Introduction à la sécurité informatique appliquée avec Python

Ryan Lahfa (ryan@lahfa.xyz)

Présentation des enjeux de la sécurité informatique

Quelques statistiques: Marché

- Un marché à $\geq$ 151 Mds de dollars d'ici 2023

Quelques statistiques: Cyber-attaques

Source:
https://www.statista.com/topics/1712/information-security/

Présentation des enjeux de la sécurité informatique

Quelques statistiques: Marché

- Un marché à $\geq$ 151 Mds de dollars d'ici 2023
- En 2018, les entreprises ont dépensé au moins 58,9 Mds en produits/services de sécurité

Quelques statistiques: Cyber-attaques

Source:
https://www.statista.com/topics/1712/information-security/

Présentation des enjeux de la sécurité informatique

Quelques statistiques: Marché

- Un marché à $\geq$ 151 Mds de dollars d'ici 2023
- En 2018, les entreprises ont dépensé au moins 58,9 Mds en produits/services de sécurité
- En 2019, les produits et services de sécurité ont rapporté 124,12 Mds de dollars

Quelques statistiques: Cyber-attaques

Source:
https://www.statista.com/topics/1712/information-security/

Présentation des enjeux de la sécurité informatique

Quelques statistiques: Marché

- Un marché à $\geq$ 151 Mds de dollars d'ici 2023
- En 2018, les entreprises ont dépensé au moins 58,9 Mds en produits/services de sécurité
- En 2019, les produits et services de sécurité ont rapporté 124,12 Mds de dollars

Quelques statistiques: Cyber-attaques

- 10,52 Mds, c'est le nombre d'attaques informatique en 2018

Source:
https://www.statista.com/topics/1712/information-security/

Présentation des enjeux de la sécurité informatique

Quelques statistiques: Marché

- Un marché à $\geq$ 151 Mds de dollars d'ici 2023
- En 2018, les entreprises ont dépensé au moins 58,9 Mds en produits/services de sécurité
- En 2019, les produits et services de sécurité ont rapporté 124,12 Mds de dollars

Quelques statistiques: Cyber-attaques

- 10,52 Mds, c'est le nombre d'attaques informatique en 2018
- 27,32 M de dollars, c'est le coût moyen d'une attaque informatique aux États Unis pour une entreprise

Source:
https://www.statista.com/topics/1712/information-security/

Quelques grands moments de la sécurité informatique

- Heartbleed (2014): pouvoir lire arbitrairement la RAM d'un serveur HTTPS qui utilisait OpenSSL

Quelques grands moments de la sécurité informatique

- Heartbleed (2014): pouvoir lire arbitrairement la RAM d'un serveur HTTPS qui utilisait OpenSSL
- Shellshock (2014): devenir `root` en abusant `bash` (utilisable contre les serveurs web CGI, DHCP, qmail, SSH)

Quelques grands moments de la sécurité informatique

- Heartbleed (2014): pouvoir lire arbitrairement la RAM d'un serveur HTTPS qui utilisait OpenSSL
- Shellshock (2014): devenir `root` en abusant `bash` (utilisable contre les serveurs web CGI, DHCP, qmail, SSH)
- Stagefright (2015): devenir `root` en faisant lire un MP4 sur un téléphone

Quelques grands moments de la sécurité informatique

- Heartbleed (2014): pouvoir lire arbitrairement la RAM d'un serveur HTTPS qui utilisait OpenSSL
- Shellshock (2014): devenir `root` en abusant `bash` (utilisable contre les serveurs web CGI, DHCP, qmail, SSH)
- Stagefright (2015): devenir `root` en faisant lire un MP4 sur un téléphone
- Dirty COW (2016): devenir `root` en abusant les optimisations de la mémoire du noyau Linux

Quelques grands moments de la sécurité informatique

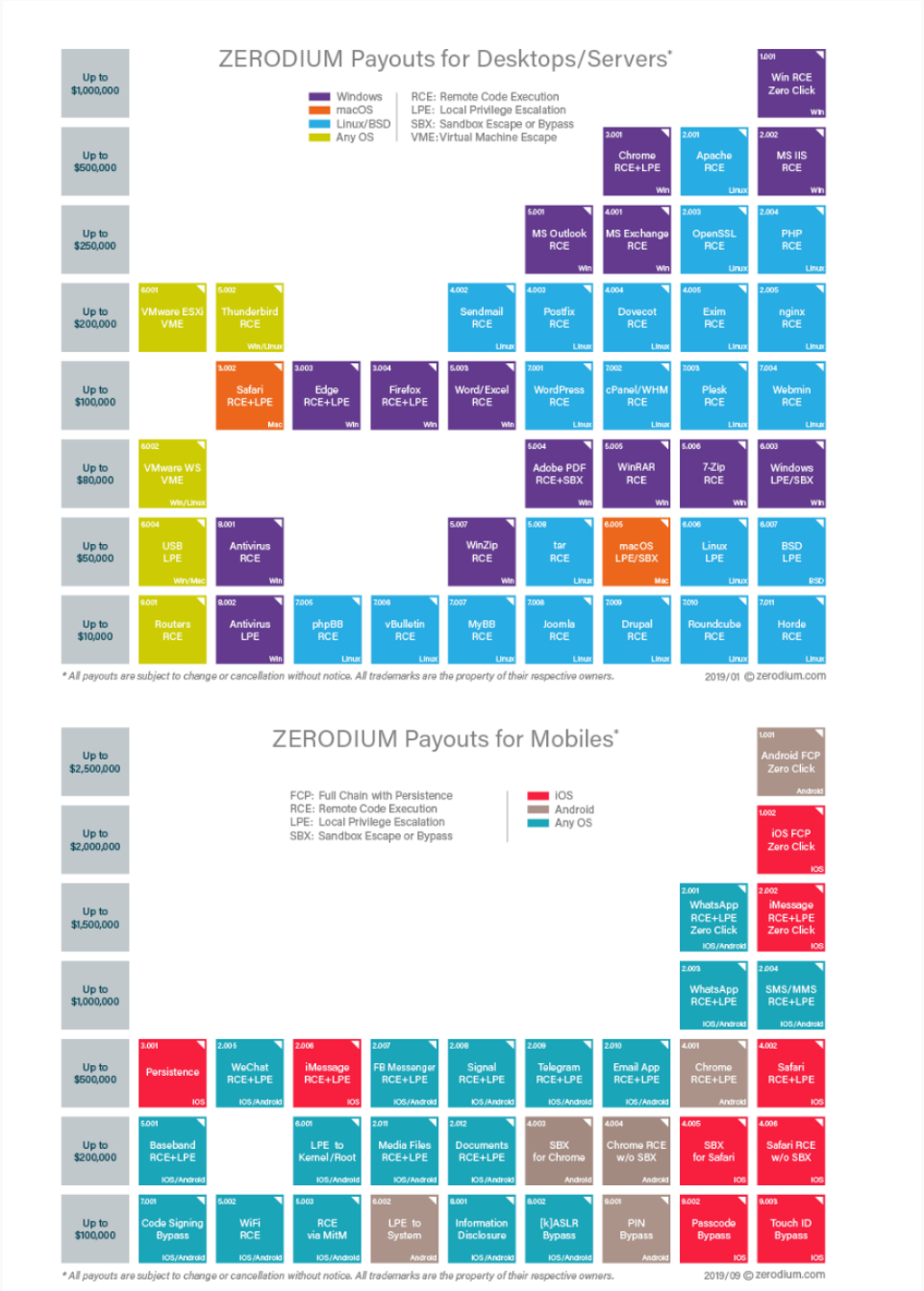
- Heartbleed (2014): pouvoir lire arbitrairement la RAM d'un serveur HTTPS qui utilisait OpenSSL
- Shellshock (2014): devenir `root` en abusant `bash` (utilisable contre les serveurs web CGI, DHCP, qmail, SSH)
- Stagefright (2015): devenir `root` en faisant lire un MP4 sur un téléphone
- Dirty COW (2016): devenir `root` en abusant les optimisations de la mémoire du noyau Linux
- EternalBlue (2017): devenir `root` à distance en abusant le protocole SMBv1

Quelques grands moments de la sécurité informatique

- Heartbleed (2014): pouvoir lire arbitrairement la RAM d'un serveur HTTPS qui utilisait OpenSSL
- Shellshock (2014): devenir `root` en abusant `bash` (utilisable contre les serveurs web CGI, DHCP, qmail, SSH)
- Stagefright (2015): devenir `root` en faisant lire un MP4 sur un téléphone
- Dirty COW (2016): devenir `root` en abusant les optimisations de la mémoire du noyau Linux
- EternalBlue (2017): devenir `root` à distance en abusant le protocole SMBv1
- Spectre & Meltdown & ZombieLoad (2018 & 2019): détruit toutes les couches d'isolations déployées par les processeurs Intel [1]

[1] On a longtemps cru qu'AMD n'était pas touché, mais si. Mais aussi ARM!

Combien ça coûte une faille ?



Quelques idées de débouchées et de parcours possibles

- Métiers de l'attaque: « red teamer », « offensive security engineer », « pentester »

Quelques idées de débouchées et de parcours possibles

- Métiers de l'attaque: « red teamer », « offensive security engineer », « pentester »
- Métiers de la défense: « blue teamer », « malware reverse engineer »

Quelques idées de débouchées et de parcours possibles

- Métiers de l'attaque: « red teamer », « offensive security engineer », « pentester »
- Métiers de la défense: « blue teamer », « malware reverse engineer »
- Intégrer des équipes d'experts en sécurité informatique: Google Project Zero

Quelques idées de débouchées et de parcours possibles

- Métiers de l'attaque: « red teamer », « offensive security engineer », « pentester »
- Métiers de la défense: « blue teamer », « malware reverse engineer »
- Intégrer des équipes d'experts en sécurité informatique: Google Project Zero
- Rejoindre des agences: les CERTs, les agences de renseignements [2], les agences de « sûreté » [3]

[2] DGSI/DGSE/NSA/GCHQ par exemple.
[3] ANSSI/ENISA/(ISC)^2 par exemple.

Quelques idées de débouchées et de parcours possibles

- Métiers de l'attaque: « red teamer », « offensive security engineer », « pentester »
- Métiers de la défense: « blue teamer », « malware reverse engineer »
- Intégrer des équipes d'experts en sécurité informatique: Google Project Zero
- Rejoindre des agences: les CERTs, les agences de renseignements [2], les agences de « sûreté » [3]
- Les entreprises de sécurité: Galois, etc.

[2] DGSI/DGSE/NSA/GCHQ par exemple.
[3] ANSSI/ENISA/(ISC)^2 par exemple.

Nouveaux enjeux, encore plus de travail

- Les limites des barrières (“boundaries”)

Nouveaux enjeux, encore plus de travail

- Les limites des barrières (“boundaries”)
- L’apparition du « cloud computing »

Nouveaux enjeux, encore plus de travail

- Les limites des barrières (“boundaries”)
- L’apparition du « cloud computing »
- Les blockchains & smart contracts

Nouveaux enjeux, encore plus de travail

Les limites des barrières (“boundaries”)

Au cours des dernières années, les navigateurs web ont pris une place démesuré sur nos ordinateurs, ils remplacent presque nos systèmes d’exploitations en procurant une quantité astronomique d’APIs systèmes (http://chromestatus.com/).

[4] Le moteur d’exécution JavaScript de Google Chrome.

[5] Le fonctionnement d’un compilateur « juste à temps » impose de pouvoir écrire dans la RAM du code à la volée, donc requiert des pages exécutables, tout le temps. C’est une limite inhérente des JITs.

Nouveaux enjeux, encore plus de travail

Les limites des barrières (“boundaries”)

Au cours des dernières années, les navigateurs web ont pris une place démesuré sur nos ordinateurs, ils remplacent presque nos systèmes d’exploitations en procurant une quantité astronomique d’APIs systèmes (http://chromestatus.com/).

Ajoutons que la dette technique (JavaScript) accumulé a forcé des miracles afin de faire tourner vite des langages mal conçus (JavaScript), ce qui a donné V8[4] mais aussi l’usage des compilateurs « juste à temps », ce qui conduit à une menace constante de franchir la barrière des sandbox puisque une bonne partie des adresses mémoires ne peuvent pas être marquées comme non-exécutables[5].

[4] Le moteur d’exécution JavaScript de Google Chrome.

[5] Le fonctionnement d’un compilateur « juste à temps » impose de pouvoir écrire dans la RAM du code à la volée, donc requiert des pages exécutables, tout le temps. C’est une limite inhérente des JITs.

Au delà de cela, JavaScript n'est pas suffisant pour tout faire mais les entreprises veulent à tout prix continuer à faire vivre leur programme dans le navigateur, d'où la naissance de Flash puis NaCl[6] notamment, qui ont été aujourd'hui tués par la plupart des acteurs du métier.

[6] Google Native Client, sandbox de code x86.

Au delà de cela, JavaScript n'est pas suffisant pour tout faire mais les entreprises veulent à tout prix continuer à faire vivre leur programme dans le navigateur, d'où la naissance de Flash puis NaCl[6] notamment, qui ont été aujourd'hui tués par la plupart des acteurs du métier.

Pour autant, le rêve se poursuit, aujourd'hui, avec WebAssembly, qui a pour destin de remplacer JavaScript et de résoudre les problèmes de sécurité et de performance qui ont conduit à l'existence du moteur V8.

[6] Google Native Client, sandbox de code x86.

Au delà de cela, JavaScript n'est pas suffisant pour tout faire mais les entreprises veulent à tout prix continuer à faire vivre leur programme dans le navigateur, d'où la naissance de Flash puis NaCl[6] notamment, qui ont été aujourd'hui tués par la plupart des acteurs du métier.

Pour autant, le rêve se poursuit, aujourd'hui, avec WebAssembly, qui a pour destin de remplacer JavaScript et de résoudre les problèmes de sécurité et de performance qui ont conduit à l'existence du moteur V8.

Cependant, c'est pas gagné avec Spectre, Meltdown & Zombie Load...

[6] Google Native Client, sandbox de code x86.

L'apparition du « cloud computing »

Le cloud computing est à la fois une arnaque et une aubaine.

[7] Oui, Amazon ne gagne pas d'argent en vendant des choses! Son e-commerce n'est là que pour justifier l'existence de son cloud.

L'apparition du « cloud computing »

Le cloud computing est à la fois une arnaque et une aubaine.

Il représente la quasi-totalité des profits d'Amazon [7], puisque ces derniers doivent constamment faire face à des montées de charge pour leur site de e-commerce et peuvent se resservir de leur infrastructure pour en louer une partie aux autres, lorsque leurs serveurs ne sont pas très demandés.

[7] Oui, Amazon ne gagne pas d'argent en vendant des choses! Son e-commerce n'est là que pour justifier l'existence de son cloud.

L'apparition du « cloud computing »

Le cloud computing est à la fois une arnaque et une aubaine.

Il représente la quasi-totalité des profits d'Amazon [7], puisque ces derniers doivent constamment faire face à des montées de charge pour leur site de e-commerce et peuvent se resservir de leur infrastructure pour en louer une partie aux autres, lorsque leurs serveurs ne sont pas très demandés.

Le problème, c'est que le cloud c'est les ordinateurs des autres, donc on doit leur faire confiance, or comment voulez vous faire confiance à un ordinateur sur lequel votre voisin de processeur est peut-être un acteur malveillant qui peut abuser une faille comme Meltdown et voler les secrets de vos machines de production.

Pire, ce problème ne s'arrête pas là, comment pouvez vous faire confiance aux entreprises avec vos données tout court ?

[7] Oui, Amazon ne gagne pas d'argent en vendant des choses! Son e-commerce n'est là que pour justifier l'existence de son cloud.

Il y a un désir ardent de modèle de calcul qui respecte la vie privée des individus, un modèle dans lequel on peut prendre vos données et effectuer des opérations sans jamais pouvoir ouvrir la boîte et lire ce qu'il y a vraiment, juste dériver des valeurs « anonymisées ».

[8] Hardware Security Modules

Il y a un désir ardent de modèle de calcul qui respecte la vie privée des individus, un modèle dans lequel on peut prendre vos données et effectuer des opérations sans jamais pouvoir ouvrir la boîte et lire ce qu'il y a vraiment, juste dériver des valeurs « anonymisées ».

Bienvenue dans le monde des composants de confiance: Intel SGX, ARM TrustZone, les HSM [8].

[8] Hardware Security Modules

Blockchain & « smart contracts »

Et maintenant, la naissance d'Ethereum, une machine virtuelle décentralisée [9] amène de nouvelles difficultés techniques.

[9] ridiculement faible en puissance.

Blockchain & « smart contracts »

Et maintenant, la naissance d'Ethereum, une machine virtuelle décentralisée [9] amène de nouvelles difficultés techniques.

Lorsque vous avez un smart contract, il est hors de question de vous contenter de garantie faible sur la qualité de votre code, voire de ne pas démontrer que votre code fait exactement ce que vous voulez qu'il fasse.

[9] ridiculement faible en puissance.

D'où l'avènement de l'analyse statique avancée et la certification de programme dans l'industrie « mainstream » avec Tezos notamment.

[10] Bien que ça a déjà été fait plein de fois. :)

D'où l'avènement de l'analyse statique avancée et la certification de programme dans l'industrie « mainstream » avec Tezos notamment.

L'enjeu est de pouvoir prouver qu'un smart contract n'introduit pas de bugs grave qui pourraient être employés par des attaquants afin de voler des millions de dollars [10]

[10] Bien que ça a déjà été fait plein de fois. :)

Présentation de Python

Pourquoi ce langage ?

- Inventé par Guido van Rossum, qui a été recruté assez vite par Google afin d'être payé pour développer Python à temps-plein

Pourquoi ce langage ?

- Inventé par Guido van Rossum, qui a été recruté assez vite par Google afin d'être payé pour développer Python à temps-plein
- Python a été l'outil de productivité d'une grande communauté de développeurs

Présentation de Python

Pourquoi ce langage ?

- Inventé par Guido van Rossum, qui a été recruté assez vite par Google afin d'être payé pour développer Python à temps-plein
- Python a été l'outil de productivité d'une grande communauté de développeurs
- Il a des quantités astronomiques de librairies

Pourquoi ce langage ?

- Inventé par Guido van Rossum, qui a été recruté assez vite par Google afin d'être payé pour développer Python à temps-plein
- Python a été l'outil de productivité d'une grande communauté de développeurs
- Il a des quantités astronomiques de librairies
- Il n'a pas de gros problèmes de concurrence en raison de la GIL [11]

[11] Global Interpreter Lock: une primitive de synchronisation qui empêche Python en général d'avoir des « race conditions »

L'industrie qui utilise Python

Une majorité des boîtes et surtout les très grandes utilisent en Python au quotidien, en particulier:

- Disney utilise intensivement Python pour complémenter ses outils d'éditions très puissants (catégoriser les rushs, etc.)

L'industrie qui utilise Python

Une majorité des boîtes et surtout les très grandes utilisent en Python au quotidien, en particulier:

- Disney utilise intensivement Python pour complémenter ses outils d'éditions très puissants (catégoriser les rushs, etc.)
- Certains équipes d'experts en sécurité informatique prototypent en Python leurs exploits: (l'équipe derrière la faille avec les VPNs)

Présentation de ce cours

Objectifs

- Vous montrer le milieu de la sécurité informatique pratique

Présentation de ce cours

Objectifs

- Vous montrer le milieu de la sécurité informatique pratique
- Vous introduire aux techniques classiques de pénétration & exfiltration des réseaux d'entreprises (« réfléchir comme un attaquant »)

Présentation de ce cours

Objectifs

- Vous montrer le milieu de la sécurité informatique pratique
- Vous introduire aux techniques classiques de pénétration & exfiltration des réseaux d'entreprises (« réfléchir comme un attaquant »)
- Vous introduire aux techniques classiques de défense des réseaux d'entreprises [12]

[12] Appliqué par tellement peu de personnes que si vous le faites, vous êtes déjà au dessus d'un bon paquet de boîtes...

Présentation de ce cours

Objectifs

- Vous montrer le milieu de la sécurité informatique pratique
- Vous introduire aux techniques classiques de pénétration & exfiltration des réseaux d'entreprises (« réfléchir comme un attaquant »)
- Vous introduire aux techniques classiques de défense des réseaux d'entreprises [12]
- Vous faire réfléchir sur les surfaces d'attaque et sur l'importance de mesurer son modèle de menace

[12] Appliqué par tellement peu de personnes que si vous le faites, vous êtes déjà au dessus d'un bon paquet de boîtes...

Plan

Voici le plan approximatif de ce cours:

- Rappels de Python (2 heures à 3 heures): base de l'OO, OO avancée, librairie standard (`collections`, `ctypes`), librairie externes (`scapy`, `pwnlib`, `dpkt`, `impacket`, Unicorn Engine)

Plan

Voici le plan approximatif de ce cours:

- Rappels de Python (2 heures à 3 heures): base de l'OO, OO avancée, librairie standard (`collections`, `ctypes`), librairie externes (`scapy`, `pwnlib`, `dpkt`, `impacket`, Unicorn Engine)
- Tromper le réseau d'une entreprise (ARP/DHCP poisoning, exploitation simple)

Plan

Voici le plan approximatif de ce cours:

- Rappels de Python (2 heures à 3 heures): base de l'OO, OO avancée, librairie standard (`collections`, `ctypes`), librairie externes (`scapy`, `pwnlib`, `dpkt`, `impacket`, Unicorn Engine)
- Tromper le réseau d'une entreprise (ARP/DHCP poisoning, exploitation simple)
- Tromper un attaquant sur le réseau (IDS/IPS, honeypot, DPI)

Plan

Voici le plan approximatif de ce cours:

- Rappels de Python (2 heures à 3 heures): base de l'OO, OO avancée, librairie standard (`collections`, `ctypes`), librairie externes (`scapy`, `pwnlib`, `dpkt`, `impacket`, Unicorn Engine)
- Tromper le réseau d'une entreprise (ARP/DHCP poisoning, exploitation simple)
- Tromper un attaquant sur le réseau (IDS/IPS, honeypot, DPI)
- Tromper un humain sur le réseau d'une entreprise (social engineering, dépôt de malware, rogue AP)

Plan

Voici le plan approximatif de ce cours:

- Rappels de Python (2 heures à 3 heures): base de l'OO, OO avancée, librairie standard (`collections`, `ctypes`), librairie externes (`scapy`, `pwnlib`, `dpkt`, `impacket`, Unicorn Engine)
- Tromper le réseau d'une entreprise (ARP/DHCP poisoning, exploitation simple)
- Tromper un attaquant sur le réseau (IDS/IPS, honeypot, DPI)
- Tromper un humain sur le réseau d'une entreprise (social engineering, dépôt de malware, rogue AP)
- Détection de signatures, émulation, antivirus (Yara/Snort, émulation d'architecture, analyse statique)

Plan

Voici le plan approximatif de ce cours:

- Rappels de Python (2 heures à 3 heures): base de l'OO, OO avancée, librairie standard (`collections`, `ctypes`), librairie externes (`scapy`, `pwnlib`, `dpkt`, `impacket`, Unicorn Engine)
- Tromper le réseau d'une entreprise (ARP/DHCP poisoning, exploitation simple)
- Tromper un attaquant sur le réseau (IDS/IPS, honeypot, DPI)
- Tromper un humain sur le réseau d'une entreprise (social engineering, dépôt de malware, rogue AP)
- Détection de signatures, émulation, antivirus (Yara/Snort, émulation d'architecture, analyse statique)
- Fabriquer les malwares, maintenir sa présence et exfiltrer discrètement toutes les données (évasion d'EDR/AV, `ctypes`/Win32 API, stéganographie, obfsproxy)

Plan

Voici le plan approximatif de ce cours:

- Rappels de Python (2 heures à 3 heures): base de l'OO, OO avancée, librairie standard (`collections`, `ctypes`), librairie externes (`scapy`, `pwnlib`, `dpkt`, `impacket`, Unicorn Engine)
- Tromper le réseau d'une entreprise (ARP/DHCP poisoning, exploitation simple)
- Tromper un attaquant sur le réseau (IDS/IPS, honeypot, DPI)
- Tromper un humain sur le réseau d'une entreprise (social engineering, dépôt de malware, rogue AP)
- Détection de signatures, émulation, antivirus (Yara/Snort, émulation d'architecture, analyse statique)
- Fabriquer les malwares, maintenir sa présence et exfiltrer discrètement toutes les données (évasion d'EDR/AV, `ctypes`/Win32 API, stéganographie, obfsproxy)
- Point récapitulatif et coordination des compétences

Déroulé

Autant que possible, les cours seront agrémentés de démonstration live dans des machines virtuelles de certaines attaques, ce qu'on appelera durant tout le long un « exemple jouet ».

Les machines virtuelles seront disponibles à un lien précisé ultérieurement et testable à la maison.

L'examination se fera essentiellement par des QCMs toutes les deux, trois parties:

- 1 QCM sur Python

En plus de cela, s'ajoute un projet.

Déroulé

Autant que possible, les cours seront agrémentés de démonstration live dans des machines virtuelles de certaines attaques, ce qu'on appelera durant tout le long un « exemple jouet ».

Les machines virtuelles seront disponibles à un lien précisé ultérieurement et testable à la maison.

L'examination se fera essentiellement par des QCMs toutes les deux, trois parties:

- 1 QCM sur Python
- 1 QCM sur les attaques & la défense réseau

En plus de cela, s'ajoute un projet.

Déroulé

Autant que possible, les cours seront agrémentés de démonstration live dans des machines virtuelles de certaines attaques, ce qu'on appelera durant tout le long un « exemple jouet ».

Les machines virtuelles seront disponibles à un lien précisé ultérieurement et testable à la maison.

L'examination se fera essentiellement par des QCMs toutes les deux, trois parties:

- 1 QCM sur Python
- 1 QCM sur les attaques & la défense réseau
- 1 QCM sur les malwares, l'émulation et un peu de cryptographie

En plus de cela, s'ajoute un projet.

Le projet

Un projet libre [13] autour des points abordés dans ce cours, hébergé sur le serveur Git de votre choix préféré [14].

L'idée est d'utiliser ce que vous apprendrez dans ce cours pour construire un outil de sécurité intéressant à partir de briques plus élémentaires, vous aurez à votre disposition une liste d'une quinzaine d'idées que vous pourrez mettre à l'œuvre.

La notation s'oriente en priorité autour de:

- La mise en œuvre de compétences techniques en sécurité

La date de rendu est autour de juillet.

[13] Dans les deux sens! Il doit être open source & libre mais aussi le choix du sujet

Le projet

Un projet libre [13] autour des points abordés dans ce cours, hébergé sur le serveur Git de votre choix préféré [14].

L'idée est d'utiliser ce que vous apprendrez dans ce cours pour construire un outil de sécurité intéressant à partir de briques plus élémentaires, vous aurez à votre disposition une liste d'une quinzaine d'idées que vous pourrez mettre à l'œuvre.

La notation s'oriente en priorité autour de:

- La mise en œuvre de compétences techniques en sécurité
- L'usage de Python (et d'autres technologies)

La date de rendu est autour de juillet.

[13] Dans les deux sens! Il doit être open source & libre mais aussi le choix du sujet

Le projet

Un projet libre [13] autour des points abordés dans ce cours, hébergé sur le serveur Git de votre choix préféré [14].

L'idée est d'utiliser ce que vous apprendrez dans ce cours pour construire un outil de sécurité intéressant à partir de briques plus élémentaires, vous aurez à votre disposition une liste d'une quinzaine d'idées que vous pourrez mettre à l'œuvre.

La notation s'oriente en priorité autour de:

- La mise en œuvre de compétences techniques en sécurité
- L'usage de Python (et d'autres technologies)
- L'intérêt pratique / théorique de votre outil

La date de rendu est autour de juillet.

[13] Dans les deux sens! Il doit être open source & libre mais aussi le choix du sujet

Le projet

Un projet libre [13] autour des points abordés dans ce cours, hébergé sur le serveur Git de votre choix préféré [14].

L'idée est d'utiliser ce que vous apprendrez dans ce cours pour construire un outil de sécurité intéressant à partir de briques plus élémentaires, vous aurez à votre disposition une liste d'une quinzaine d'idées que vous pourrez mettre à l'œuvre.

La notation s'oriente en priorité autour de:

- La mise en œuvre de compétences techniques en sécurité
- L'usage de Python (et d'autres technologies)
- L'intérêt pratique / théorique de votre outil
- Son originalité

La date de rendu est autour de juillet.

[13] Dans les deux sens! Il doit être open source & libre mais aussi le choix du sujet

Le projet

Un projet libre [13] autour des points abordés dans ce cours, hébergé sur le serveur Git de votre choix préféré [14].

L'idée est d'utiliser ce que vous apprendrez dans ce cours pour construire un outil de sécurité intéressant à partir de briques plus élémentaires, vous aurez à votre disposition une liste d'une quinzaine d'idées que vous pourrez mettre à l'œuvre.

La notation s'oriente en priorité autour de:

- La mise en œuvre de compétences techniques en sécurité
- L'usage de Python (et d'autres technologies)
- L'intérêt pratique / théorique de votre outil
- Son originalité
- Son implémentation et la facilité à laquelle on peut contribuer

La date de rendu est autour de juillet.

[13] Dans les deux sens! Il doit être open source & libre mais aussi le choix du sujet

Le projet

Un projet libre [13] autour des points abordés dans ce cours, hébergé sur le serveur Git de votre choix préféré [14].

L'idée est d'utiliser ce que vous apprendrez dans ce cours pour construire un outil de sécurité intéressant à partir de briques plus élémentaires, vous aurez à votre disposition une liste d'une quinzaine d'idées que vous pourrez mettre à l'œuvre.

La notation s'oriente en priorité autour de:

- La mise en œuvre de compétences techniques en sécurité
- L'usage de Python (et d'autres technologies)
- L'intérêt pratique / théorique de votre outil
- Son originalité
- Son implémentation et la facilité à laquelle on peut contribuer
- Sa réutilisabilité dans des contextes professionnels

La date de rendu est autour de juillet.

[13] Dans les deux sens! Il doit être open source & libre mais aussi le choix du sujet

Modèles de menaces & Surfaces d'attaque

Définition : Un modèle de menace c'est un ensemble d'hypothèses contre lequel vous prétendez être capable de vous défendre.

Si vous êtes une PME et que votre produit c'est un site d'e-commerce, votre modèle de menace c'est:

- Vos serveurs de productions font tourner les dernières versions de NGINX/Apache/etc.

Vous vous fichez des menaces étatiques, e.g. la Chine, la Russie ou alors des espions industriels, vous êtes trop petits pour vous en soucier.

Si vous êtes une PME et que votre produit c'est un site d'e-commerce, votre modèle de menace c'est:

- Vos serveurs de productions font tourner les dernières versions de NGINX/Apache/etc.
- Votre processeur de paiement est fiable

Vous vous fichez des menaces étatiques, e.g. la Chine, la Russie ou alors des espions industriels, vous êtes trop petits pour vous en soucier.

Si vous êtes une PME et que votre produit c'est un site d'e-commerce, votre modèle de menace c'est:

- Vos serveurs de productions font tourner les dernières versions de NGINX/Apache/etc.
- Votre processeur de paiement est fiable
- Le code de votre site n'introduit pas de failles évidentes

Vous vous fichez des menaces étatiques, e.g. la Chine, la Russie ou alors des espions industriels, vous êtes trop petits pour vous en soucier.

Si vous êtes une PME et que votre produit c'est un site d'e-commerce, votre modèle de menace c'est:

- Vos serveurs de productions font tourner les dernières versions de NGINX/Apache/etc.
- Votre processeur de paiement est fiable
- Le code de votre site n'introduit pas de failles évidentes
- Le backoffice et la base de données tournent derrière des VPNs [15]

Vous vous fichez des menaces étatiques, e.g. la Chine, la Russie ou alors des espions industriels, vous êtes trop petits pour vous en soucier.

[15] Anecdote: il fut un temps où un grand site d'e-commerce informatique faisait tourner son backoffice sur Internet sans protection de mot de passe...

Surfaces d'attaque

Définition : La surface d'attaque c'est l'ensemble des éléments de votre système qui sont susceptibles de posséder des failles de sécurité et dont la compromission créent un risque pour votre organisation.

[16] Un processeur de paiement.

Surfaces d'attaque

Définition : La surface d'attaque c'est l'ensemble des éléments de votre système qui sont susceptibles de posséder des failles de sécurité et dont la compromission créent un risque pour votre organisation.

Exemples : Un serveur web, un champ de texte qui n'échappe pas le JavaScript, un buffer de mémoire contrôlé par l'utilisateur, un hyperviseur, Kubernetes, les informations de connexion d'un cloud, les clefs d'API de Stripe[16].

[16] Un processeur de paiement.

La surface d'attaque et le modèle de menace sont des outils conceptuels qui fonctionnent ensemble, vous utilisez le modèle de menace et vous étudiez votre surface d'attaque afin de voir les parties de vos système qui doivent être changé ou les risques à mitiger en urgence.

[17] Vous devez défendre contre une quantité incalculables de menaces venant de partout, donc ça va coûter très cher !

La surface d'attaque et le modèle de menace sont des outils conceptuels qui fonctionnent ensemble, vous utilisez le modèle de menace et vous étudiez votre surface d'attaque afin de voir les parties de vos système qui doivent être changé ou les risques à mitiger en urgence.

En pratique, c'est aussi un outil qui permet de chiffrer les risques, plus la surface d'attaque est large et le modèle de menace est résilient, plus le coût des contre-mesures en sécurité explose. [17]

[17] Vous devez défendre contre une quantité incalculables de menaces venant de partout, donc ça va coûter très cher !