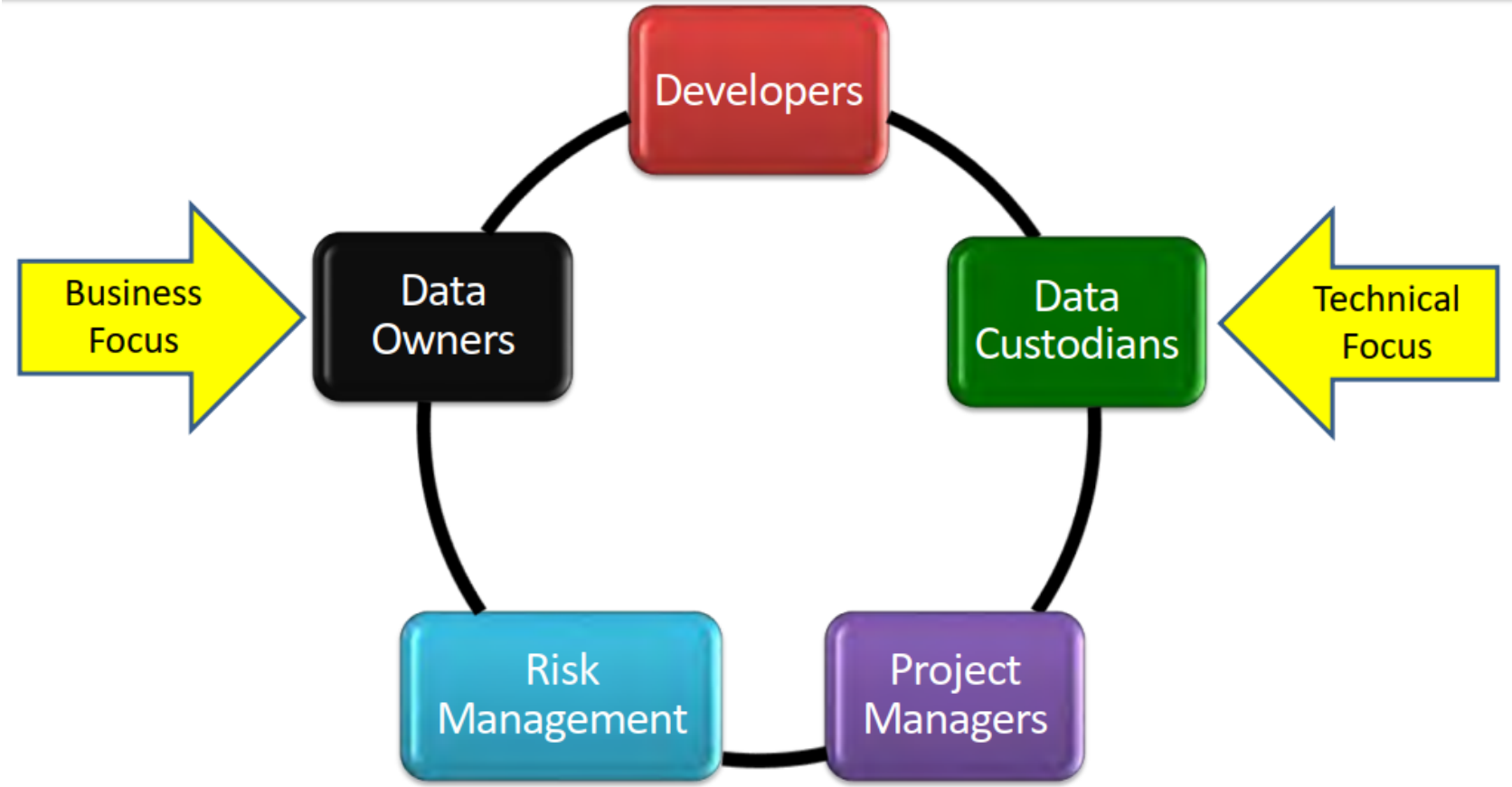
- Tests et acceptation du système
  - Cette phase doit valider ou confirmer que le système ou le logiciel développé répond à toutes les exigences fonctionnelles telles que saisies lors de la phase d'analyse des exigences du système.
  - Des représentants distincts du groupe de développement doivent effectuer des tests d'assurance qualité (AQ) internes.
  - Le ou les représentants du groupe d'utilisateurs doivent effectuer des tests d'acceptation par les utilisateurs.
  - La documentation pendant les tests doit détailler et faire correspondre les critères de test aux exigences spécifiques.
  - Alors que les tests unitaires et de modules doivent être effectués tout au long du SDLC, cette phase implique des tests holistiques du produit fini et les tests d'acceptation finaux par le ou les utilisateurs.
  - Les tests finaux d'évaluation de la sécurité sont maintenant effectués.
  - Tout problème identifié au cours des phases précédentes doit être résolu ou résolu avant la mise en œuvre.
- Déploiement du système
  - Le système ou le logiciel fini, testé et accepté par l'utilisateur est déplacé de l'environnement de test vers la production.
  - Tous les outils, codes ou mécanismes d'accès utilisés pour le développement ou le test du système ou du logiciel doivent être supprimés du logiciel qui est transféré dans un environnement de production.
  - Toute formation utilisateur nécessaire doit être effectuée avant ou pendant cette phase.

SDLC - Software Development Life Cycle

- Entretien du système
  - Cette phase correspond à la durée de vie continue du système ou du logiciel.
  - Contrairement aux autres phases, cette phase ne se termine que lorsque le système ou le logiciel est mis hors service.
  - Une structure de support client/utilisateur et tout autre processus de support opérationnel nécessaire doivent être en place.
  - Toute modification planifiée du système ou du logiciel doit être planifiée, communiquée et documentée.
  - Des tests d'intrusion de sécurité continus sont effectués sur le système ou le logiciel tout au long de son cycle de vie à des intervalles réguliers.
  - Des tests de sécurité obligatoires sont effectués lorsqu'un changement majeur de configuration ou d'architecture est effectué.



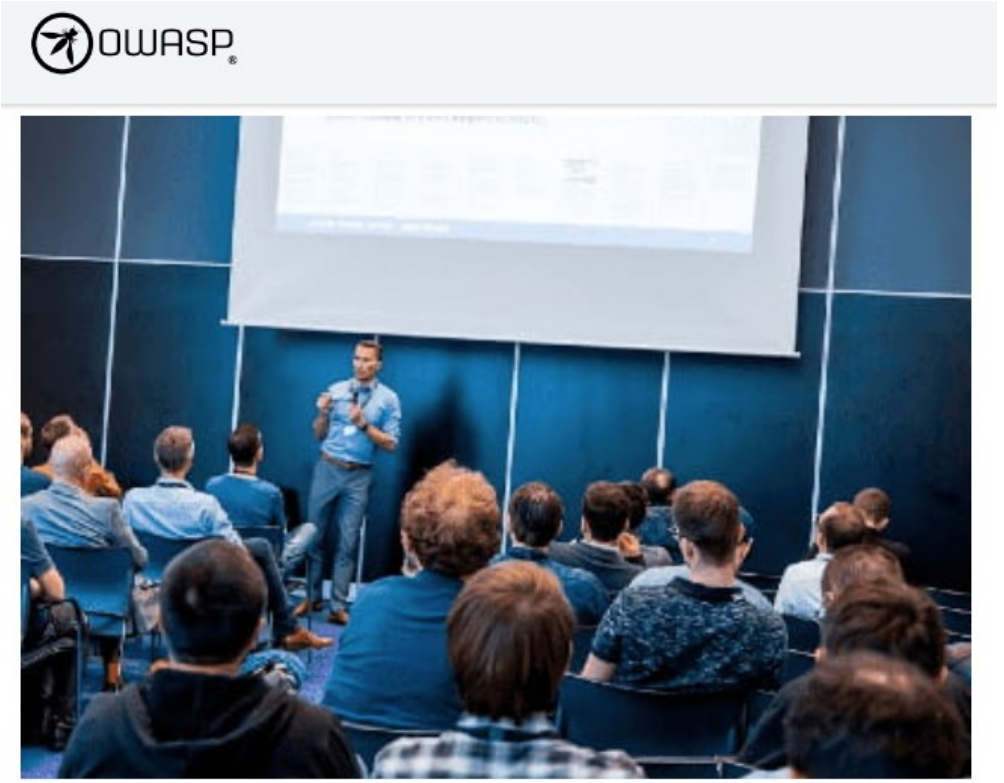
Équipe de produit intégrée



Migration des systèmes vers l'infonuagique

OWASP – Open Worldwide Application Security Project

- L'Open Worldwide Application Security Project® (OWASP) est une fondation à but non lucratif qui travaille à améliorer la sécurité des logiciels.
- Grâce à des projets de logiciels open source dirigés par la communauté, des centaines de chapitres locaux dans le monde, des dizaines de milliers de membres et des conférences éducatives et de formation de premier plan, la Fondation OWASP est la source pour les développeurs et les technologues pour sécuriser le Web.
  - Outils et ressources
  - Communauté et réseautage
  - Éducation et formation



OWASP Cheat Sheet

OWASP – Open Worldwide Application Security Project

- Quelques projets intéressants :
  - Web Security Testing Guide
    - L'empreinte digitale du serveur Web consiste à identifier le type et la version du serveur Web sur lequel une cible s'exécute. Bien que les empreintes digitales de serveur Web soient souvent encapsulées dans des outils de test automatisés, il est important que les chercheurs comprennent les principes fondamentaux de la manière dont ces outils tentent d'identifier les logiciels, et pourquoi cela est utile.
  - Modèle de développement SAMM
    - Fournir une méthode efficace et mesurable d'analyser et d'améliorer votre cycle de vie de développement sécurisé. SAMM prend en charge le cycle de vie complet du logiciel et est indépendant de la technologie et des processus. Nous avons conçu SAMM pour qu'il soit évolutif et axé sur les risques, car il n'y a pas de recette unique qui fonctionne pour toutes les organisations.
  - OWASP TopTen
    - Le Top 10 OWASP est un document de sensibilisation standard pour les développeurs et la sécurité des applications Web. Il représente un large consensus sur les risques de sécurité les plus critiques pour les applications Web.
  - OWASP Application Security Verification Standard
    - Le projet OWASP Application Security Verification Standard (ASVS) fournit une base pour tester les contrôles techniques de sécurité des applications Web et fournit également aux développeurs une liste d'exigences pour un développement sécurisé.

Le modèle SAMM

- La mission du Software Assurance Maturity Model (SAMM) de l'OWASP est d'être le modèle de maturité principal pour l'assurance logicielle qui fournit un moyen efficace et mesurable pour tous les types d'organisations d'analyser et d'améliorer leur posture de sécurité logicielle. OWASP SAMM prend en charge le cycle de vie complet du logiciel, y compris le développement et l'acquisition, et est indépendant de la technologie et des processus. Il est intentionnellement conçu pour être évolutif et axé sur les risques.
- Le modèle original (v1.0) a été écrit par Pravir Chandra et date de 2009. Au cours des 10 dernières années, il s'est avéré être un modèle largement distribué et efficace pour améliorer les pratiques logicielles sécurisées dans différents types d'organisations à travers le monde.
- La version 2.0 améliore le modèle pour faire face à certaines de ses limitations actuelles.

Le modèle SAMM

- SAMM est un modèle normatif, un cadre ouvert simple à utiliser, entièrement défini et mesurable. Les détails de la solution sont assez faciles à suivre, même pour les gens ne touchant pas à la de sécurité. Il aide les organisations à analyser leurs pratiques actuelles en matière de sécurité logicielle, à créer un programme de sécurité par itérations définies, à montrer des améliorations progressives des pratiques sécurisées et à définir et mesurer les activités liées à la sécurité.
- SAMM a été défini dans un souci de flexibilité afin que les petites, moyennes et grandes organisations utilisant n'importe quel style de développement puissent le personnaliser et l'adopter. Il fournit un moyen de savoir où en est votre organisation dans son cheminement vers l'assurance logicielle et de comprendre ce qui est recommandé pour passer au niveau de maturité suivant.
- SAMM n'insiste pas pour que toutes les organisations atteignent le niveau de maturité maximum dans chaque catégorie. Chaque organisation peut déterminer le niveau de maturité cible pour chaque pratique de sécurité qui lui convient le mieux et adapter les modèles disponibles à ses besoins spécifiques.

MEASURABLE

Defined maturity levels across security practices

ACTIONABLE

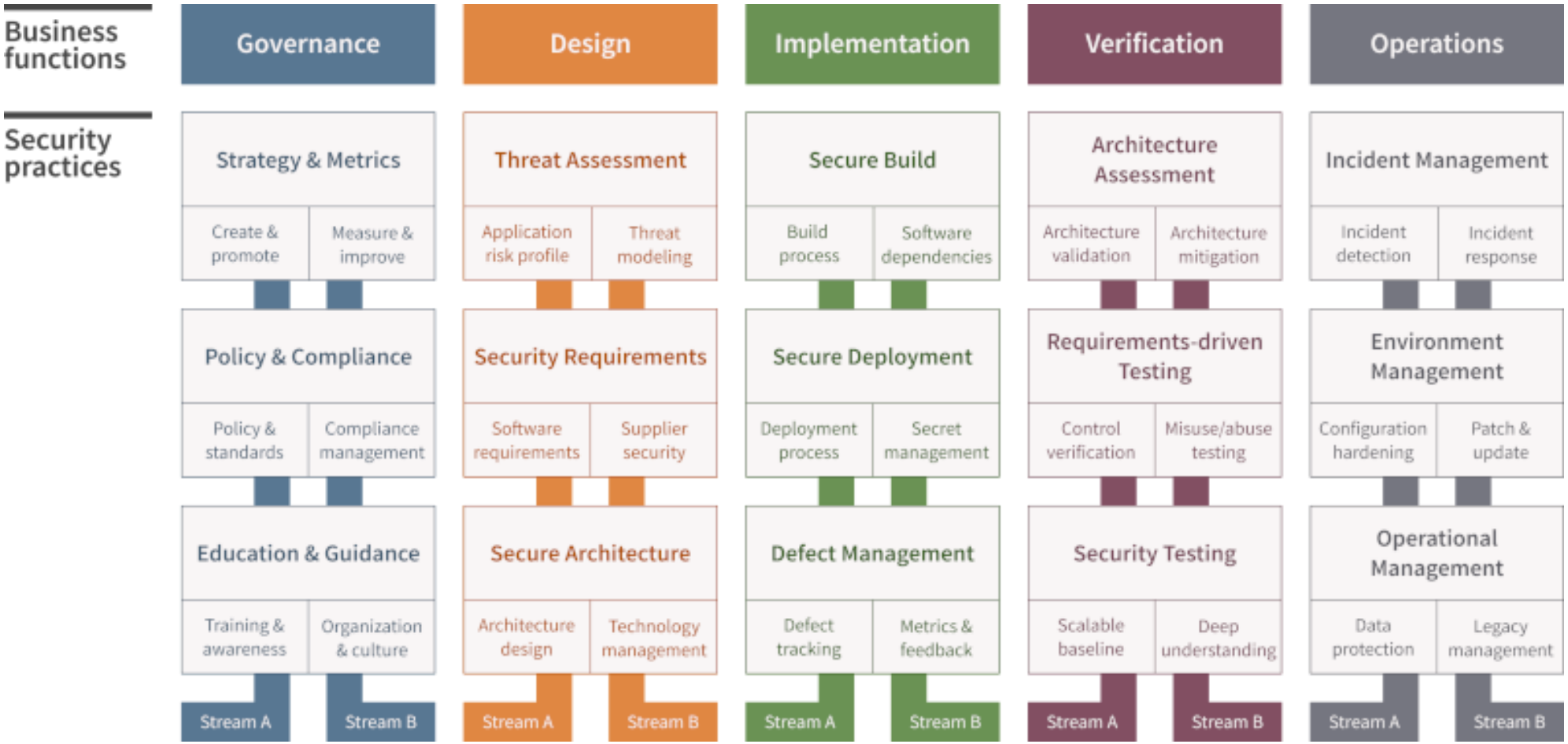
Clear pathways for improving maturity levels

VERSATILE

Technology, process, and organization agnostic

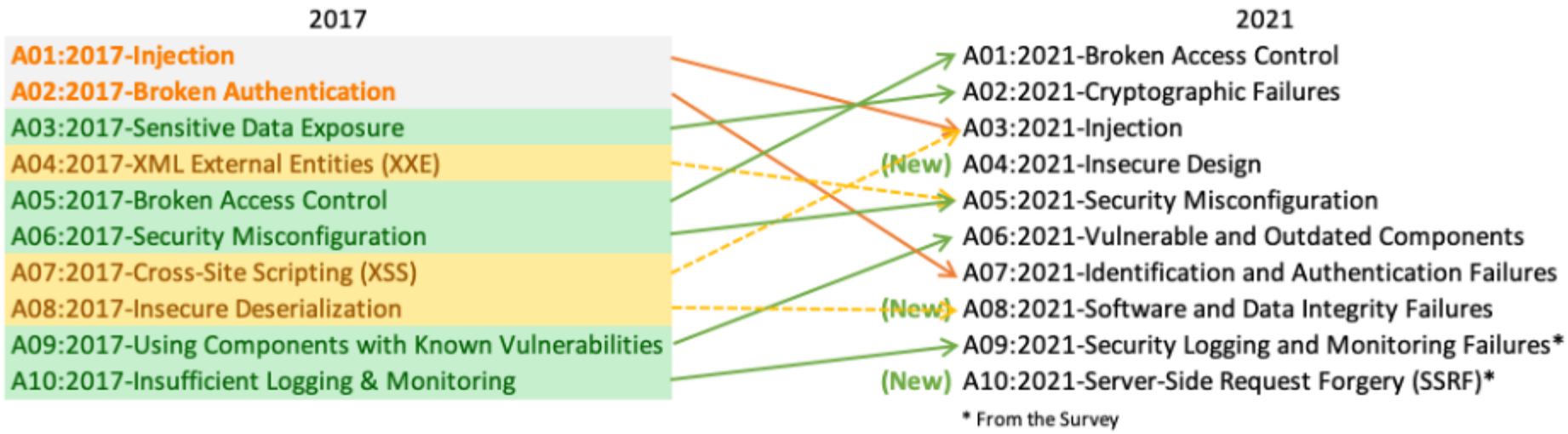
[Modèle SAMM V2]

Software Assurance Maturity Model (SAMM)



Top 10 OWASP

- Le Top 10 OWASP est un document de sensibilisation standard pour les développeurs et la sécurité des applications Web. Il représente un large consensus sur les risques de sécurité les plus critiques pour les applications Web.



Top 10 OWASP

- Le Top 10 de l'OWASP est avant tout un document de sensibilisation. Cependant, cela n'a pas empêché les organisations de l'utiliser comme norme AppSec de facto de l'industrie depuis sa création en 2003.
- Si vous souhaitez utiliser le Top 10 de l'OWASP comme norme de codage ou de test, considérez le comme étant le minimum à mettre en place et un point de départ. Ne vous y limitez pas!
- L'une des difficultés de l'utilisation du Top 10 OWASP comme standard est qu'il documente les risques AppSec, et non pas nécessairement les problèmes facilement testables.
  - Par exemple, A04:2021-Insecure Design dépasse le cadre de la plupart des formes de test.

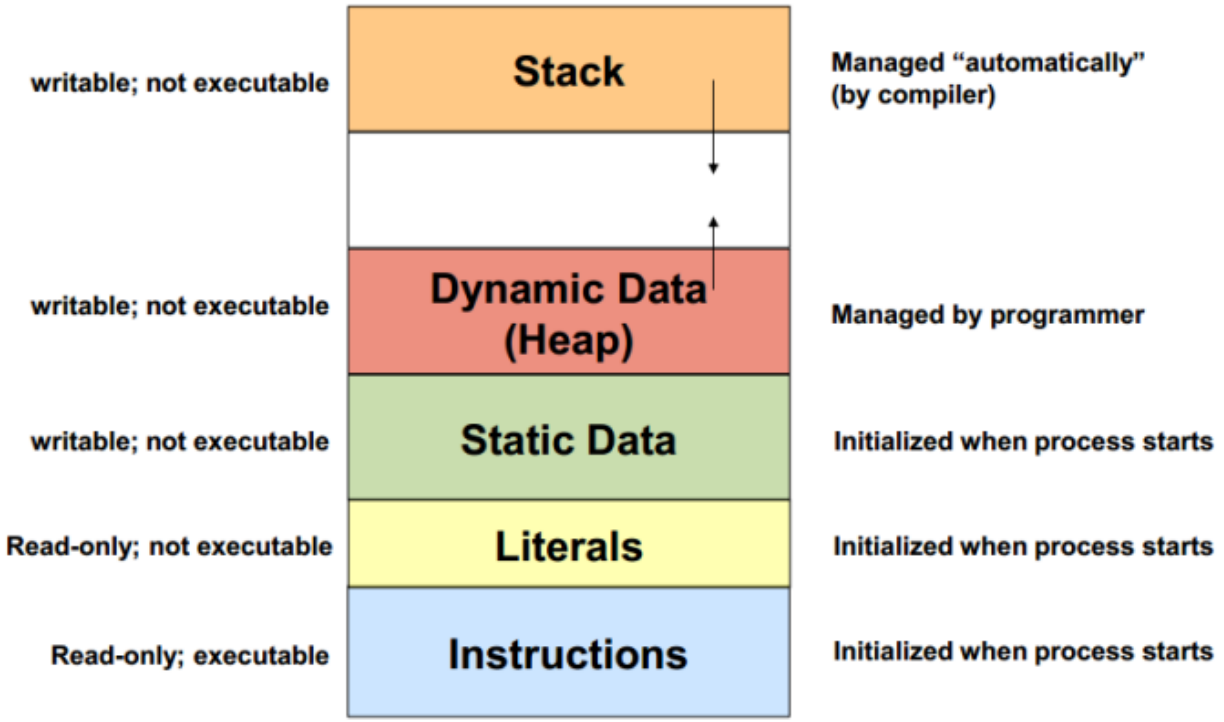
Quelques vulnérabilités courantes

Buffer Overflow / Débordement de mémoire

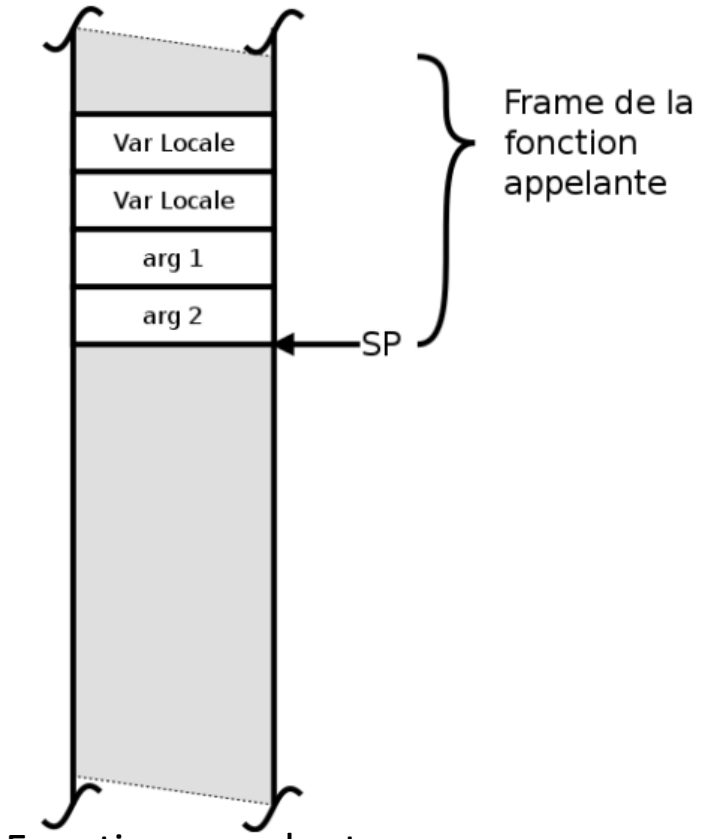
- La pile d'appel
  - La quasi-totalité des langages de programmation sont stack-based :
    - La pile est un mécanisme fondamental du langage
    - Elle sert aux appels de fonctions et au passage de paramètre
    - Elle sert aux variables locales des fonctions
    - ...mais les détails dépendent fortement du type de processeur.
  - Sur x86 et AMD64 (x86-64) :
    - Le segment de mémoire de pile est distinct du segment de code et du heap
    - La pile « croît » vers le bas (pour des raisons historiques)
    - Les valeurs sont alignées sur des frontières de 16, 32, 64 bits.

Différence entre le heap et le stack

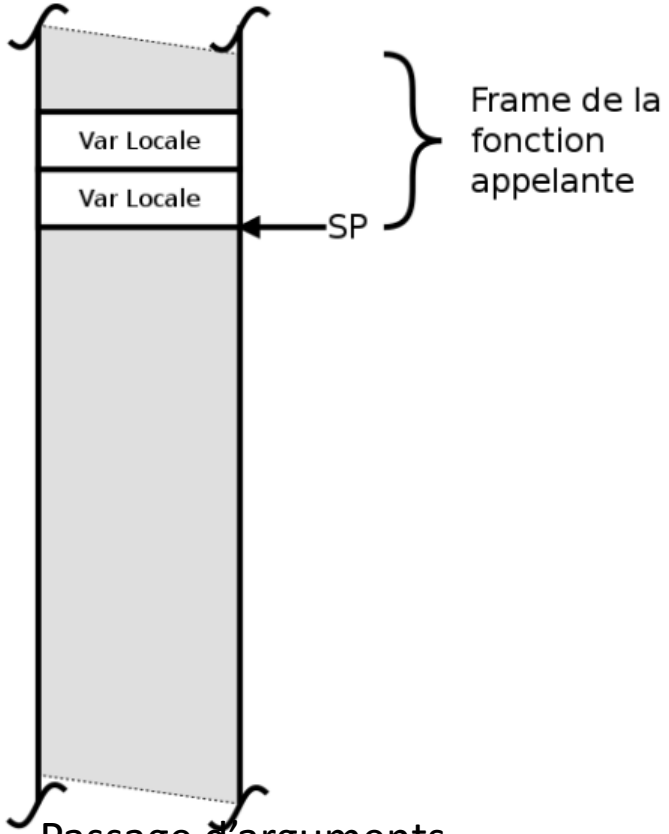
- Heap :
  - Le heap est une région de mémoire allouée dynamiquement.
  - Il est utilisée pour stocker des données de manière flexible.
  - Il utilisé pour stocker des données qui doivent être allouées et désallouées dynamiquement pendant l'exécution d'un programme
    - Ex: Stocker des tableaux de données, Stocker des matrices, etc.
  - Il est utilisé pour allouer de la mémoire à des variables qui doivent être accessibles de manière globale dans un programme.
- Stack (pile)
  - La pile est utilisée pour stocker des variables locales et temporaires lors de l'exécution d'une fonction



Exécution d'une fonction

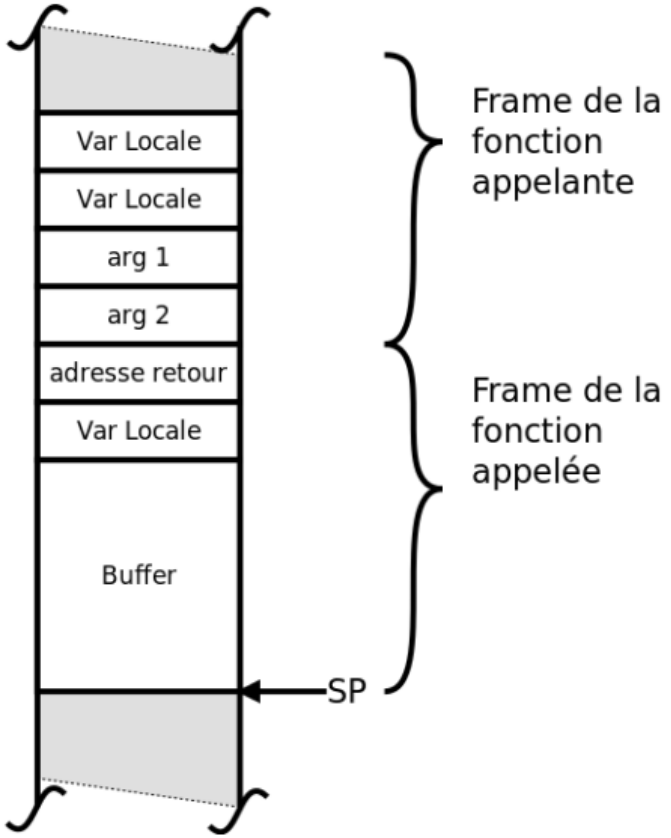


Fonction appelante

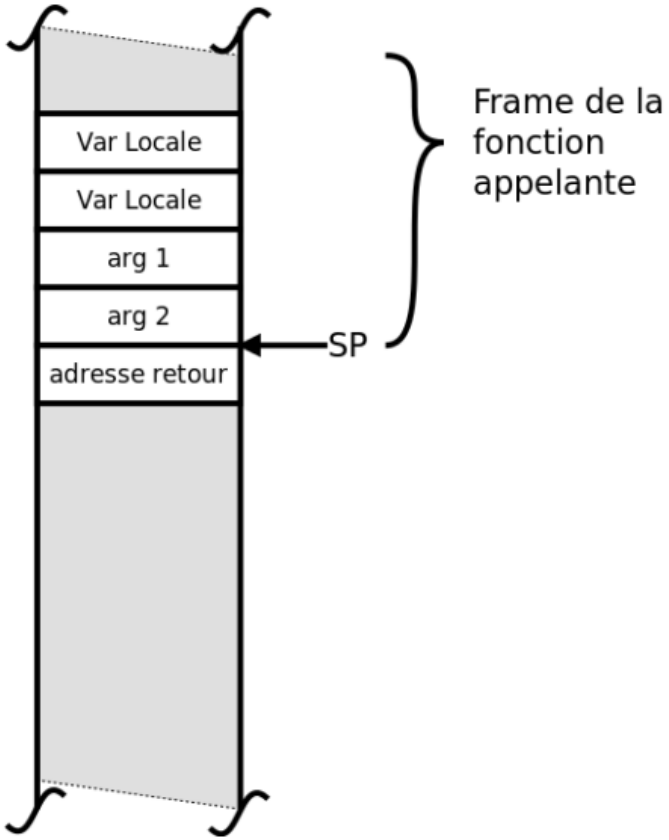


Passage d'arguments

Appel d’une fonction

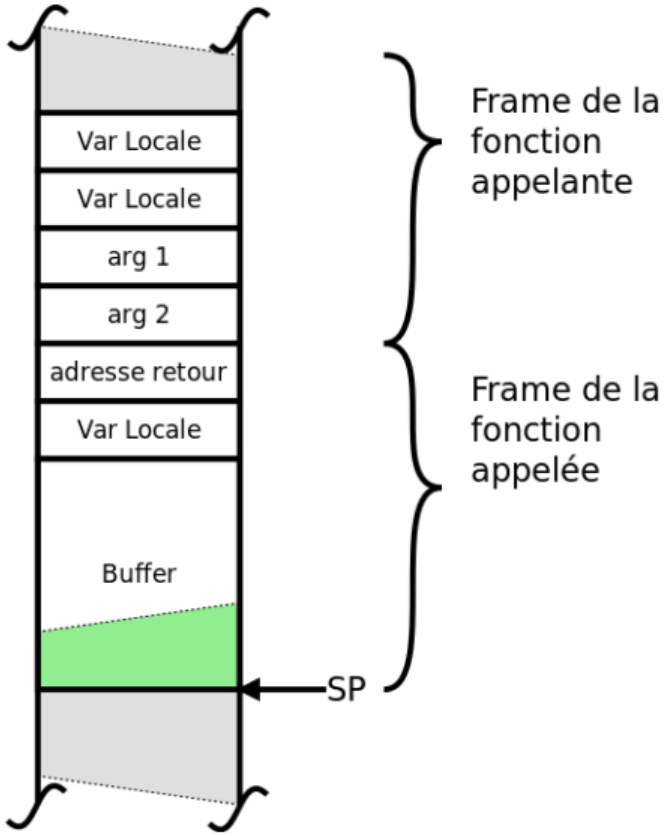


Appel

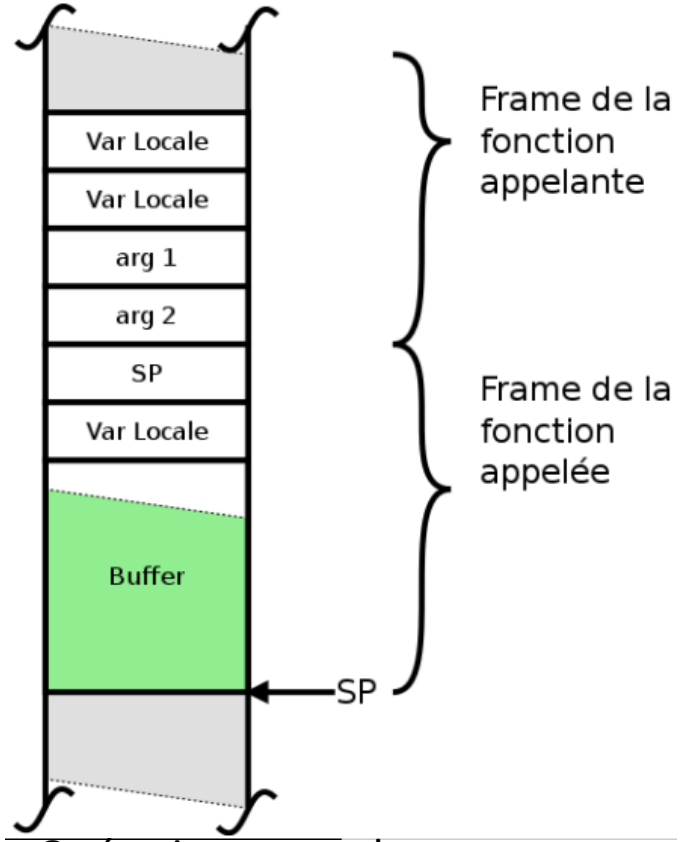


Initialisation de la fonction appelée

Opération de la fonction appelée

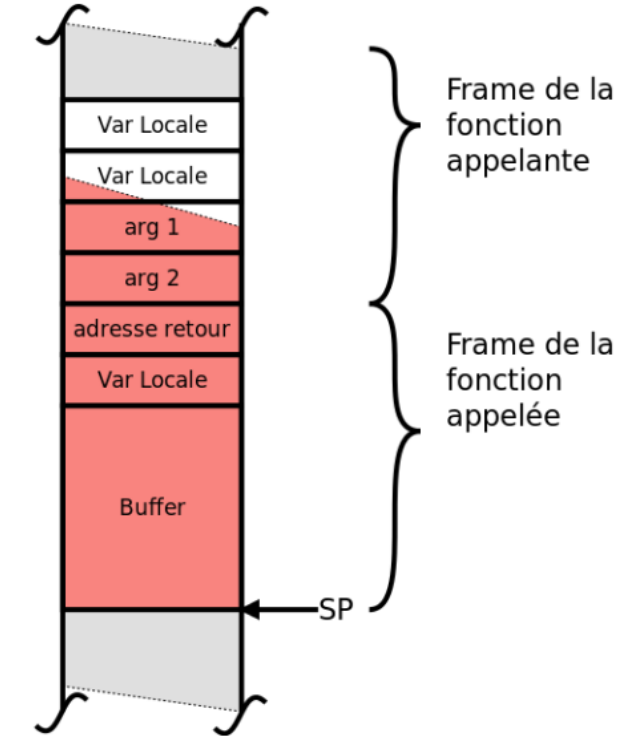


Opération normale

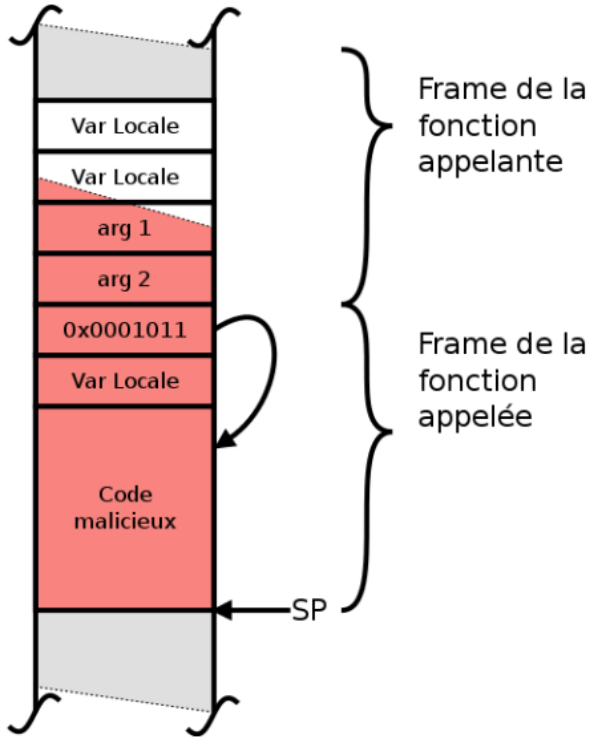


Opération normale

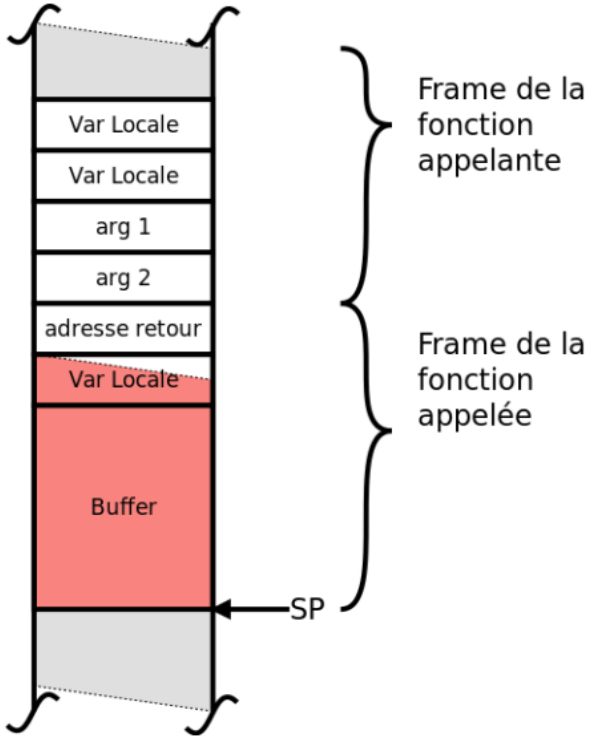
Débordement de mémoire et prise de contrôle



Dépassement



Dépassement



Prise de contrôle

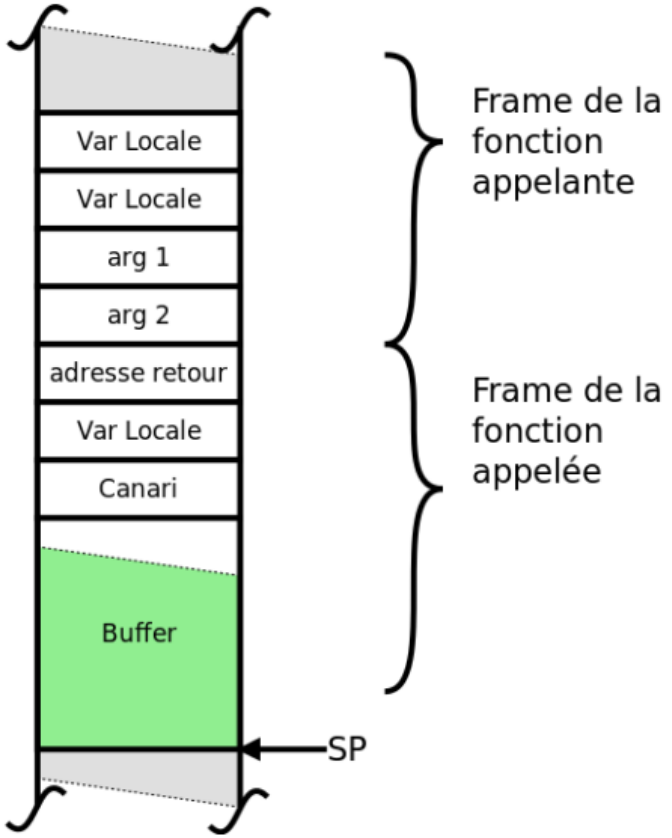
Comment éviter les débordements

- Pour déjouer les buffer overflows sur la pile :
  - Écrire du meilleur code
    - Valider toutes les entrées utilisateur telles que les entrées de formulaires web ou les arguments de ligne de commande.
    - Valider que les entrées sont de la taille et du format attendus.
    - Rejeter les entrées invalides.
  - Allocation dynamique de mémoire sécurisé :
    - Vérifier la taille de la zone mémoire allouée pour éviter de déborder la zone allouée.
    - Libérer toute la mémoire allouée lorsque celle-ci n'est plus utilisée.
  - Utiliser la pile et le heap de façon appropriée
    - Tableaux de taille fixe doivent être stockés sur la pile plutôt que sur le heap tas, car cela limite leur taille maximale et évite les débordements de tampon.
    - Les données de taille inconnue ou qui changent souvent doivent être stockées sur le heap, mais en utilisant des fonctions telles que malloc() et free() de manière appropriée.
  - Activer le bit NX
    - Marque le segment de la pile comme non-exécutable
  - Insérer des canaris…

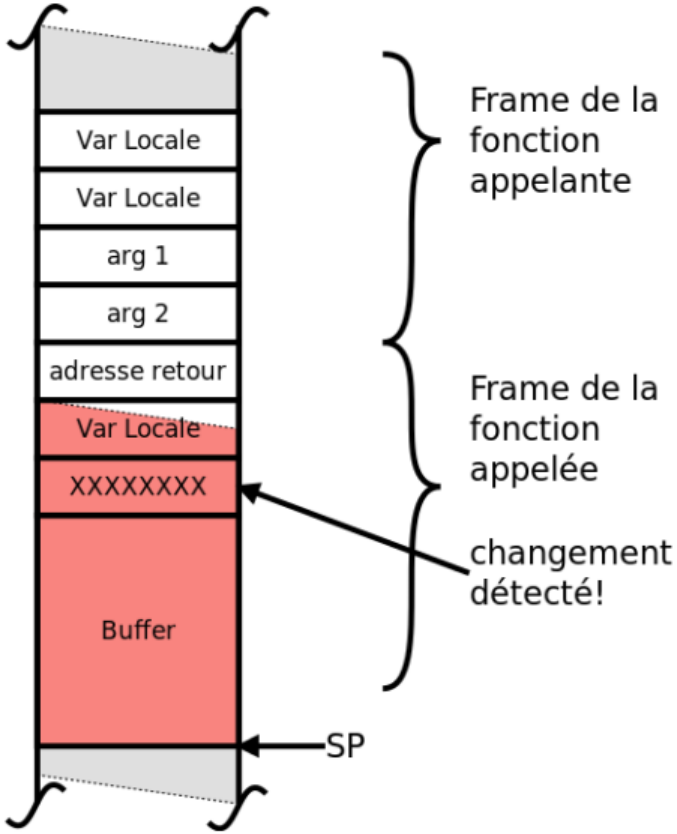
Les canaris

- Valeurs aléatoires qu'on insère à des endroits stratégiques sur la pile (généralement à la fin d'une mémoire tampon).
- La valeur du canari est aléatoire pour empêcher un attaquant d'écraser avec la même valeur pendant le débordement
- Si le canari est mort (a été modifié) alors cela indique probablement un débordement ou une écriture au mauvais endroit.
- La violation doit être détectée par le programme et celui-ci doit cesser son exécution (avant d'exécuter le code malicieux)
- La détection peut se faire par une fonction de validation qui est appelée selon un cycle prédéterminé.

L’utilisation d’un canari



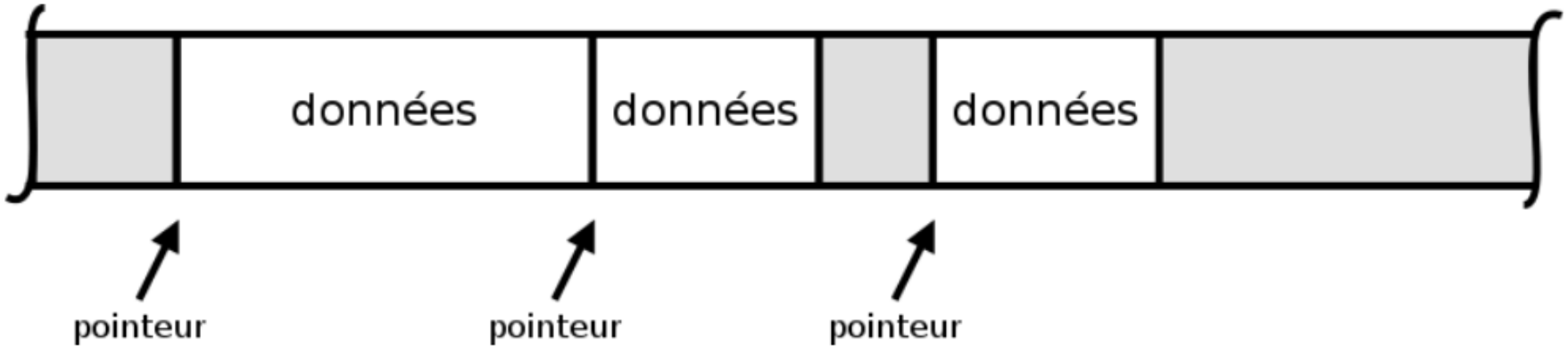
Insertion d’un canari dans la mémoire



Mort du canari

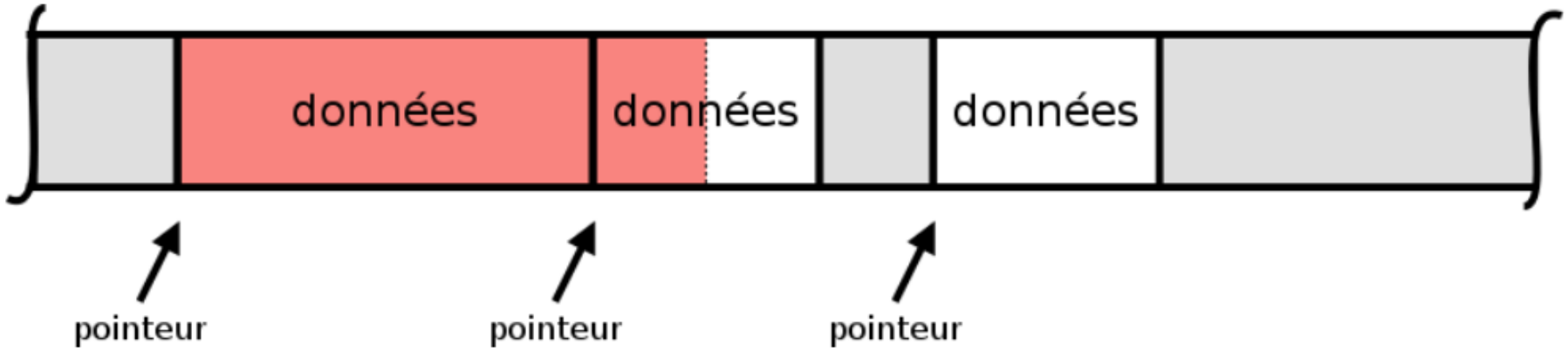
Données sur le heap

- Le heap est une grande région de mémoire où sont alloués dynamiquement des plages de mémoire utilisées par le programme.
- Les blocs sont alloués de façon à maximiser l'utilisation de la mémoire
- Dans la plupart des langages (compilés), les blocs ne changent pas de place pendant leur vie



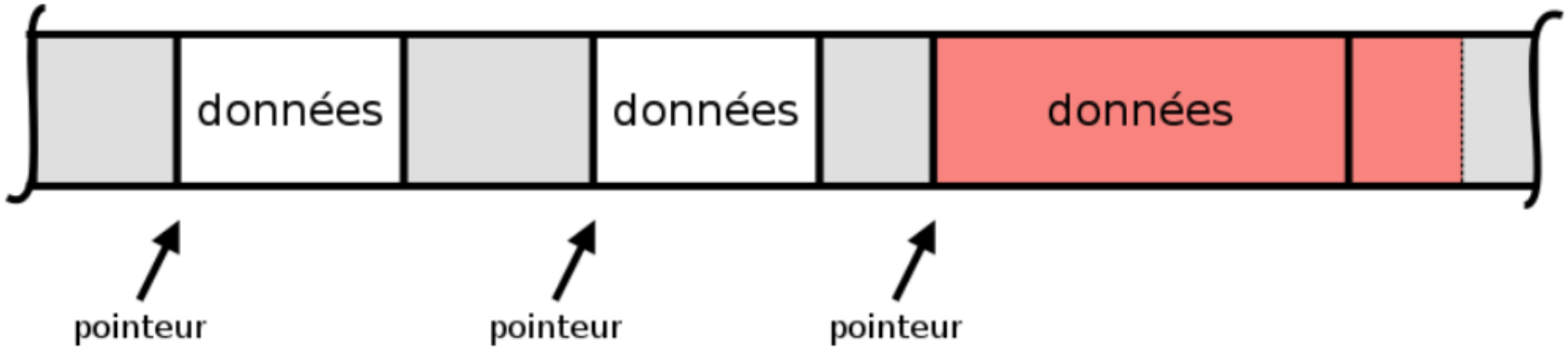
Débordement du heap

- Ce qui fait qu'un débordement de tableau est susceptible d'écraser des données voisines
- Ce qui offre un vecteur d'attaque : on peut corrompre des données (ex : écraser un security descriptor)



Déjouer les débordement sur le heap

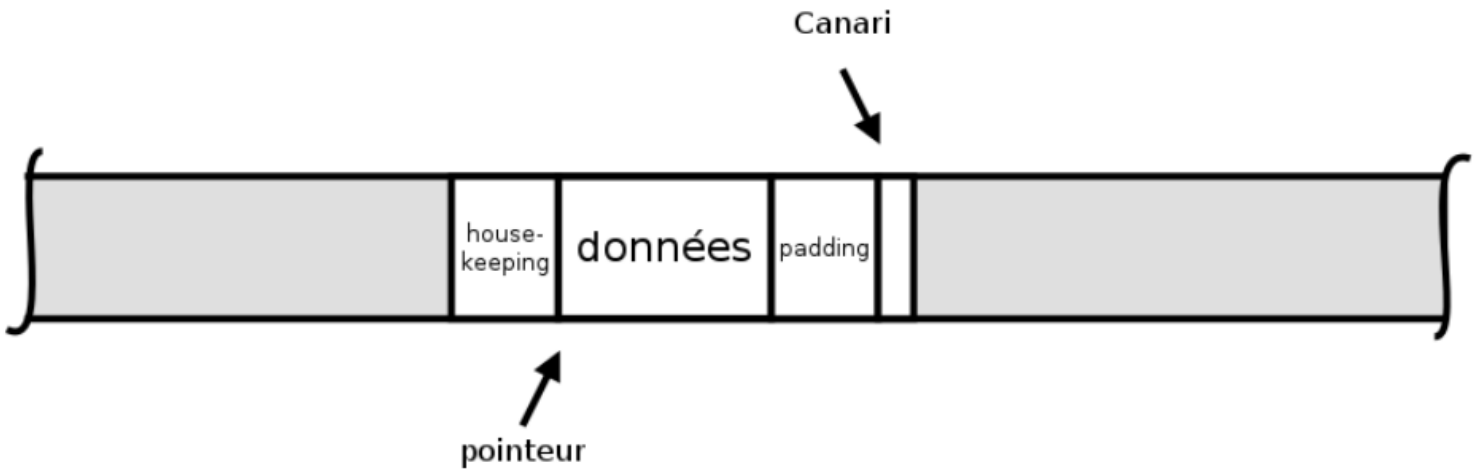
- On peut allouer de façon aléatoire l'allocation pour empêcher un attaquant d'exploiter la proximité des blocs



Utiliser le canari sur le heap

N’est pas gérée automatiquement comme pour la pile.

- L’allocation est compliquée : il y a des méta données, et du padding.
- On peut ajouter des canaris (mais c’est plus difficile de les vérifier)
- On peut exploiter les fonctions du processeur pour empêcher les débordement en ségrégant par segment



Déjouer les débordement sur le heap

- Écrire du meilleur code !
- Utiliser du code qui vérifie l'intégrité des blocs
- Utiliser les fonctions du processeur pour limiter où on peut écrire
  - Table de pagination
  - Bit de protection en écriture
  - Mémoire virtuelle
  - Mode utilisateur / noyau

Injection de code

- Il s’agit d’une technique d'attaque dans laquelle un code malveillant est inséré dans une application ou un processus en cours d'exécution. Elle permet à l’agent malveillant d'exécuter du code non autorisé sur un système et avoir un certain contrôle sur celui-ci.
    1. Injection SQL : Dans les applications web, l’agent malveillant peut utiliser cette technique pour exécuter des commandes SQL dans une base de données. Cette technique permet de récupérer des informations sensibles ou de modifier les données stockées dans la base de données.
    2. Injection de code JavaScript : l’agent malveillant peut injecter du code JavaScript dans une page web, par exemple en utilisant des champs de formulaire vulnérables. Ce code peut être utilisé pour voler des informations sensibles, tels que les cookies ou les informations d'identification, ou pour rediriger l'utilisateur vers des sites malveillants.
    3. Injection de code binaire : l’agent malveillant peut injecter du code binaire malveillant dans des applications en cours d'exécution. Cette technique est plus complexe, car elle nécessite la capacité de compiler le code malveillant dans un format binaire compatible avec le système cible.
- Pour prévenir l'injection de code, il est important de mettre en place des mesures de sécurité appropriées, telles que la validation des entrées utilisateur, la restriction des autorisations d'accès et l'utilisation de mécanismes de prévention des attaques, tels que les pare-feu d'application web (WAF) et les outils de détection d'intrusion (IDS/IPS).

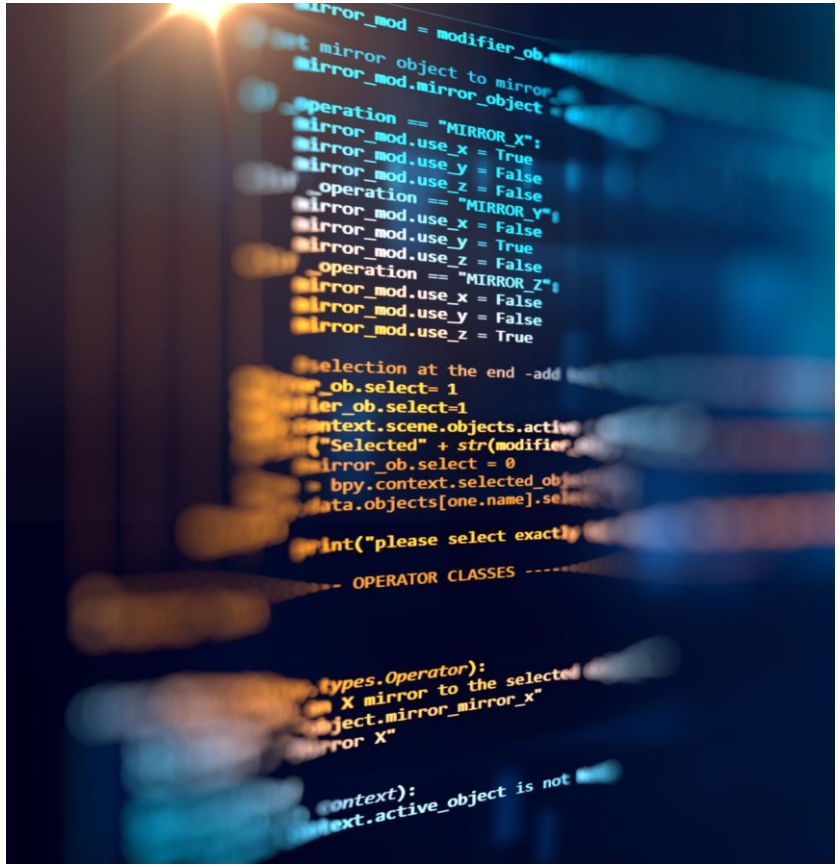
Injection de code dans Python

- Le langage Python est un langage interprété (sans compilation requise).
- Typage dynamique
- Fonctions first class objects
- La fonction eval(expr) donne accès à l'interpréteur.
- N'importe quelle expression Python peut être interprétée
- On peut injecter du code arbitraire dans l'application, eval() va l'exécuter comme si c'était dans le programme même.

Comment déjouer l'injection de code Python

…Et d'autres langages interprétés : JS, PHP, Ruby, Perl, etc.

- Utiliser un framework web sécurisé comme Django ou Flask qui offre des fonctionnalités de sécurité intégrées comme la protection contre les attaques CSRF, XSS, injection SQL, etc.
- Valider les saisies des utilisateurs.
- Utiliser uniquement des bibliothèques tierces qui ont été testées et validées pour leur sécurité.
- Sécuriser les identifiants et les mots de passe et utiliser des mécanismes d'authentification robustes comme OAuth ou OpenID Connect.
- Assurez-vous que les autorisations sont configurées adéquatement pour les fichiers, les répertoires et les bases de données pour éviter les fuites de données.
- Utiliser HTTPS pour les communications.
- Effectuer sur une base régulière des tests de sécurité pour détecter les vulnérabilités et les failles de sécurité et corriger rapidement les problèmes identifiés.
- Garder tous les composants du système (framework, bibliothèques tierces, serveur web, base de données, etc.) à jour avec les dernières versions.



Injection de code HTML

XSS

- Les pages HTML sont générées (lorsqu'elles ne sont pas statiques) server-side par des scripts.
- Les scripts vont aller pêcher des valeurs dans une base de données et construire des pages avec ces valeurs.
- Si ces valeurs ne sont pas contrôlées, on peut s'en servir pour injecter du HTML (et du javascript)

Déjouer les injection HTML

- Retirer les caractères spéciaux dans les champs de saisie comme les guillemets, les apostrophes, etc. avant de les afficher dans une page HTML. PHP et Python ont des fonctions pour cela.
- Toutes les saisies doivent être validées et filtrées pour éviter les attaques par injection XSS. L'utilisation de framework permettent de valider les entrées utilisateur.
- Utiliser des templates et des bibliothèques sécurisées plutôt que d'écrire notre propre code : Les templates et les bibliothèques fournissent des fonctions pour afficher le contenu dynamique dans une page HTML de manière sécurisée. Il est recommandé d'utiliser des templates et des bibliothèques sécurisées plutôt que d'écrire du code HTML personnalisé.
- Utiliser des CSP (Content Security Policy) : Les CSP permettent de limiter les sources autorisées pour les scripts et les ressources dans une page HTML, ce qui peut aider à prévenir les attaques XSS.
- Configurer correctement les en-têtes HTTP : Les en-têtes HTTP peuvent être utilisés pour empêcher les attaques XSS. Par exemple, l'en-tête "X-XSS-Protection" peut être configuré pour activer la protection XSS intégrée du navigateur.
- Effectuer des tests de sécurité réguliers : Effectuer régulièrement des tests de sécurité pour détecter les vulnérabilités et les failles de sécurité et corriger rapidement les problèmes identifiés.

Injection SQL

- SQL est susceptible aux mêmes types d'attaques que HTML, mais les conséquences peuvent être plus graves !

- Si on a :
  - SELECT * FROM unetable WHERE tag="+var+";"
  - ...et que var="0; DROP TABLE unetable", le code ci-dessus devient :

  - SELECT * FROM unetable WHERE tag=0; DROP TABLE unetable;
  - ...ce qui détruit la base de données ! (ou un bon morceau)

Autre exemple SQL Injection

➔ https://insecure-website.com/products?category=Gifts

- `SELECT * FROM products WHERE category = 'Gifts' AND released = 1`
  - Tous les champs (*)
  - Dans la table products
  - Lorsque la catégorie est Gifts
  - Le produit est publié (released) = 1

➔https://insecure-website.com/products?category=Gifts'--

- `SELECT * FROM products WHERE category = 'Gifts'--' AND released = 1`
- https://insecure-website.com/products?category=Gifts'+OR+1=1--
- `SELECT * FROM products WHERE category = 'Gifts' OR 1=1--' AND released = 1`

- The modified query will return all items where either the category is Gifts, or 1 is equal to 1. Since 1=1 is always true, the query will return all items.

https://portswigger.net/web-security/sql-injection

Déjouer les injections SQL

Avoir recours à des requêtes paramétrées (ex: menu déroulant) afin de séparer les données d'entrée des commandes SQL, ce qui réduit considérablement le risque d'injection SQL en vérifiant les saisies.

Utilisez des frameworks de développement sécurisés qui intègrent des protections contre les injections SQL, tels que la vérification des données d'entrée et la limitation des caractères spéciaux.

Imiter les privilèges d'accès à la base de données. Les comptes utilisateur doivent être limités aux privilèges dont ils ont besoin pour effectuer leurs tâches.

Utilisez des pare-feux applicatifs (WAF) afin de surveiller les requêtes SQL pour détecter les tentatives d'injections SQL et bloquer les attaques.

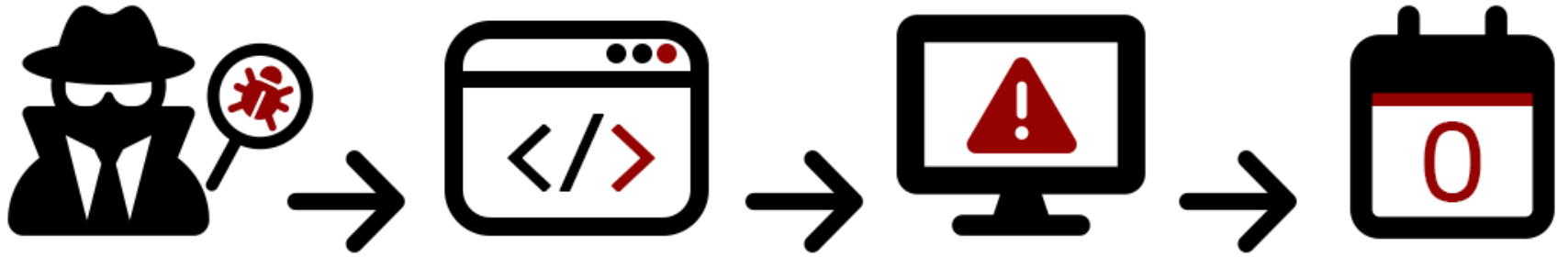
Mettre à jour régulièrement les logiciels.

Autres types d’attaques logicielles

- Les attaques sont les techniques utilisées par des agents malveillants pour exploiter les vulnérabilités des applications.
- Les attaques sont souvent confondues avec les vulnérabilités, alors il faut essayez de penser dans l’autre sens pour tenter de ce protéger :
  - Qu’est-ce que l’agent malveillant ferai s’il voudrait attaquer l’application ?
    - Plutôt que de trouver les faiblesses dans une application.

https://owasp.org/www-community/attacks/

L’œuf ou la poule ?



VULNERABILITY

A vulnerability in a system, software or service is found by an individual who decides to keep it a secret from the vendor.

EXPLOIT

Knowledge of the vulnerability is then used by either the person discovering it or a partner or contact to develop “exploit code” that is able to leverage the vulnerability.

ATTACK

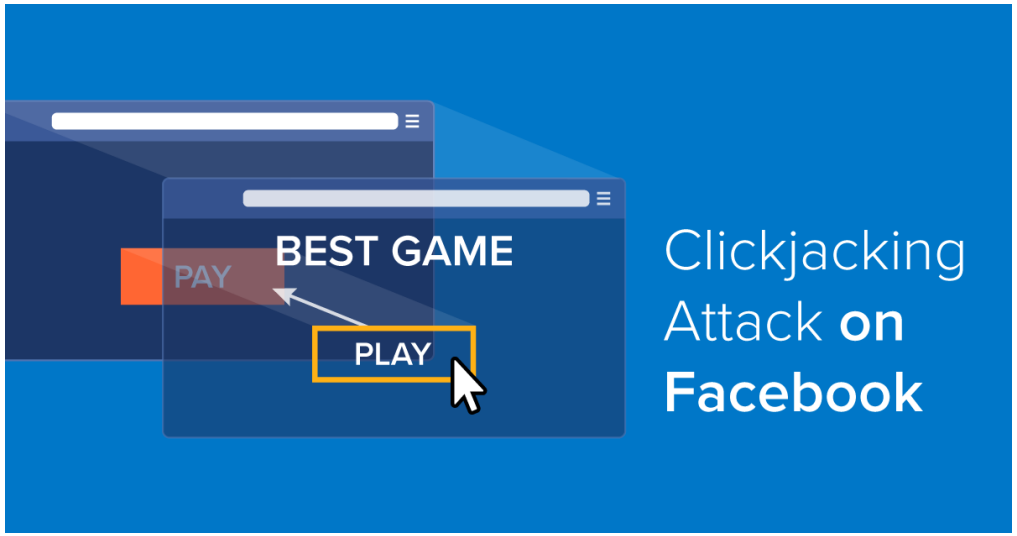
The exploit is then used to perform one or more attacks on vulnerable systems.

DAY ZERO

“Day zero” is the day the vendor learns of the vulnerability and begins working on a fix.

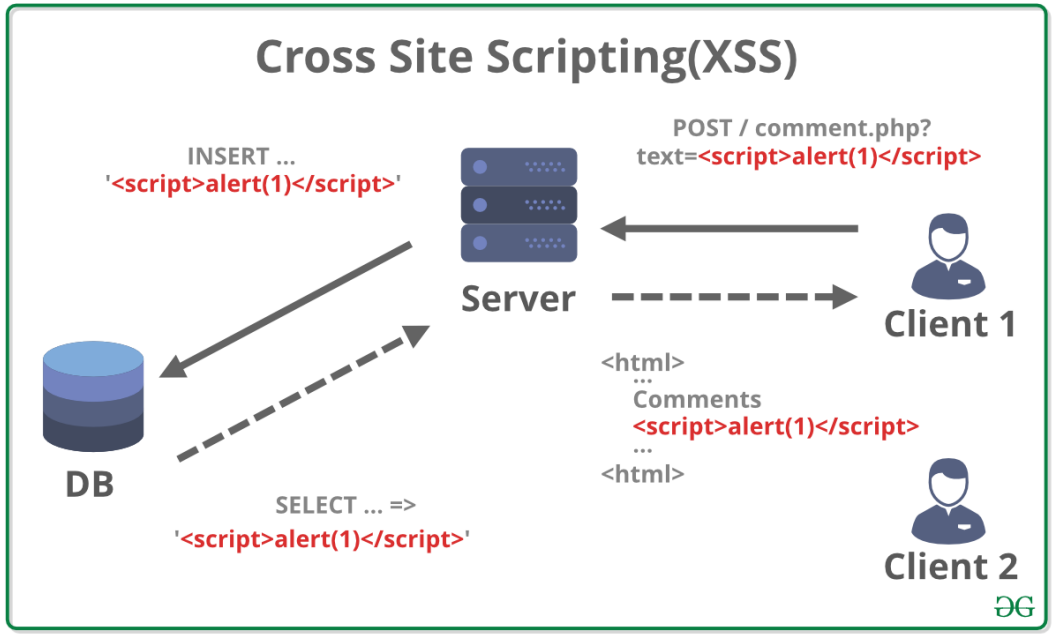
Clickjacking / Détournement de clic

- Détournement de clic, également connu sous le nom d'attaque de réparation de l'interface utilisateur, se produit lorsqu'un attaquant utilise plusieurs couches transparentes ou opaques pour inciter un utilisateur à cliquer sur un bouton ou un lien sur une autre page alors qu'il avait l'intention de cliquer sur la page de niveau supérieur. Ainsi, l'attaquant « détourne » les clics destinés à sa page et les achemine vers une autre page, très probablement détenue par une autre application, un autre domaine ou les deux.
- En utilisant une technique similaire, les frappes peuvent également être détournées. Avec une combinaison soigneusement conçue de feuilles de style, d'Iframes et de zones de texte, un utilisateur peut être amené à croire qu'il tape le mot de passe de son e-mail ou de son compte bancaire, mais qu'il tape à la place dans un cadre invisible contrôlé par l'attaquant.
  - https://owasp.org/www-community/attacks/Clickjacking
  - https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html



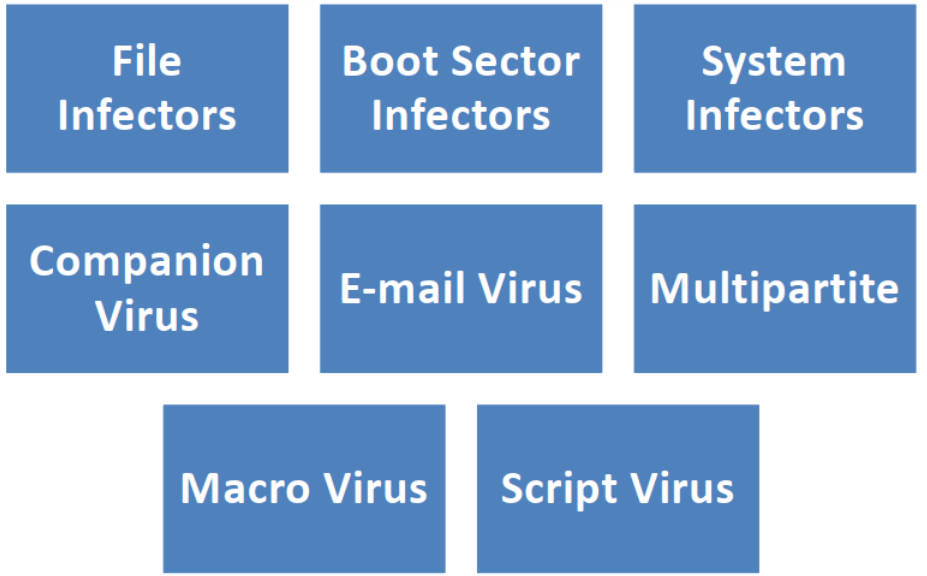
XSS / Cross-site Scripting / Exécution de code croisé

- Les attaques de type Cross-Site Scripting (XSS) sont un type d'injection, dans lequel des scripts malveillants sont injectés dans des sites Web autrement bénins et fiables.
- Les attaques XSS se produisent lorsqu'un attaquant utilise une application Web pour envoyer un code malveillant, généralement sous la forme d'un script côté navigateur, à un autre utilisateur final.
- Les failles qui permettent à ces attaques de réussir sont assez répandues et se produisent partout où une application Web utilise l'entrée d'un utilisateur dans la sortie qu'elle génère sans la valider ou l'encoder.
- Un attaquant peut utiliser XSS pour envoyer un script malveillant à un utilisateur sans méfiance. Le navigateur de l'utilisateur final n'a aucun moyen de savoir que le script ne doit pas être approuvé et exécutera le script. Parce qu'il pense que le script provient d'une source fiable, le script malveillant peut accéder à tous les cookies, jetons de session ou autres informations sensibles conservés par le navigateur et utilisés avec ce site. Ces scripts peuvent même réécrire le contenu de la page HTML. Pour plus de détails sur les différents types de failles XSS.
  - https://owasp.org/www-community/attacks/xss/
  - https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html



Virus

- Les virus de boot : Ils infectent la zone de démarrage du disque dur et sont activés lors du démarrage de l'ordinateur.
- Les virus de fichiers : Ils infectent les fichiers exécutables et se propagent lorsqu'un utilisateur lance l'application infectée.
- Les virus de macro : Ils se propagent via des macros dans des programmes tels que Microsoft Office et peuvent être activés lorsque l'utilisateur ouvre un document infecté.
- Les virus de script : Ils sont écrits en langage de script et peuvent être exécutés à partir de sites Web, de courriels ou de fichiers téléchargés.

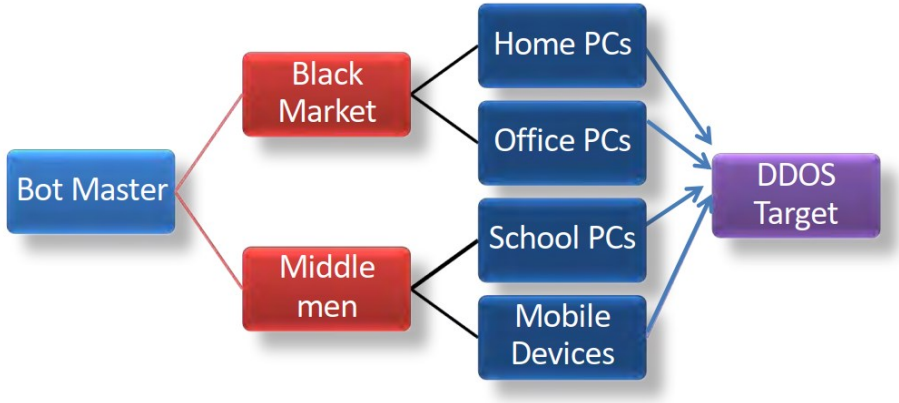


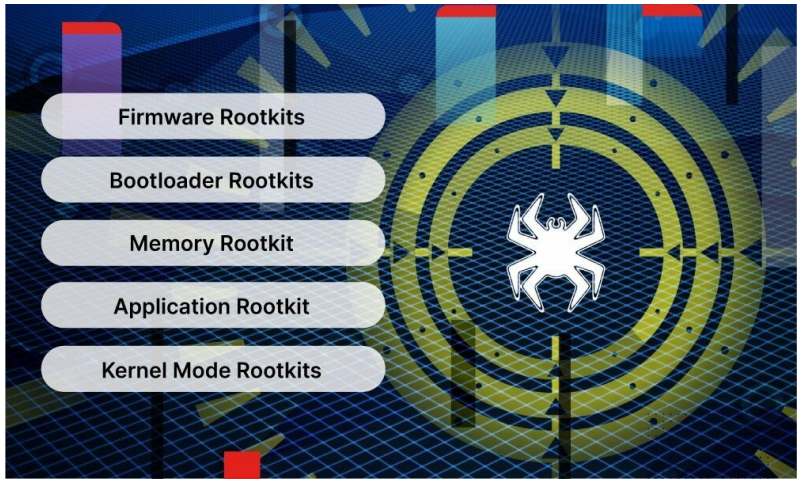
Autres types d’agents malicieux

- Les chevaux de Troie : Ils sont conçus pour se cacher dans des programmes légitimes et peuvent ouvrir une porte dérobée sur l'ordinateur de la victime pour permettre à un attaquant d'accéder à distance à l'ordinateur.
- Les vers informatiques : Ils se propagent de manière autonome sur les réseaux informatiques en exploitant les vulnérabilités du système.
- Root Kit : Ce sont des outils installés de façon malicieuse sur des machines pour permettre un accès à distance malicieuse sur la machine. L’accès peut ensuite être revendu sur le dark web.

Autres types d’agents malicieux

- Botnet : Ce sont des ordinateurs qui sont infectés par un logiciel malveillant contrôlable à distance qui rend l’ordinateur « zombie ». Les Bot peuvent être utilisé par une personne malicieuse ou encore vendu sur le dark web.
- Ransomware : Logiciel malveillant chiffrant un poste et demandant une rançon à l’utilisateur en échange de récupérer l’accès à son ordinateur.
- Root Kit : Logiciel malveillant qui cache la présence d'un intrus sur un système en modifiant ou en remplaçant des parties du système d'exploitation ou d'autres logiciels du système. Ils permettent un accès admin (root) sur le système





Comment identifié si notre système est infecté ?

IoC

- C’est un élément de preuve ou une anomalie indiquant qu’une compromission/attaque à eu lieu.
- Les IOC sont utilisés pour détecter et répondre aux incidents de sécurité informatique, en identifiant les activités suspectes ou malveillantes.
- Présentes sout différentes formes : fichiers malveillants, adresses IP suspectes, noms de domaine malveillants, signatures de virus, journaux d'événements de sécurité, comportements anormaux du réseau, etc.
- Peut être retrouvé sur différentes sources : pare-feu, IDS/IPS, antivirus, SIEM, Système de surveillance de vulnérabilités, log, etc.
- Lorsqu’identifié, ils permettent de prendre les actions nécessaires en réponse : quarantaine, blocage, isolement, isolement, restauration, etc.
- Les IOC sont importants pour la détection et la réponse aux incidents et ils sont de plus en plus utilisés et partagés dans les organisations pour améliorer la sécurité des systèmes informatiques et la protection des données.

Comment identifier si notre système est infecté ?

- Artefacts dans un réseau ou sur un hôte qui a été attaqué
  - Valeurs observables (exemples) :
    - Adresses IP ou domaines contactés par la machine victime
    - Adresses e-mail dans les messages suspects
    - Valeurs de hachage
    - Clés de registre qui ont été ajoutées ou modifiées
    - Chaînes DNS
    - Noms de fichiers nouveaux, différents ou inhabituels
  - Actions observables (exemples) :
    - Création d'une clé de registre
    - Utiliser des méthodes de persévérance
    - Tactiques et techniques des attaquants
    - Trafic réseau suspect ou inhabituel

Conclusions

Les mauvaises pratiques mènent éventuellement à des failles

Les exploitations sont déjouées par du meilleur code

Les exploitations sont déjouées (parfois causées) par des fonctions du processeur,

Les exploitations sont déjouées par des stratégies globales (meilleur code, nettoyage des données, utilisation de frameworks sécuritaires, etc.)