RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE
RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 55 02 80 70
https://www.ifri-uac.bj Courriel : contact@ifri-uac.bj

MÉMOIRE

pour l'obtention du

Diplôme de Master en Informatique

Option : Sécurité Informatique

Présenté par :
HUEHA Ayékossou Stanislas

Analyse et atténuation des attaques ALPACA dans le contexte de l'authentification TLS

Sous la supervision :
Dr Ing Emery ASSOGBA
Enseignant-Chercheur à l'IFRI

Membres du jury :

Pr. Eugène C. EZIN	Professeur Titulaire	IFRI,IMSP/UAC	Président
Dr. Arnaud AHOUANDJINOU	Enseignant-Chercheur	IFRI/UAC	Examinateur
Dr. Ratheil HOUNDJI	Enseignant-Chercheur	IFRI/UAC	Examinateur

Année Académique : **2019-2020**

Sommaire

Dédicace ii

Remerciements iii

Liste des abréviations iv

Liste des tableaux v

Table des figures vi

Résumé ix

Abstract xi

Introduction 1

1 Généralités et concepts de base 4

2 Conception et matériels 13

3 Résultats 31

4 Discussion 41

Conclusion générale et perspectives 46

Bibliographie 48

Webographie 49

Table des matières 50

Dédicace

A l'Eternel Dieu tout puissant,
Sans qui nous n'aurions pas suivi cette formation. Nous lui rendons grâce de tout notre cœur ; qu'il fructifie ce travail.
A notre père HUEHA Paulin
Votre perspicacité nous a permis de nous forger une identité.
A nos frères et sœurs,
A qui nous renouvelons ici l'expression de nos affections. Vos soutiens de tout genre nous ont fortifié tout au long de nos années de formation. Trouvez dans ce travail toutes nos gratitudes.
A tous ceux et celles qui nous estiment ou nous aiment,
Cette œuvre est également la vôtre.
A nos proches et à ceux qui nous sont chers,
A tous ceux qui, de près ou de loin, ont contribué à faire de nous ce que nous sommes.

Remerciements

Nous exprimons notre profonde gratitude aux personnes physiques et institutions qui ont contribué à rendre meilleur notre travail, notamment :

— **A Dieu le Père**, pour nous avoir protégé et donné le courage nécessaire. Que toutes les Nations le glorifient et le louent à jamais.

— **A Mon Père HUEHA Paulin**, pour ses prières sans cesse renouvelées, aux intentions de cette formation.

— **A mes maman GANDAHO Cécile et AGBADJA Julienne**, pour leur soutien

— **Au Dr Emery ASSOGBA**, notre Maître de mémoire et enseignant chercheur à IFRI, pour sa sollicitude, sa disponibilité et surtout pour ses orientations dans la réalisation de ce travail.

— **Au Pr Eugène EZIN**, pour sa disponibilité et surtout pour ses conseils dans la réalisation de ce travail.

— **Au Dr Gaston EDA**, Directeur Adjoint de l'IFRI-UAC, pour son attention particulière à notre promotion, sa disponibilité et surtout ses conseils pour la réussite de notre formation.

— **A tous nos formateurs de l'IFRI**, pour le transfert de leur savoir et le partage de leurs expériences. Qu'ils soient assurés de notre reconnaissance.

Liste des abréviations

TLS : Transport Layer Security

HTTP : Hyper Text Transfer Protocol

SMTP : Simple Mail Transfer Protocol

IMAP : Internet Message Access Protocol

POP3 : Post Office Protocol

FTP : File Transfer Protocol

STARTTLS : Opportunistic TLS

RFC : Request For Comments

DNS : Domain Name System

UDP : User Datagram Protocol

FQDN : Fully Qualified Domain Name

TCP : Transport Control Protocol

MITM : Man In The Middle

SSL : Secure Socket Layer

FAI : Fournisseur d'Acces à Internet

URL : Uniform Resource Locator

vsftp : Very Secure FTP

HTTPS : Hyper Text Transfer Protocol Secure

CVE : Common Vulnerabilities and Exposures

CWE : Common Weakness Enumeration

Liste des tableaux

2.1 Liste de matériels 30

Table des figures

2.1 Architecture réseau de l'infrastruture du Laboratoire 15
2.2 Page d'accueil target.com affichant risque de sécurité 18
2.3 Target.com cookie de sécurité affiché 18
2.4 Activation de mode de sécurité sur le proxy pour les attaques 18
2.5 Alerte de sécurité sur attacker.com 19
2.6 Page attacker.com acceptation de risque 19
2.7 Page attacker.com avec différents types d'attaques interprotocoles 19
2.8 Page web target.com innacessible ; attaque TLS 20

Glossaire

TLS : Le TLS est une version plus récente et plus sécurisée du SSL.

Protocole : Dans le domaine des réseaux, un protocole est un ensemble de règles permettant de formater et de traiter les données. Les protocoles de réseau sont comme un langage commun pour les ordinateurs.

Couche application : La couche application n'héberge pas du tout l'application. Elle est là pour rendre des services aux applications placées au-dessus d'elles. En effet, nombre d'applications ont besoin d'échanger des fichiers, d'être accessibles à distance par des terminaux en mode caractères, de se baser sur des carnets d'adresses pour identifier les utilisateurs, d'échanger des messages.En résumé, la couche Application est une couche qui offre aux applications des sous-programmes et protocoles standards. Les utiliser, plutôt que d'intégrer les fonctions dans un développement spécifique, offre des avantages en terme de portabilité des processus, d'homogénéité d'architecture, de gains en développement

FQDN : Un nom de domaine pleinement qualifié ou FQDN est l'URL complète d'un certain site ou serveur. L'utilisation du FQDN d'un site est souvent plus fiable que l'utilisation de son adresse IP ou, dans certains cas, de son nom de domaine partiellement qualifié.

TCP : TCP (qui signifie Transmission Control Protocol, soit en français : Protocole de Contrôle de Transmission) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP). Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 (Pour savoir que le protocole en amont est TCP...). TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission.

UDP : Le protocole de datagramme utilisateur, ou UDP, est un protocole de communication utilisé sur Internet pour les transmissions particulièrement sensibles au temps, telles que la lecture de vidéos ou les recherches DNS. Il accélère les communications en n'établissant pas formellement une connexion avant le transfert des données. Cela permet de transférer des données très rapidement, mais cela peut également entraîner la perte de paquets en transit et créer des opportunités d'exploitation sous la forme d'attaques DDoS.

Drop ChangeCipherSpec : Vulnérabilité de divulgation d'informations dans OpenSSL. La vulnérabilité est due à une validation d'entrée insuffisante dans l'application

lors de la gestion d'un message de spécification de chiffrement de modification SSL contrefait. Un attaquant intermédiaire peut exploiter cela pour déchiffrer et lire le trafic ssl à partir d'un serveur vulnérable en injectant des requêtes SSL contrefaites.

MITM : Une attaque de type "man-in-the-middle" (MitM), par définition, est une forme d'écoute et de vol de données où un attaquant intercepte les données d'un expéditeur à un destinataire, puis du destinataire à l'expéditeur.

SSL : Le SSL est une technologie standard de sécurisation des connexions Internet par le chiffrement des données transitant entre un navigateur et un site web (ou entre deux serveurs). Durant leur transfert, les données (personnelles, financières) sont ainsi protégées des hackers qui ne peuvent ni les voir ni les détourner.

algorithme DES : Il s'agit d'un système de chiffrement symétrique par blocs de 64 bits, dont 8 bits (un octet) servent de test de parité (pour vérifier l'intégrité de la clé). Chaque bit de parité de la clé (1 tous les 8 bits) sert à tester un des octets de la clé par parité impaire, c'est-à-dire que chacun des bits de parité est ajusté de façon à avoir un nombre impair de '1' dans l'octet à qui il appartient. La clé possède donc une longueur « utile » de 56 bits, ce qui signifie que seuls 56 bits servent réellement dans l'algorithme. L'algorithme consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le déchiffrement). La combinaison entre substitutions et permutations est appelée code produit.

Application : En informatique, l'application est un programme informatique employé afin d'effectuer une action ou plusieurs actions dans une même thématique. Les applications sont très populaires dans le domaine informatique comme par exemple, les jeux vidéo, les navigateurs web et les boîtes de messagerie. L'exécution des applications se fait en employant le système d'exploitation de l'ordinateur.

Librairie : En informatique, une bibliothèque ou librairie logicielle (ou encore, bibliothèque de programmes) est un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire.

Web : Le web est un système de pages, raccordées entre elles par des hyperliens. Il n'est pas rare d'entendre dire que le web et Internet, c'est pareil. Les deux notions sont parfois confondues, mais ce sont différents sujets.

Résumé

Plusieurs modèles d'attaquants classiques existent pour faire des attaques sur TLS ; des modèles d'attaques telles que BEAST, CRIME, POODLE et Lucky. Avec le Man-In-The-Middle, les attaquants sont en mesure de déclencher et faire des attaques sur internet visant plusieurs services tel que le HTTP ; le web et services relatifs. En utilisant ce type d'attaque, l'attaquant redirige les requêtes sur Internet vers d'autres services créant ainsi des confusions. Pendant les attaques, les composants des services tel que serveur et applications fonctionnent comme prévu et que la violation de la sécurité est une propriété émergente due à la capacité de l'attaquant à recombiner les composants d'une manière non intentionnelle. La cause fondamentale est l'incapacité du mécanisme d'authentification TLS à empêcher la confusion en premier lieu. TLS est largement utilisé pour ajouter la confidentialité, l'authenticité et l'intégrité aux protocoles de la couche application tels que HTTP, SMTP, IMAP, POP3 et FTP. Cependant, TLS ne lie pas une connexion TCP au protocole de couche d'application prévu. Cela permet à un attaquant de type "man-in-the-middle" de rediriger le trafic TLS vers un autre point de terminaison du service TLS sur une autre adresse IP et/ou un autre port. Par exemple, si des sous-domaines partagent un certificat joker, un attaquant peut rediriger le trafic d'un sous-domaine vers un autre, ce qui donne lieu à une session TLS valide. L'authentification de TLS est ainsi rompue et des attaques inter-protocoles sont possibles, le comportement d'un service pouvant compromettre l'intégrité de la session, la sécurité de l'autre au niveau de la couche application. Dans ce document, nous avions étudié les attaques inter-protocoles sur TLS en général et avons mené une étude de cas systématique sur les serveurs web, en redirigeant les requêtes HTTPS du navigateur web d'une victime vers des serveurs SMTP, IMAP, POP3 et FTP en utilisant un lab avec lequel nous avions présenté l'attaque alpaca avec des exemples de code à l'appui pour éttayer.Nous avions évoqué les débuts de l'attaque alpaca et comment TLS est combiné avec d'autres protocoles pour assurer la sécurité, une facon de faire contourné grâce à l'attaque alpaca. Avec le Lab mis en place, nous avions montré que dans des scénarios, avec alpaca attaque, l'attaquant peut extraire des cookies de session et d'autres données privées de l'utilisateur ou exécuter du JavaScript arbitraire dans le contexte du serveur web vulnérable, contournant ainsi TLS et la sécurité des applications web. Il a aussi été question des réelles conséquences de alpaca attaque à travers son CVE et les CWE qui s'y rapportent. Nous avions aussi vu de long en large le fonctionnement du protocole TCP/IP et clarifier comment alpaca attaque peut facilement compromettre les serveurs et sommes penchés sur les sites web ; ces derniers utilisant le protocol HTTP ou encore le HTTPS. Avec l'outil Wireshark qui est un renifleur nous avions analysé deux sites au Bénin qui en prenant en comptes le CVE de alpaca et les CWE qui s'y rapportent et aussi le fonctionnement du protocol TCP/IP sont vulnérables à l'attaque alpaca. L'attaque alpaca ne s'arrête pas sur les sites web au Bénin mais aussi élargir sur les infrastructures réseaux qui ne respectent pas les standards, les normes internationnales ; ce qui faire grimper les cas de cyber criminalité au Bénin. Cepandant le Bénin fait un grand effort avec des agences et organes mis en place pour

lutter contre la cyber criminalité.

Abstract

Several classic attacker models exist for attacks on TLS; attack models such as BEAST, CRIME, POODLE and Lucky. With Man-In-The-Middle, attackers are able to trigger and carry out attacks on the Internet, targeting multiple services such as HTTP, web and related services. Using this type of attack, the attacker redirects Internet requests to other services, creating confusion. During the attack, service components such as servers and applications work as intended, and the security breach is an emergent property due to the attacker's ability to to recombine the components in an unintended way. The root cause is the inability of the TLS authentication mechanism to prevent confusion in the first place. TLS is widely used to add confidentiality, authenticity and integrity to layer protocols application such as HTTP, SMTP, IMAP, POP3 and FTP. However, TLS does not bind a TCP connection to the intended application layer protocol. This allows a "man-in-the-middle" type attacker to redirect TLS traffic to another endpoint of the TLS service on another IP address and/or another port. For example, if subdomains share a wildcard certificate, an attacker can redirect traffic from a subdomain to another, which gives rise to a valid TLS session. TLS authentication is thus broken and inter-protocol attacks are possible, the behavior of a service can compromise the integrity of the session, the security of the other at the application layer level. In this document, we had studied inter-protocol attacks on TLS in general and are conducting a study of systematic case on web servers, by redirecting HTTPS requests from a victim's web browser to SMTP, IMAP, POP3 and FTP servers using a lab with which we had presented the alpaca attack with supporting code examples to try.We had mentioned the beginnings of the alpaca attack and how TLS is combined with other protocols to ensure security, a way to get around thanks to the alpaca attack. With the Lab set up, we had shown that in scenarios, with alpaca attack, the attacker can extract session cookies and other private data from the user or execute arbitrary JavaScript in the context of the vulnerable web server, thus bypassing TLS and web application security. It was also discussed the real consequences of alpaca attack through its CVE and the CWE related to it. We had also seen first-hand the operation of the TCP/IP protocol and clarified how alpaca attack can easily compromise servers and looked at websites; the latter using the HTTP protocol or HTTPS. With the Wireshark tool which is a sniffer we had analyzed two sites in Benin which, taking into account the CVE of alpaca and the CWE related to it and also the operation of the TCP / IP protocol, are vulnerable to the alpaca attack. The alpaca attack does not stop on websites in Benin but also on network infrastructures that do not respect standards, international standards; which increases the cases of cyber crime in Benin. However, Benin is making a great effort with agencies and bodies set up to fight cyber crime.

Introduction Générale

Les sécurité des systèmes d'information devient un besoin d'une urgence capitale pour la majorité des entreprises. Ce fait est d'autant plus vrai que les systèmes d'information deviennent de plus en plus complexes augmentant les risques auxquels ils sont exposés. Le gouvernement du Bénin s'est fixé comme objectif d'inclure la sécurité des systèmes d'information dans tous les projets des services publics en mettant l'accent sur la protection des services critiques de l'état. L'un des risques qui menace ces services est l'attaque inter-protocole. Une mauvaise configuration d'un protocole réseau peut servir de vecteur d'attaque vers un autre protocole réseau de la même organisation, exposant ainsi les données sensibles de celle-ci. Ce genre d'attaque cible le plus souvent le protocole TLS. En général, les attaques inter-protocoles peuvent être envisagées entre deux protocoles sécurisés par TLS. Bien que certaines combinaisons de protocoles soient plus susceptibles de mener à des attaques réussies que d'autres, même des formats de données très différents peuvent être inter-opérables. Par exemple, un protocole binaire à longueur spécifiée peut être intégré à HTTP. Le nombre de scénarios d'attaque potentiels connaît une croissance quadratique avec le nombre de protocoles d'application et d'implémentations. Les attaques inter-protocoles sur HTTP non protégé ont été décrites pour la première fois par Jochen Topf. La première attaque inter protocole sur un serveur web sécurisé par TLS a été découverte par Jann Horn (avec l'aide de Michał Zalewski), qui a démontré des attaques Cross-Site-Scripting (XSS) réfléchies et stockées en utilisant un serveur FTP exploitable. Horn a considéré un attaquant MitM qui a également des privilèges d'homme dans le navigateur (MitB) en servant une page web avec du JavaScript malveillant à la victime.

0.1 Problématique

Il existe des modèles d'attaquant classique pour les attaques sur la couche d'enregistrement TLS, telles que BEAST , CRIME, POODLE et Lucky. À l'aide du MitB, l'attaquant déclenche une requête HTTPS POST vers le serveur Web cible, dont le corps contient des

commandes FTP valides. En utilisant le MitM, l'attaquant redirige la requête vers un serveur FTP avec un certificat compatible. Le navigateur termine la poignée de main TLS avec le serveur FTP et envoie la requête HTTP en tant que données d'application. Si le serveur FTP est tolérant aux erreurs, il peut ignorer les données invalides telles que l'en-tête HTTP et exécuter les commandes FTP valides dans le corps HTTP. Un serveur FTP exploitable renvoie le JavaScript intégré au client dans un message d'erreur. Bien que la réponse du serveur FTP ne soit pas un HTTP valide, un navigateur exploitable trouve le JavaScript en reniflant le contenu et l'exécute dans le contexte de la demande originale, complétant ainsi l'attaque. Ce genre de scénario reste possible à cause d'une mauvaise configuration des services et des protocoles. Ce mémoire vise donc à donner un prototype aux équipes rouge et bleu des organisation afin que ceux-ci sensibilises plus facilement les administrateurs systèmes.

0.2 Contexte et justification

Au Bénin on note beaucoup d'avancés dans l'amélioration de la sécurité du cyberespace. On peut citer la mise en oeuvre de la PSSIE, la création et l'opérationnalisation de l'Autorité de protection des données à caractère personnel (APDP), le centre national d'investigation numérique (CNIN). Néanmoins le chantier est vaste et il reste beaucoup à faire pour l'amélioration contitnue du SMSI national. Dans ce cadre la sécurisation des services critiques de l'état est une priorité. Pour cela il s'avère important de comprendre les attaques ciblants les services critiques et faire des recommandations. Ce mémoire se focalise sur les attaques inter-protocole. C'est une forme d'attaque réseau qui fait intervenir chaque composant (navigateur, serveur web et serveur d'application). La cause fondamentale est l'incapacité du mécanisme d'authentification TLS à empêcher la confusion en premier lieu.

Dans ce travail, nous étudions les attaques inter-protocoles sur TLS en général et sur HTTPS en particulier avec une étude de cas, où nous ciblons un serveur web pour lequel l'utilisateur peut avoir établi une sorte de session privilégiée (c'est-à-dire que l'utilisateur est connecté à un compte).

Il est aussi cruciale pour nous de se pencher à l'échelle national, le Bénin qui est un pays dont l'accès à la connexion Internet est difficile ; l'infrastructure des réseaux à revoir car la connexion Internet lente occasionnant l'inaccessibilité des sites internet, sites web, des cas qui sont vraiment une entré à l'attaque alpaca sans pour autant oublier l'effort du Bénin et que faire dans se sens.

0.3 Objectifs

Dans le cadre de cette étude, notre but est de:

— présenter le fonctionnement de l'authentification TLS
— analyser les attaques de type ALPACA
— proposer et détailler une solution choisie parmi des pistes de solution existantes pour atténuer les cracks dans l'authentification TLS
— proposer des solutions sur les cas d'attaques sur internet au Bénin.

0.4 Organisation du mémoire

Le présent mémoire fait le point de nos travaux et comporte quatre (4) chapitres. Le premier chapitre porte sur les généralités et l'état de l'art. Le second chapitre abordera les matériels et méthodes. Le troisième chapitre sera consacré aux résultats. Le quatrième présentera les discussions.

Chapitre 1

Généralités et concepts de base

1.1 TLS et certificats X.509

Le protocole TLS (Transport Layer Security) est une couche cryptographique située entre la couche transport (c'est-à-dire TCP) et un protocole de couche application. Le protocole TLS se compose de deux phases. Dans la première phase, le client et le serveur effectuent une poignée de main TLS pour échanger les versions utilisées, les aléas, les algorithmes cryptographiques et les extensions prises en charge, afin d'obtenir des clés symétriques. Dans la deuxième phase, les clés symétriques sont utilisées pour protéger les données des applications, telles que HTTP, SMTP, IMAP, POP3 ou FTP. [1] [2]

1.1.1 STARTTLS

Les protocoles existants non protégés peuvent être étendus pour prendre en charge TLS en mettant à niveau une connexion en clair à l'aide d'une commande STARTTLS spécifique au protocole. Une fois la poignée de main TLS réussie, le protocole existant est poursuivi dans les données chiffrées de l'application. STARTTLS a été normalisé pour la première fois dans la RFC 2487 comme une extension de SMTP. [1] [2]

1.1.2 Certificats de serveur

Dans un scénario TLS typique sur le Web, un serveur s'authentifie auprès d'un client TLS à l'aide d'un certificat X.509 pendant la poignée de main. Le certificat contient la clé publique du serveur (qui est utilisée dans la poignée de main), le nom de domaine du serveur, la date d'expiration et plusieurs extensions. Par exemple, il existe des extensions pour définir l'utilisation de la clé (par exemple, la signature et le chiffrement), l'utilisation étendue de la clé (par exemple, la protection du serveur WWW) et la localisation de la liste de révocation

du certificat. Pour la sécurité de la connexion, il est crucial que le client TLS valide toutes ces propriétés du certificat. [1] [2]

1.1.3 Indication du nom du serveur (SNI)

Dans certains déploiements, plusieurs services Web (ou autres) peuvent être hébergés à la même adresse IP et au même port. Pour prendre en charge cette configuration d'hébergement virtuel, le client indique le nom d'hôte souhaité dans l'extension SNI (Server Name Indication). Si le SNI est bien supporté par les serveurs HTTPS, il est beaucoup moins courant dans d'autres protocoles d'application tels que SMTP, IMAP, POP3 et FTP. [1] [2]

1.1.4 Négociation du protocole de la couche application (ALPN)

L'extension ALPN permet aux pairs TLS de sélectionner un protocole de couche d'application spécifique. Avec ALPN, un serveur web peut offrir différents protocoles de couche d'application sur le même port, par exemple, des versions plus performantes du protocole HTTP (en particulier, HTTP/2 et HTTP/1.1), tout en évitant des allers-retours supplémentaires pour la négociation du protocole. Le client envoie au serveur les protocoles qu'il prend en charge sous la forme d'une liste de chaînes de caractères, et le serveur sélectionne un protocole qu'il prend en charge ou envoie une alerte si aucun protocole pris en charge par le serveur n'est trouvé dans la liste. Les noms de protocole sont présentés en ASCII et attribués par l'IANA. Les noms pour HTTP, IMAP, POP3 et FTP ont déjà été normalisés, tandis que SMTP n'est pas encore enregistré. [1]

1.1.5 Certificats X509 v3

. L'objet d'un certificat est de certifier une clé publique. Pour utiliser une clé publique il faut être sûr de l'identité de son détenteur. La première méthode est d'avoir une relation de confiance directe avec son détenteur, comme dans le modèle "web of trust" associé à PGP ou GPG. La deuxième méthode repose sur le principe que tous les interlocuteurs ont confiance en un tiers, qui certifiera les clés en les signant. X509 est un standard définissant le format des certificats émis par ces tiers, ou émetteurs de certificats (Certificate Authority, CA en anglais). Il est à noter que l'utilisation de certificats ne fait que reporter la question de la confiance (en bout de chaîne, il y a forcément un émetteur qui s'auto-certifie, en qui la communauté cible des utilisateurs doit avoir confiance). Il appartient aux émetteurs de certificats, ou autorités de certification, de s'assurer que telle clé appartient bien à telle personne (ou entité), et de publier les procédures définissant cette assurance (politiques de certification). Lorsqu'un groupe précis d'utilisateurs n'a pas la main sur l'AC ou la PKI (ou encore ICP, Infrastructure à Clés Publiques), on peut émettre des réserves sur la sécurité de l'émission et de la gestion des certificats. Les émetteurs de certificats sur Internet ne sont pas, selon moi,

la panacée. En particulier lorsqu'ils n'émanent pas d'institutions légitimes, en qui l'on peut avoir confiance. Lorsqu'en plus ces émetteurs se retrouvent en situation de quasi monopole.

Les certificats X509, version 3, sont utilisés (entre autres) dans les applications, protocoles et standards suivants : SSL/TLS, IPSec, S/MIME, SET, chiffrement, authentification, signature, non-répudiation, horodatage. eux. [4]

1.1.6 Rôles des certificats X.509 dans l'authentification et la confidentialité des communications

Les certificats X.509 jouent un rôle central dans l'authentification et la confidentialité des communications dans TLS. Un certificat X.509 est un fichier numérique qui lie une entité (comme un site web) à une clé publique. Il est émis par une autorité de certification (AC) de confiance, et il est utilisé pour vérifier l'identité de l'entité lors de l'établissement d'une connexion TLS. Lorsqu'un client se connecte à un serveur TLS, le serveur envoie son certificat X.509 au client. Le client peut alors vérifier l'authenticité du certificat en vérifiant la signature numérique à l'aide de la clé publique de l'AC. Si la vérification réussit, le client a maintenant la certitude qu'il communique avec le serveur légitime et peut poursuivre la négociation de la session TLS. [9] Le certificat X.509 contient également des informations sur le propriétaire du certificat, telles que le nom de domaine du site web, le nom de l'entité ou de l'organisation, ainsi que la période de validité du certificat. Ces informations aident à établir la confiance dans l'entité avec laquelle le client communique.

1.2 Protocoles de la couche d'application

Il existe plusieurs protocoles qui fonctionnent pour les utilisateurs dans la couche d'application. Les protocoles de la couche application peuvent être divisés en deux catégories : les protocoles qui sont utilisés par les utilisateurs, par exemple le courrier électronique (eMail). Les protocoles qui aident et supportent les protocoles utilisés par les utilisateurs, par exemple le DNS. Quelques-uns des protocoles de la couche application sont décrits ci-dessous : Système de nom de domaine ; le système de nom de domaine (DNS) fonctionne sur le modèle client-serveur. Il utilise le protocole UDP pour la communication de la couche transport. Le DNS utilise un schéma de dénomination hiérarchique basé sur le domaine. Le serveur DNS est configuré avec des noms de domaine entièrement qualifiés (FQDN) et des adresses de courrier électronique associées à leurs adresses de protocole Internet respectives. Un serveur DNS est sollicité avec un FQDN et il répond avec l'adresse IP qui lui correspond. Le DNS utilise le port UDP 53. Protocole de transfert de courrier simple, le protocole SMTP (Simple Mail Transfer Protocol) est utilisé pour transférer du courrier électronique d'un utilisateur à

un autre. Cette tâche est effectuée au moyen d'un logiciel client de messagerie (agents utilisateurs) que l'utilisateur utilise. Les agents utilisateurs aident l'utilisateur à taper et formater le courrier électronique et à le stocker jusqu'à ce qu'Internet soit disponible. Lorsqu'un courriel est soumis pour être envoyé, le processus d'envoi est géré par l'agent de transfert de messages qui est normalement intégré au logiciel client de messagerie. L'agent de transfert de messages utilise le protocole SMTP pour transmettre l'e-mail à un autre agent de transfert de messages (côté serveur). Alors que le SMTP est utilisé par l'utilisateur final uniquement pour envoyer des e-mails, les serveurs utilisent normalement le SMTP pour envoyer et recevoir des e-mails. Le SMTP utilise les ports TCP 25 et 587. Le logiciel client utilise les protocoles IMAP (Internet Message Access Protocol) ou POP pour recevoir les e-mails. Protocole de transfert de fichiers, le protocole de transfert de fichiers (FTP) est le protocole le plus largement utilisé pour le transfert de fichiers sur le réseau. FTP utilise le protocole TCP/IP pour la communication et fonctionne sur le port TCP 21. FTP fonctionne sur le modèle client/serveur où un client demande un fichier au serveur et le serveur renvoie la ressource demandée au client. FTP utilise le contrôle hors bande, c'est-à-dire que FTP utilise le port TCP 20 pour échanger des informations de contrôle et les données réelles sont envoyées sur le port TCP 21. Protocole Post Office (POP) ; le Post Office Protocol version 3 (POP 3) est un protocole simple de récupération de courrier utilisé par les agents utilisateurs (logiciel de messagerie client) pour récupérer les courriers du serveur de messagerie. [1] Lorsqu'un client a besoin de récupérer des messages du serveur, il ouvre une connexion avec le serveur sur le port TCP 110. L'utilisateur peut alors accéder à ses messages et les télécharger sur son ordinateur local. POP3 fonctionne en deux modes. Le mode le plus courant, le mode suppression, consiste à supprimer les e-mails du serveur distant après leur téléchargement sur les machines locales. Le second mode, le mode de conservation, ne supprime pas les e-mails du serveur de messagerie et donne à l'utilisateur la possibilité d'accéder ultérieurement aux e-mails sur le serveur de messagerie. Protocole de transfert hypertexte (HTTP) ; le protocole de transfert hypertexte (HTTP) est la base du World Wide Web. L'hypertexte est un système de documentation bien organisé qui utilise des hyperliens pour relier les pages des documents texte. Le protocole HTTP fonctionne sur le modèle client-serveur. Lorsqu'un utilisateur souhaite accéder à une page HTTP sur l'Internet, la machine cliente de l'utilisateur établit une connexion TCP avec le serveur sur le port 80. Pour accéder aux pages web, un client utilise normalement des navigateurs web, qui sont responsables de l'établissement, du maintien et de la fermeture des connexions TCP. HTTP est un protocole sans état, ce qui signifie que le serveur ne conserve aucune information sur les demandes précédentes des clients.[1]

1.3 Attaques inter-protocoles basées sur TLS

L'attaque Drop ChangeCipherSpec est une faiblesse de SSL2.0 découverte par David Wagner et Bruce Schneier. Le message ChangeCipherSpec est utilisé pour notifier aux deux parties de la communication de mettre à jour l'état de la session en fonction des paramètres négociés lors de la phase de prise de contact. Lorsque la phase de poignée de main initiale est terminée, le client et le serveur échangent un message ChangeCipherSpec pour signaler à l'autre partie qu'à l'avenir, toutes les communications se feront uniquement avec les paramètres convenus. Cependant, avant qu'un côté n'envoie le message ChangeCipherSpec, l'attaquant MITM peut envoyer le message Finished à l'autre côté, envoyer le message Finished à l'autre partie, ce qui aurait pour effet de faire démarrer la communication sans aucun changement ni adoption des paramètres de sécurité convenus, ou il peut simplement supprimer le message ChangeCipherSpec de sorte que le client et le serveur ne pourront jamais établir de communication. La solution à ce problème consiste à obliger les deux parties à s'assurer qu'un message ChangeCipherSpec est reçu avant d'accepter le message Finished. L'attaque par retour de version est une vulnérabilité de SSL 3.0. Il s'agit d'un type d'attaque où l'attaquant peut amener le client et le serveur de la communication à utiliser une version inférieure du protocole SSL2.0 au lieu de celle qu'ils devraient utiliser et qui supporte SSL3.0. L'attaquant modifie le message ClientHello SSL3.0, afin qu'il ressemble au message ClientHello SSL2.0. De cette façon, il peut utiliser les faiblesses du protocole SSL2.0, comme une suite de chiffrement plus faible des algorithmes proposés (c'est-à-dire une liste avec l'algorithme DES ce qui permet à un attaquant d'effectuer une attaque par force brute et de compromettre les données données sensibles).[1]

1.4 Attaques inter-protocoles sur HTTPS

Nous profitons tous de l'incroyable diversité des sites Web sur Internet. Divertissement, information, inspiration, prestations de service et bien d'autres encore sont apparemment disponibles en quantités illimitées. Mais malheureusement, tous les sites Web ne sont pas bénins. Il existe, comme en dehors du monde numérique, des hommes d'affaires douteux, des cybercriminels et du crime organisé. C'est ainsi que des utilisateurs de services de banque en ligne sont attirés vers de faux sites Web, afin de pouvoir intercepter leurs identifiants. C'est pour rendre le Web plus sûr que le HTTPS a été développé.Toute communication transmise au moyen de ce protocole de transport est ainsi chiffrée de manière que seul le vrai destinataire (navigateur Web ou serveur Web) puisse lire le contenu transmis. Les hackeurs profitent de la perception courante des consommateurs selon laquelle le HTTPS est une option sécurisée

pour s'engager sur Internet. Le protocole HTTPS est une version mise à jour du protocole HTTP, qui utilise le chiffrement et un certificat de sécurité validé par votre navigateur lors de la visite d'un site web. Différents services en ligne, dont les fournisseurs de courrier électronique, les banques et les e-commerçants, utilisent le protocole qui est désormais devenu synonyme de fiabilité et de légitimité. Cependant, certains développeurs Web peu recommandables peuvent aussi utiliser l'HTTPS pour créer des attaques, grâce à des éditeurs de certificats de sécurité gratuits, volés ou falsifiés. Nous voyons de nouvelles variantes d'attaques d'usurpation d'identité qui ciblent de nouveaux contacts et de nouveaux départements au sein des entreprises. Bien que cela soit vrai, HTTPS fait référence à un protocole d'application destiné au transfert sécurisé de données hypertextes et basé sur le protocole HTTP , mais il en est la version sécurisée. Ainsi, il s'agit d'un protocole qui offre la possibilité d' établir une connexion fiable entre le client et le serveur afin d' éviter toute interception par des personnes non autorisées. En ce sens, le protocole sécurisé pour le transfert hypertexte fonctionne comme un type de connexion qui sert à crypter les données des utilisateurs. Quant à son fonctionnement , en gros, lorsqu'un utilisateur confirme une entrée d'URL dans la barre d'adresse avec la touche Entrée ou, à défaut, clique sur un lien ; le navigateur établit une connexion automatiquement.[1]

1.4.1 Méthodes d'attaque

1.4.2 Attaques par déni de service (DoS) et par déni de service distribué (DDoS)

Une attaque par déni de service submerge les ressources d'un système afin que ce dernier ne puisse pas répondre aux demandes de service. Une attaque DDoS vise elle aussi les ressources d'un système, mais elle est lancée à partir d'un grand nombre d'autres machines hôtes infectées par un logiciel malveillant contrôlé par l'attaquant. À la différence des attaques conçues pour permettre à un attaquant d'obtenir ou de faciliter des accès, le déni de service ne procure pas d'avantage direct aux attaquants. Le déni de service est une satisfaction en soi pour certains pirates. Cependant, si la ressource attaquée appartient à un concurrent, l'avantage pour l'attaquant est alors bien réel. Une attaque DoS peut aussi avoir pour but de mettre un système hors ligne afin de pouvoir lancer un autre type d'attaque. Un exemple courant de cette technique est le détournement de session, que je décrirai plus loin. Un attaquant exploite l'utilisation de l'espace tampon lors du handshake d'initialisation de session TCP. La machine de l'attaquant inonde de demandes de connexion la petite file d'attente de traitement du système cible, mais elle ne réagit pas lorsque le système cible répond à ces demandes. Le système cible se met alors à temporiser en attendant la réponse de la machine de l'attaquant, ce qui fait planter le système ou le rend inutilisable lorsque la file d'attente de connexion se remplit.

1.4.3 Attaque teardrop

Cette attaque provoque le chevauchement des champs de longueur et de décalage de fragmentation des paquets séquentiels du protocole Internet (IP) au niveau de l'hôte attaqué ; au cours de ce processus, le système attaqué tente de reconstruire les paquets mais échoue. Le système cible s'embrouille et plante. En l'absence de correctifs pour se protéger contre cette attaque DoS, désactivez SMBv2 et bloquez les ports 139 et 445.

1.4.4 Attaque Smurf

Cette attaque implique d'usurper une adresse IP et d'utiliser l'ICMP pour saturer de trafic un réseau cible. Cette méthode d'attaque utilise des demandes d'écho ICMP ciblant des adresses IP de diffusion. Ces demandes ICMP proviennent d'une adresse usurpée. Si, par exemple, l'adresse de la victime choisie est 10.0.0.0.10, l'attaquant simule une demande d'écho ICMP de 10.0.0.0.10 à l'adresse de diffusion 10.255.255.255.255. Cette demande est envoyée à toutes les adresses IP de la plage, et toutes les réponses sont renvoyées à 10.0.0.0.10, submergeant ainsi le réseau. Ce processus est répétable et peut être automatisé en vue de générer des encombrements considérables sur le réseau. Ping of death Ce type d'attaque pingue un système cible avec des paquets IP dont la taille est supérieure au maximum de 65 535 octets. Les paquets IP de cette taille ne sont pas autorisés, le pirate les fragmente donc. Lorsque le système cible réassemble les paquets, il peut subir des débordements de tampon et d'autres plantages. Les attaques ping de la mort peuvent être bloquées à l'aide d'un pare-feu qui vérifie la taille maximale des paquets IP fragmentés. Botnets Les botnets sont des réseaux constitués de millions de systèmes infectés par des logiciels malveillants et contrôlés par des pirates informatiques afin d'effectuer des attaques DDoS. Ces bots ou systèmes zombies sont utilisés pour effectuer des attaques contre les systèmes cibles, souvent en submergeant leur bande passante et leurs capacités de traitement. Lorsqu'une attaque DDoS est détectée, l'hôte BGP (Border Gateway Protocol) doit envoyer des mises à jour de routage aux routeurs des FAI, afin qu'ils acheminent tout le trafic à destination des serveurs victimes vers une interface null0 lors du saut suivant. Le détournement de session, dans ce type d'attaque MitM, un attaquant détourne une session entre un client de confiance et un serveur réseau. L'ordinateur attaquant substitue son adresse IP au client de confiance pendant que le serveur poursuit la session, croyant qu'il communique avec le client. [1]

1.4.5 Attaque par mot de passe

Les mots de passe étant le mécanisme le plus couramment utilisé pour authentifier les utilisateurs d'un système informatique, l'obtention de mots de passe est une approche d'attaque courante et efficace. Le mot de passe d'une personne peut être obtenu en fouillant le bureau physique de la personne, en surveillant la connexion au réseau pour acquérir des mots

de passe non chiffrés, en ayant recours à l'ingénierie sociale, en accédant à une base de données de mots de passe ou simplement en devinant. Cette dernière approche peut s'effectuer de manière aléatoire ou systématique : Les attaques par force brute consistent à adopter une approche aléatoire : essayer différents mots de passe en espérant que l'un d'entre eux fonctionnera. Une certaine logique peut être appliquée : essayer des mots de passe liés au nom de la personne, à son poste, à ses passe-temps ou à des éléments similaires. Dans une attaque par dictionnaire, un dictionnaire des mots de passe courants est utilisé pour tenter d'accéder à l'ordinateur et au réseau d'un utilisateur. Une approche consiste à copier un fichier chiffré contenant les mots de passe, à appliquer le même chiffrement à un dictionnaire des mots de passe couramment utilisés et à comparer les résultats. Pour vous protéger contre les attaques par dictionnaire ou par force brute, vous devez mettre en œuvre une politique de verrouillage des comptes qui verrouillera un compte après quelques tentatives infructueuses de saisie du mot de passe. Ces bonnes pratiques de verrouillage des comptes vous permettront de faire cela correctement. Pour organiser une bonne défense, il faut connaître les attaques.[1]

1.4.6 Tolérance des navigateurs Web

Au cours du deuxième trimestre 2021, CyberRatings.org a effectué, de façon indépendante, un test de la protection contre les attaques d'hameçonnage proposée par les navigateurs web. 80 tests discrets ont été effectués sur une période de 20 jours. Pour assurer une protection contre les attaques d'hameçonnage, Microsoft Edge utilise Microsoft Defender SmartScreen, alors que Google Chrome et Mozilla Firefox utilisent l'API Google Safe Browsing. Microsoft Edge a bloqué 92,3 % des URL d'hameçonnage, soit le taux de protection le plus élevé, et s'est également retrouvé en première position à l'heure H0, avec un taux de 93,5 %. Google Chrome a fourni la deuxième protection la plus élevée, bloquant en moyenne 84,6 % des programmes malveillants, suivi par Mozilla Firefox à 83,2 %. Les systèmes de réputation d'URL réduisent le temps dont disposent les cybercriminels pour atteindre leurs objectifs, en empêchant l'accès à une URL d'hameçonnage connue et en avertissant les utilisateurs de cette menace. Cependant, dans la mesure où les utilisateurs visitent un large éventail de sites web, dont beaucoup sont récents, les systèmes de réputation d'URL ne peuvent pas bloquer toutes les nouvelles URL. Conscients de cela, les cybercriminels renouvellent constamment leurs campagnes d'hameçonnage, la majorité des attaques se produisant durant les premières heures suivant leur lancement. Tout au long du test, de nouvelles URL d'hameçonnage ont été ajoutées quotidiennement, et les URL qui n'étaient plus accessibles ou qui n'étaient plus à l'origine d'attaques d'hameçonnage ont été supprimées. Chaque point de données représente le taux de protection à un moment précis. Si une URL est bloquée dès le début, le score de régularité de la protection du navigateur s'améliore au fil du temps. De la même façon, si le navigateur ne bloque pas l'URL, son score diminue. Nous avons évalué la capacité de chaque navigateur à bloquer les URL malveillantes dès leur repérage sur Internet. Ces tests

ont été effectués toutes les six heures afin de déterminer le temps nécessaire à un navigateur pour ajouter une protection. L'illustration ci-dessus indique le temps nécessaire à chaque navigateur pour bloquer un site d'hameçonnage une fois la menace intégrée au cycle de test. L'hameçonnage est un type d'attaque par ingénierie sociale qui consiste à inciter la victime à fournir des informations personnelles sensibles au cybercriminel. Des informations sensibles sont, par exemple, les numéros de carte de crédit, les numéros de sécurité sociale et les identifiants et mots de passe liés à des comptes bancaires. Les e-mails, les messages instantanés, les SMS et les liens sur les réseaux sociaux sont autant de moyens utilisés pour réaliser des attaques d'hameçonnage. La page d'arrivée d'un site d'hameçonnage cherche souvent à accéder furtivement à l'ordinateur d'un visiteur et à y installer un programme malveillant (téléchargement furtif). Permettant de compromettre ou de récupérer des informations sensibles personnelles ou liées à une entreprise, ces attaques d'hameçonnage présentent un risque important pour les individus et les entreprises. Le groupe de travail antihameçonnage (Anti-PhishingWorking Group, APWG) a signalé un total de 396 688 campagnes d'hameçonnage par e-mail distinctes au T4 2020. La protection contre l'hameçonnage est assurée par une application au sein d'un navigateur web, qui inter- roge un service cloud concernant la réputation d'une URL, celui-ci scannant Internet à la recherche de sites d'hameçonnage et les ajoutant ensuite à une liste d'URL bloquées. Ainsi, lorsqu'un navigateur web tente d'accéder à une URL, la protection contre l'hameçonnage du navigateur (soit Safe Browsing, SmartScreen, etc.) redirige l'utilisateur vers un message d'avertissement expliquant que l'URL est malveillante. Certains systèmes de réputation incluent également des contenus pédagogiques. À l'inverse, si un site web est jugé « fiable », le navigateur web n'effectue aucune action. Google et Mozilla utilisent l'API Google Safe Browsing pour vérifier la réputation des URL, mais également pour alerter les utilisateurs concernant le téléchargement de certains types de fichiers. Microsoft Defender SmartScreen, la technologie de protection intégrée à Microsoft Edge, fournit une protection contre les attaques reposant sur les URL, via un service cloud intégré de réputation d'URL et d'applications destiné au blocage de fichiers malveillants. [1]

1.4.7 Configuration du laboratoire

Pour démontrer que les serveurs d'application exploitables peuvent être utilisés dans des attaques pleinement opérationnelles sur HTTPS, nous avons créé un laboratoire contenant des preuves de concept pour chaque protocole et méthode d'attaque de notre évaluation. En particulier, le laboratoire contient Sendmail 8.16.1 (SMTP), Cyrus 2.4.17 (IMAP), Courier 1.0.6-1 (POP3), et vsftpd 3.0.2 (FTP). Le laboratoire est basé sur le logiciel de conteneurisation Docker, et contient une attaque par téléchargement pour Courier POP3, et une attaque par réflexion pour Sendmail en combinaison avec Edge Legacy et IE. Les attaques sont implémentées en Python. La configuration complète du laboratoire, y compris toutes les attaques de preuve de concept, est disponible sur Github. [1] [2]

Chapitre 2

Conception et matériels

Introduction

TLS, Transport Layer Security utilisé pour sécuriser les applications ; ces applications qui sont toutes fois vulnérables ; ici nous nous focalisons sur des applications codées pour fonctionner pas sans un réseau informatique. Notre objectif étant de proposer une solution sur comment atténuer les attaques ALPACA, nous allons dans ce chapitre présenter à partir d'un LAB l'attaque ALPACA ensuite nous allons à partir du CVE de ALPACA chercher les applications et protocoles qui sont liée à cette attaque et aussi comment peut on rendre ces applications et protocoles non vulnérables aux attaques ALPACA.

2.1 Méthodologie

2.1.1 Analyse critique des attaques interprotocoles

Les attaques interprotocoles sont des attaques qui utilisent des vulnérabilités du TLS pour nuire aux systèmes d'informations ; entre autre aux applications cibles attaquées. Parlant d'attaques interprotocoles nous avons : le Checksum Spoofing ; un adversaire falsifie un message de somme de contrôle dans le but de faire apparaître une charge utile comme ayant une somme de contrôle correspondante valide. Les sommes de contrôle sont utilisées pour vérifier l'intégrité des messages. Elles consistent en une valeur basée sur la valeur du message qu'elles protègent. Les codes de hachage sont un mécanisme courant de somme de contrôle. L'expéditeur et le destinataire sont tous deux capables de calculer la somme de contrôle sur la base du contenu du message. Si le contenu du message change entre l'expéditeur et le destinataire, l'expéditeur et le destinataire calculeront des valeurs de somme de contrôle différentes. Étant donné que la valeur de la somme de contrôle de l'expéditeur est transmise avec le message, le destinataire sait qu'une modification a eu lieu. Dans le cas

de l'usurpation de la somme de contrôle, un adversaire modifie le corps du message et modifie ensuite la somme de contrôle correspondante de sorte que le calcul de la somme de contrôle du destinataire corresponde à la somme de contrôle (créée par l'adversaire) dans le message. Cela empêche le destinataire de se rendre compte qu'une modification a eu lieu. spoofing of UDDI ;un attaquant falsifie un message UDDI, ebXML ou similaire afin d'usurper l'identité d'un fournisseur de services dans le cadre d'une transaction commerciale électronique. Les normes UDDI, ebXML et autres normes similaires sont utilisées pour identifier les entreprises dans les transactions de commerce électronique. Elles permettent notamment d'identifier un participant particulier, les informations WSDL pour les transactions SOAP et les protocoles de communication pris en charge, y compris les protocoles de sécurité. En usurpant l'un de ces messages, un pirate pourrait se faire passer pour une entreprise légitime dans une transaction ou manipuler les protocoles utilisés entre un client et une entreprise. Cela pourrait entraîner la divulgation d'informations sensibles, la perte d'intégrité des messages, voire une fraude financière. intent spoof ;un adversaire, par le biais d'une application malveillante préalablement installée, émet une intention dirigée vers un composant spécifique d'une application de confiance dans le but d'atteindre une variété d'objectifs différents, y compris la modification de données, la divulgation d'informations et l'injection de données. Les composants qui ont été exportés involontairement et rendus publics sont sujets à ce type d'attaque. Si le composant fait confiance à l'action intentionnelle sans vérification, l'application cible exécute la fonctionnalité à la demande de l'adversaire, l'aidant ainsi à obtenir l'impact technique négatif souhaité. Avec ces informations tirées et analysées du site https://capec.mitre.org/data/definitions/1000.html, les attaques interprocoles encores appelées ALPACA attacks sont des attaques sévères qu'il faudra chercher des solutions et mécanismes pour éviter à tout prix.

2.1.2 Présentation de l'attaque ALPACA

Afin de présenter ce que c'est que l'attaque ALPACA, nous avons utilisé un LAB dont la source se retrouve sur le lien. Le LAB présentant des vulnérabilité ALPACA, consiste à la mise en place d'un serveur local sur une machine virtuelle Ubuntu pour étudier les attaques interprotocoles sur TLS en général et mener une étude de cas systématique sur les serveurs web, rediriger les requêtes HTTPS du navigateur Web d'une victime vers Serveurs SMTP, IMAP, POP3 et FTP.

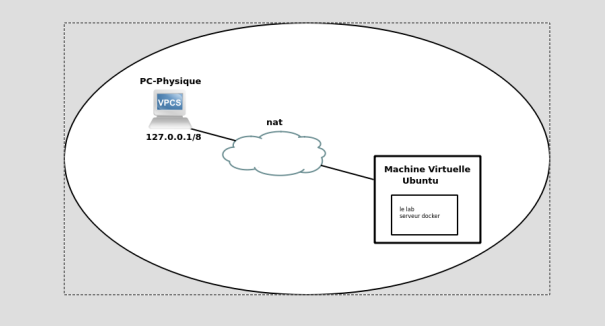


Figure 2.1 – Architecture réseau de l'infrastruture du Laboratoire

Comme décrit dans la documentation de notre LAB suivant le lien https://github.com/RUB-NDS/alpaca-code/tree/master/testlab notre configuration (l'infrastructure de notre configuration) repose sur une architecture client serveur où sur une machine physique est installé une machine virtuelle Ubuntu qui héberge le Lab en utilisant docker (serveur docker) sur lequel est installé les services et applications indépendemment du réseau (indépendemment de l'architecture de l'infrastructure). Pour entrer dans les détails du LAB, il est constitué de librairies et se repose sur TLS dont voici quelques codes sources pour expliquer. Le code suivant permet de compiler et de mettre en place sur la machine Ubuntu le serveur du Lab.

```
#!/bin/bash
for library in baseimage openssl bearssl botan java gnutls golang mbedtls wolfssl rustls;
do (cd "$library"./build.sh);
done
```

Dans ce code se trouve une commande ; la commande do (cd "$library"./build.sh) ; qui permet de faire le build après le parcours des librairies qui constituent le Lab afin de les compiler ; les compilations qui se font au niveau du docker et le Lab enfin disponible avec les services et applications. Ci-dessous en code les différents services (serveurs) et liens entre ces services dans le lab.

```
version : "3"
services :
nginx-proxy :
image : jwilder/nginx-proxy
container_name : alpaca-nginx-proxy
ports :
    - "127.0.0.1 :80 :80"
    - "127.0.0.1 :443 :443"
volumes :
    - /var/run/docker.sock :/tmp/docker.sock :ro
    - ./files/cert/ :/etc/nginx/certs/ :ro
nginx-target :
image : php :7.3-apache
container_name : alpaca-nginx-target
environment :
    - VIRTUAL_HOST=target.com
    - VIRTUAL_PORT=80
volumes :
    - ./files/nginx-target/html/ :/var/www/html/ :ro
```

```
20 nginx-attacker:
21 image: nginx
22 container_name: alpaca-nginx-attacker
23 depends_on:
24     - nginx-proxy
25 environment:
26     - VIRTUAL_HOST=attacker.com
27 volumes:
28     - ./files/nginx-attacker/html/:/usr/share/nginx/html/:ro
29 vsftp:
30 image: alpaca-vsftp
31 build:
32 context: files/ftp-vsftp/
33 container_name: alpaca-vsftp
34 tty: true
35 volumes:
36     - ./files/cert/:/opt/certs/:ro
37     - ./files/ftp-vsftp/payload.html:/home/vsftpd/bob/payload.html
38 ports:
39     - 0.0.0.0:20:20
40     - 0.0.0.0:21:21
41     - 0.0.0.0:10090-10100:10090-10100
42 environment:
43     - FTP_USER=bob
44     - FTP_PASS=12345
45     - PASV_MIN_PORT=10090
46     - PASV_MAX_PORT=10100
47     - REVERSE_LOOKUP_ENABLE=NO
48 cyrus:
49 image: alpaca-cyrus-imap
50 build:
51 context: files/mail-cyrus/
52 container_name: alpaca-cyrus
53 tty: true
54 ports:
55     - "0.0.0.0:143:143"
56     - "0.0.0.0:993:993"
57 volumes:
58     - ./files/mail-cyrus/imapd.conf:/etc/imapd.conf
59     - ./files/mail-cyrus/cyrus.conf:/etc/cyrus.conf
60     - ./files/cert/:/tmp/ssl/
61 proftpd:
62 image: alpaca-proftpd
63 build:
64 context: ./files/ftp-proftpd/
65 container_name: alpaca-proftpd
66 restart: unless-stopped
67 tty: true
68 environment:
69 FTP_LIST: "bob:12345"
70 USERADD_OPTIONS: "-o --gid 33 --uid 33"
71 volumes:
72     - "./files/ftp-proftpd/bob:/home/bob"
73     - "./files/cert/:/opt/ssl/"
74 ports:
75     - 0.0.0.0:20:20
76     - 0.0.0.0:21:21
77     - 0.0.0.0:10101-10200:10101-10200
78 courier:
```

```
79 image: alpaca-courier
80 build:
81 context: ./files/mail-courier/
82 container_name: alpaca-courier
83 restart: unless-stopped
84 ports:
85     - "0.0.0.0:110:110"
86     - "0.0.0.0:143:143"
87 volumes:
88     - "./files/cert/:/opt/ssl/"
89 tty: true
90 sendmail:
91 image: alpaca-sendmail
92 build:
93 context: files/mail-sendmail/
94 container_name: alpaca-sendmail
95 tty: true
96 ports:
97     - "0.0.0.0:25:25"
98 volumes:
99     - ./files/cert/:/tmp/certs/
```

Une fois les librairies que constitue le Lab sont compilés et le serveur mis en place, un certificat ssl est généré et mis en place pour être utilisé par Mozila Firefox pour les tests. En rentrant dans les détails du Lab, nous avons différents services qui sont contenarizés dans docker avec les protocoles suivants vsftp, cyrus, cyrus-proftpd, courier, sendmail. Ces services nous permettent d'avoir un ensemble d'images docker qui interragissent. Dans le code source ci-dessus, nous avons différents contextes pour les build qui sont **files/ftp-vsftp**, **cyrus**, **files/mail-cyrus**, **proftpd**, **files/ftp-proftpd**, **courier**, **files/mail-courier**, **sendmail**, **files/mail-sendmail** avec les ports pour les différents services définit, les volumes pour servir de storage ainsi que les environnements précisant les domain name, usernames pour services avec bien d'autres détails de configurations. Lorsque le serveur démarre, nous avons les différents services implémentés opérationnels et le Lab est prêt pour faire les tests.

En général les tests concernent TLS ; vulnérabilités du TLS sur les protocoles suivants vsftp, cyrus, cyrus-proftpd, courier, sendmail. C'est quoi ces protocoles ? Comment TLS intervient dans ces protocoles et l'attaque ALPACA sur ces protocoles ? Nous avons cité les protocoles vsftp, cyrus, cyrus-proftpd, courier, sendmail comme cible pour voir comment TLS interviennent et comment ils en sont vulnérables. Nous résumons ces protocoles en trois grandes catégories à savoir le FTPS, le POPS, l'IMAPS. Le setup du lab est de tel sorte qu'un attaquant man in the middle puisse être en mesure en utilisant alpaca attaque d'avoir accès à des informations protégées ; voici le lien github pour aller sur le code https://github.com/RUB-NDS/alpaca-code/blob/master/testlab/mitmproxy/main.py. Afin de procéder aux attaques dans le lab, comme décrit sur l'adresse Github https://github.com/RUB-NDS/alpaca-code/tree/master/testlab#ftps il nous faut aller en ligne de commande sur notre machine virtuelle Ubuntu et entrer les commandes docker-compose -f servers/docker-compose.yml up -d nginx-target nginx-attacker vsftp et cd mitmproxy ; python3 main.py –proto FTP –attacker_ip 127.0.0.2 127.0.0.1 21 en respectant la structure des

dossiers. Nous avons les résultats des tests ci-dessous en figures.

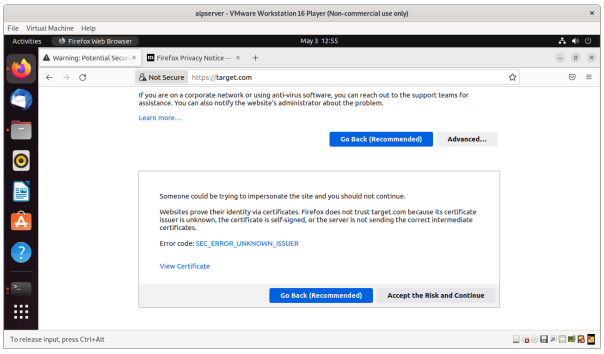


Figure 2.2 – Page d'accueil target.com affichant risque de sécurité

Pour rappel, l'objectif de ce laboratoire préconçu est d'évaluer des librairies TLS déjà à la base vulnérable à ALPACA attack dans le but de voir, de regarder et de comprendre les attaques interprotocoles sur TLS. Pour une première comme décrit un peu plus haut, au cours de la compilation des librairies pour la mise en place du Lab, un certificat ssl est généré et placé à la racine du répertoire **/home** afin que le navigateur firefox puisse l'utiliser ; un certificat ssl auto signé ; ce qui fait que firefox affiche une alerte afin de prévenir les risques de sécurité.

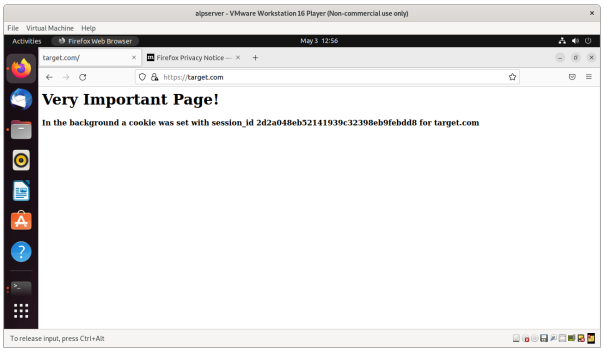


Figure 2.3 – Target.com cookie de sécurité affiché

Ici, comme nous avons accepté d'évoluer avec les risques, un cookie est affiché ; page accueil de target.com

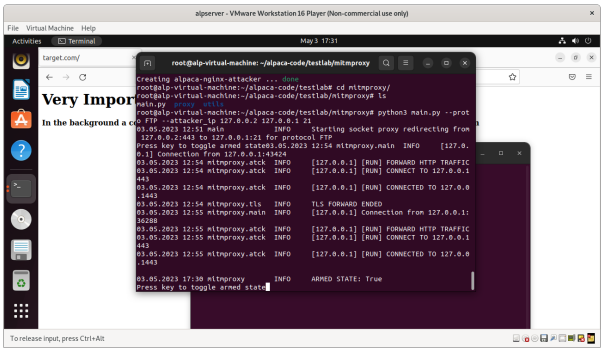


Figure 2.4 – Activation de mode de sécurité sur le proxy pour les attaques

Pour continuer, il nous faut activer le mode de sécurité sur le proxy afin de pouvoir procéder aux attaques

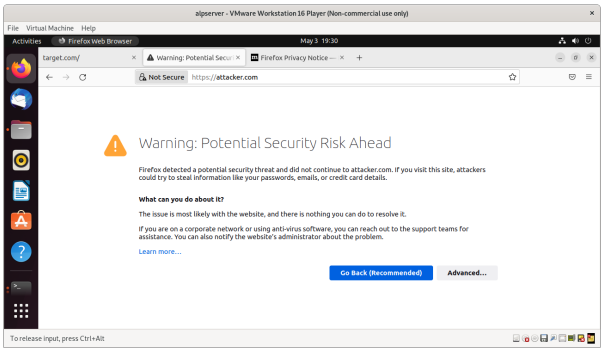


Figure 2.5 – Alerte de sécurité sur attacker.com

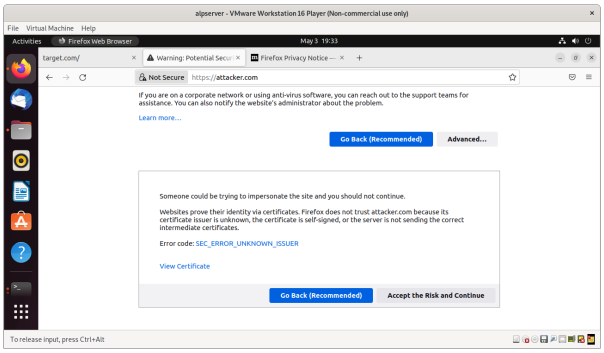


Figure 2.6 – Page attacker.com acceptation de risque

Pour évoluer et procéder aux attaques, sur attacker.com, attacker.com étant l'hôte qui affiche toutes les possibilités d'attaques interprotocoles sur TLS, en https il faudra accepter les risques comme certificat ssl auto signé et aussi mode de sécurité sur le proxy pour les attaques activées.

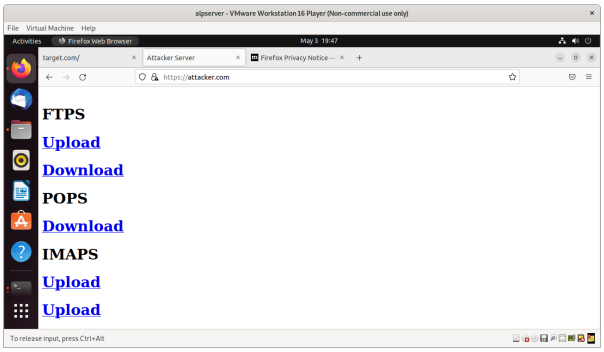


Figure 2.7 – Page attacker.com avec différents types d'attaques interprotocoles

A l'accueil de attacker.com, nous avons les différentes attaques interprotocoles sur TLS ; ALPACA attack tant pour le FTPS, POPS, IMAPS et aussi pour le HTTPS.

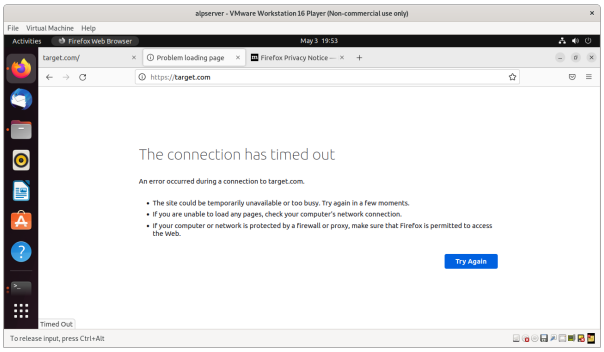


Figure 2.8 – Page web target.com innacessible ; attaque TLS

En cliquant sur le lien upload de FTPS, nous observons une redirection sur target.com mais cette fois inaccessible (time out).

En référence au site https://www.cvedetails.com/cve/CVE-2021-3618/, ALPACA est une attaque par confusion de contenu de protocole de couche d'application, qui exploite des serveurs TLS mettant en œuvre des protocoles différents mais utilisant des certificats compatibles, tels que des certificats multi-domaines ou wildcard. Un attaquant MiTM ayant accès au trafic de la victime au niveau de la couche TCP/IP peut rediriger le trafic d'un sous-domaine vers un autre, ce qui donne lieu à une session TLS valide. Cela brise l'authentification de TLS et des attaques inter-protocoles sont possibles, le comportement d'un service protocolaire pouvant compromettre l'autre au niveau de la couche application. C'est le cas dans le lab que nous avons étudié. Lorsqu'on visite un site web protégé par HTTPS, le navigateur n'échange pas de données avec le serveur web tant qu'il ne s'est pas assuré de la validité du certificat numérique du site. Cela empêche les pirates ayant la capacité de surveiller ou de modifier les données passant entre soi et le site d'obtenir des cookies d'authentification ou d'exécuter un code malveillant sur l'appareil qui visite. Mais que se passerait-il si un pirate informatique pouvait confondre le navigateur et le pousser à se connecter accidentellement à un serveur de courrier électronique ou à un serveur FTP utilisant un certificat compatible avec celui utilisé par le site web ? Étant donné que le nom de domaine du site web correspond au nom de domaine figurant dans le certificat du serveur de courrier électronique ou du serveur FTP, le navigateur établira, dans de nombreux cas, une connexion Transport Layer Security avec l'un de ces serveurs plutôt qu'avec le site web que l'utilisateur avait l'intention de visiter. Étant donné que le navigateur communique en HTTPS et que le serveur de courrier électronique ou FTP utilise SMTP, FTPS ou un autre protocole, il est possible que les choses tournent mal ; un cookie d'authentification décrypté pourrait être envoyé à l'attaquant, par exemple, ou un attaquant pourrait exécuter un code malveillant sur la machine visitée. Ces attaques sont possibles parce que le protocole TLS ne protège pas l'intégrité de la connexion TCP elle-même plutôt que l'intégrité du serveur qui parle HTTP, SMTP ou un autre langage Internet. Les attaquants de type "homme du milieu" peuvent exploiter cette faiblesse pour rediriger le trafic TLS du serveur et du protocole prévus vers un autre point de terminaison et un

autre protocole de substitution. Le principe de base est qu'un attaquant peut rediriger le trafic destiné à un service vers un autre, car TLS ne protège pas l'adresse IP ou le numéro de port. En référence au site https://alpaca-attack.com/index.html la faille générale derrière ALPACA se trouve dans l'authentification du serveur TLS, donc potentiellement tous les serveurs TLS qui ont des certificats compatibles avec d'autres services TLS sont affectés. Par conséquent, tous ces serveurs doivent être considérés comme vulnérables. Cependant, pour des raisons pratiques, cette définition n'est pas très utile, car la faille n'est exploitable que dans certains cas. Il faut donc faire la distinction entre les serveurs vulnérables et les services exploitables. Le partage de certificats entre un serveur web et un serveur FTP est presque toujours dangereux si un attaquant a un accès en écriture au serveur FTP. Il est parfois dangereux si l'attaquant n'a pas d'accès en écriture. Le partage de certificats entre un serveur web et un serveur SMTP/POP3/IMAP est parfois dangereux, en fonction du comportement exact du serveur. Voici une liste des implémentations analysées en ce qui concerne leur vulnérabilité : Sendmail SMTP a permis des attaques par réflexion qui fonctionnent dans Internet Explorer lorsqu'il est utilisé sur STARTTLS. Les serveurs IMAP de Cyrus, Kerio Connect et Zimbra permettent des attaques par téléchargement et par réflexion qui fonctionnent dans Internet Explorer. Courier, Cyrus, Kerio Connect et Zimbra ont permis des attaques par téléchargement qui fonctionnent dans Internet Explorer. Les serveurs Microsoft IIS, vsftpd, FileZilla Server et Serv-U FTP ont permis des attaques par réflexion qui fonctionnent dans Internet Explorer. Et les mêmes serveurs FTP ont permis des attaques par téléchargement en amont et en aval qui fonctionnent dans tous les navigateurs. Les navigateurs sont généralement affectés par la vulnérabilité, mais ils ne sont pas responsables de la faille. Certains navigateurs sont plus vulnérables que d'autres en raison de la manière dont ils réagissent aux réponses non HTTP. La plupart des attaques nécessitent un attaquant actif de type Man-in-the-Middle, c'est-à-dire un moyen pour un attaquant d'intercepter et de modifier les données envoyées par le navigateur de la victime au serveur web. Cela est difficile sur l'internet, mais peut constituer un modèle d'attaque plausible sur le réseau local. Par ailleurs, certaines variantes d'attaques ne nécessitent pas de Man-in-the-Browser et sont donc plus dangereuses.

2.1.3 Atténuation de l'attaque ALPACA

Notre but dans cette sous section est de trouver une solution étape par étape pour atténuer l'attaque alpaca en utilisant des sites vulnérables à l'attaque alpaca. Afin de voir les sites qui sont vulnérables à alpaca attaque nous allons ici nous baser sur le CVE de alpaca code en utilisant le lien suivant https://www.cvedetails.com/cve/CVE-2021-3618/. Le site https://www.cvedetails.com est une base de données gratuite en ligne de CVE sur laquelle on peut rechercher des fournisseurs, des produits et des versions et consulter les entrées CVE, les vulnérabilités, qui s'y rapportent. On peut aussi consulter des statistiques sur les fournisseurs, les produits et les versions des produits. ALPACA est une attaque de confusion

de contenu niveau couche application. Nous avons trouvé sur le site https://www.cvedetails.com que le CVE de ALPACA est CVE-2021-3618. Afin de trouver le CVE lié à alpaca nous sommes allés sur le site https://www.cvedetails.com et rechercher dans la case de recherche appropriée le mot alpaca ; ce qui nous a permis d'avoir le CVE de ALPACA. A partir du CVE de alpaca nous pouvons avoir les applications et protocoles qui s'y rapportent en suivant les étapes suivantes. Dans les détails du CVE de alpaca, dans la section CVSS Scores Vulnerability Types, nous avons le CWE ID qui est le numero 295 dont nous obtenons les détails à partir du lien https://cwe.mitre.org/data/definitions/295.html. Le site https://cwe.mitre.org/ est developpé par une communauté qui regroupe toute une liste de différents types de faiblesses de logiciels et de matériels ; CWE, Common Weakness Enumeration. Il sert de langage commun, d'instrument de mesure pour les outils de sécurité et de référence pour l'identification des faiblesses, l'atténuation des effets et les efforts de prévention. En se référent au CWE ID du CVE (CVE-2021-3618) de alpaca sur le site https://cwe.mitre.org/data/definitions/295.html qui signifie Improper Certificate Validation, toute application ayant pour vulnérabilité alpaca, leur vulnérabilité se concentre sur le fait qu'il ne valide pas ou valides de façon incorrecte un certificat. Lorsqu'un certificat est invalide ou malveillant, il peut permettre à un attaquant d'usurper une entité de confiance en interférant dans le chemin de communication entre l'hôte et le client. L'application peut se connecter à un hôte malveillant tout en croyant qu'il s'agit d'un hôte de confiance, ou l'application peut être trompé et accepter des données usurpées qui semblent provenir d'un hôte de confiance. La vulnérabilité alpaca ne se limite pas juste aux failles décrits. Toujours en se référent au site https://cwe.mitre.org/data/definitions/295.html qui décrit en profondeur l'attaque alpaca, cette vulnérabilité se rapporte à d'autres vulnérabilités classées par nature et types. Nous avons la vulnérabilité Improper authentication de nature ChildOf de type C, ayant pour ID 287. Il est important de lister ces vulnérabilités qui se rapportent à alpaca afin de mieux analyser une application vulnérable à alpaca. Ci-dessous nous avons la liste de ces vulnérabilités : Improper Following of a Certificate's Chain of Trust, de nature ParentOf de type B ayant pour ID 296, Improper Validation of Certificate with Host Mismatch de nature ParentOf de type V ayant pour ID 297, Improper Validation of Certificate Expiration de nature ParentOf de type V ayant pour ID 298, Improper Check for Certificate Revocation de nature ParentOf de type B ayant pour ID 299, Missing Validation of OpenSSL Certificate de nature ParentOf de type V ayant pour ID 599, Key Exchange without Entity Authentication de nature PeerOf de type B ayant pour ID 322, Authentication Errors de nature MemberOf de type C ayant pour ID 1211, Identify Actors de nature MemberOf type C et ayant pour ID 1014. L'attaque ALPACA se focalisant sur des certificats, un certificat est un jeton qui associe une identité (principal) à une clé cryptographique. Les certificats peuvent être utilisés pour vérifier si une clé publique appartient au propriétaire présumé. Parlant de ces vulnérabilités classées par type et leur nature, la vulnérabilité de type C de nature en englais ChildOf est Class ; une faiblesse décrite de manière très abstraite, généralement in-

dépendamment de tout langage ou technologie spécifique. Plus spécifique qu'une faiblesse pilier, mais plus générale qu'une faiblesse de base. Les faiblesses relatives à Class décrivent généralement les problèmes suivant les dimensions ci-après : comportement, propriété et ressource. Les vulnérabilités qui sont de type B est un point faible qui est encore largement indépendant d'une ressource ou d'une technologie, mais qui est suffisamment détaillé pour fournir des méthodes spécifiques de détection et de prévention. Les faiblesses liées à Base décrivent généralement les problèmes en termes des dimensions suivantes : comportement, propriété, technologie, langue et ressource. La vulnérabilité de nature ParentOf de type V est une faiblesse liée à un certain type de produit, impliquant généralement une langue ou une technologie spécifique. Plus spécifique qu'une faiblesse de Base. Les faiblesses de niveau variante décrivent généralement des problèmes relevant de 3 à 5 des dimensions suivantes : comportement, propriété, technologie, langue et ressource. La vulnérabilité de type C Categorie car de nature MemberOf est une entrée de CWE qui contient un ensemble d'autres entrées partageant une caractéristique commune. La vulnérabilité alpaca peut être causé durant l'implementation d'une architecture de sécurité. Lorsque le produit utilise l'épinglage de certificats, le développeur peut ne pas valider correctement tous les composants pertinents du certificat avant d'épingler ce dernier. Il peut alors être difficile ou coûteux de procéder à des tests une fois l'épinglage terminé. Cette vulnérabilité étant de type C, elle n'est spécifique à aucun langage ni plateforme. Un attaquant peut facilement avoir accès à tout mécanisme de protection, avoir privilège ou même usurper une identité. En continuant avec CWE ID 295, dans les détails spécifiques, une application ayant cette vulnérabilité ne valide pas ou valide de façon incorrecte un certificat; quand le certificat est invalide ou malicieux, cette faiblesse peut permettre à un attaquant d'usurper une entité de confiance en interférant dans le chemin de communication entre l'hôte et le client. Nous allons voir en référence du site https://cwe.mitre.org/data/definitions/295.html quelques exemples démonstratives sur certaines implémentations (code à l'appuie) de comment voir la vulnérabilité alpaca dans une application. Le bout de code suivant vérifie le certificat d'une pair connectée.

```
1 if ((cert = SSL_get_peer_certificate(ssl)) && host)
2 foo=SSL_get_verify_result(ssl);
3 if ((X509_V_OK==foo) || X509_V_ERR_SELF_SIGNED_CERT_IN_CHAIN==foo))
4 // certificate looks good, host can be trusted
```

Dans ce cas, comme le certificat est auto-signé, aucune autorité externe ne peut prouver l'identité de l'hôte. Le programme pourrait communiquer avec un autre système qui usurpe l'identité de l'hôte, par exemple en empoisonnant le cache DNS ou en utilisant une attaque AITM (Adversary-in-the-Middle) pour modifier le trafic entre le serveur et le client. Nous avons plusieurs autres exemples démonstratives que nous allons utiliser par la suite que voici ci-dessous.

```
1     cert = SSL_get_peer_certificate(ssl);
2     if (cert && (SSL_get_verify_result(ssl)==X509_V_OK)) {
3     // do secret things
4     }
```

Cet exemple de code OpenSSL obtient un certificat et le vérifie. Même si l'étape de vérification retourne X509_V_OK cette étape n'inclut pas la vérification du nom commun par rapport au nom de l'hôte. En d'autres termes, il n'y a aucune garantie que le certificat correspond à l'hôte souhaité. La connexion SSL a pu être établie avec un hôte malveillant qui a fourni un certificat valide.

```
    if (cert = SSL_get_peer(certificate(ssl)) {
    foo=SSL_get_verify_result(ssl);
    if ((X509_V_OK==foo) || (X509_V_ERR_CERT_HAS_EXPIRED==foo))
    //do stuff
}
```

Ce code OpenSSL garantit l'existence d'un certificat et permet l'utilisation de certificats expirés. Si l'appel à SSL_get_verify_result() renvoie X509_V_ERR_CERT_HAS_EXPIRED, cela signifie que le certificat a expiré. Au fur et à mesure que le temps passe, les chances de compromettre le certificat augmentent pour les attaquants.

```
if (cert = SSL_get_peer_certificate(ssl)) {
    // got a certificate, do secret things
}
```

Le code OpenSSL ci-dessus s'assure de l'existence d'un certificat avant de poursuivre l'exécution. Comme ce code n'utilise pas SSL_get_verify_results() pour vérifier le certificat, il pourrait accepter des certificats qui ont été révoqués (X509_V_ERR_CERT_REVOKED). Le logiciel pourrait communiquer avec un hôte malveillant.

```
if (cert = SSL_get_peer_certificate(ssl)) {
    // got certificate, host can be trusted
    //foo=SSL_get_verify_result(ssl);
    //if (X509_V_OK==foo) ...
}
```

Le code OpenSSL ci-dessus permet de s'assurer que l'hôte dispose d'un certificat. Notons que le code n'appelle pas SSL_get_verify_result(ssl), ce qui désactive effectivement l'étape de validation qui vérifie le certificat. Comme décrit plus haut l'attaque alpaca de cve CVE-2021-3618 ayant pour CWE ID 295 à des CWE qui s'y rapportent. Nous allons rentrer dans les détails de ces CWE avec des exemples de code à l'appui. Le CWE-287 qui est Improper Authentication, une vulnérabilité de ce type est retrouvé lorsqu'un acteur affirme avoir une identité donnée, l'application ne prouve pas ou insuffisamment que la demande de clé est correcte. Cette vulnérabilité a d'autres vulnérabilités qui s'y rapportent mais nous n'allons pas rentrer dans ces détails. Cette faiblesse est due à la mise en œuvre d'une tactique de sécurité architecturale. Elle n'est spécifique à un langage de programmation. Cette vulnérabilité peut permettre à l'attaquant de lire les données d'une application, obtenir des privilèges ou usurper une identité, Exécuter un code ou des commandes non autorisées.

Cette faiblesse peut conduire à l'exposition de ressources ou de fonctionnalités à des acteurs involontaires, ce qui peut permettre à des attaquants d'obtenir des informations sensibles ou même d'exécuter un code arbitraire. Le code suivant vise à s'assurer que l'utilisateur est

déjà connecté. Si ce n'est pas le cas, le code effectue une authentification avec le nom d'utilisateur et le mot de passe fournis par l'utilisateur. En cas de succès, il définit les cookies ”loggedin” et ”user” pour ”se souvenir” que l'utilisateur s'est déjà connecté. Enfin, le code exécute les tâches d'administrateur si l'utilisateur connecté possède le nom d'utilisateur ”Administrateur”, tel qu'il est enregistré dans le cookie utilisateur.

```
my $q = new CGI;
if ($q->cookie('loggedin) ne "true") {
    if (! AuthenticateUser($q->param('username), $q->param('password))) {
    ExitError("Error: you need to log in first");
}
else {
# Set loggedin and user cookies.
$q->cookie(
-name => "loggedin,
-value => "true
);
$q->cookie(
-name => "user,
-value => $q->param('username)
);
}
}
if ($q->cookie('user) eq "Administrator") {
DoAdministratorTasks();
}
```

Malheureusement, ce code peut être contourné. L'attaquant peut définir les cookies de manière indépendante afin que le code ne vérifie pas le nom d'utilisateur et le mot de passe. L'attaquant peut faire cela avec une requête HTTP contenant des en-têtes tels que :

```
GET /cgi-bin/vulnerable.cgi HTTP/1.1
Cookie: user=Administrator
Cookie: loggedin=true
[body of request]
```

En attribuant la valeur true au cookie loggedin, le pirate contourne l'ensemble du contrôle d'authentification. En utilisant la valeur administrator dans le cookie utilisateur, le pirate obtient également des privilèges pour administrer le logiciel. Le CWE-296 qui est Improper Following of a Certificate's Chain of Trust, quand une application en est vulnérable, cette dernière ne suit pas, ou suit de manière incorrecte, la chaîne de confiance d'un certificat jusqu'à un certificat racine de confiance, ce qui entraîne une confiance incorrecte de toute ressource associée à ce certificat. Si un système ne suit pas la chaîne de confiance d'un certificat jusqu'à un serveur racine, le certificat perd toute utilité en tant que mesure de confiance. Essentiellement, la confiance accordée par un certificat est dérivée d'une chaîne de confiance avec une entité de confiance réputée. L'utilisateur final doit faire confiance à cette source réputée, et cette source réputée doit se porter garante de la ressource en question par l'intermédiaire du certificat. Dans certains cas, cette confiance passe par plusieurs entités qui se portent garantes les unes des autres. L'entité à laquelle l'utilisateur final fait confiance se trouve à l'une des extrémités de cette chaîne de confiance, tandis que la ressource détentrice

du certificat se trouve à l'autre extrémité de la chaîne. Si l'utilisateur reçoit un certificat à l'extrémité de l'une de ces chaînes de confiance et qu'il ne vérifie ensuite que le premier maillon de la chaîne, aucune confiance réelle n'a été établie, puisqu'il faut remonter toute la chaîne jusqu'à une source de confiance pour vérifier le certificat. La chaîne de confiance peut être rompue de plusieurs façons, notamment par les moyens suivants :

— Tout certificat de la chaîne est auto-signé, sauf s'il s'agit du certificat racine.
— Tous les certificats intermédiaires ne sont pas vérifiés, en commençant par le certificat d'origine jusqu'au certificat racine.
— Un certificat intermédiaire signé par une autorité de certification ne comporte pas les contraintes de base attendues ou d'autres extensions importantes.
— Le certificat racine a été compromis ou autorisé à la mauvaise partie.

Cette faiblesse est due à la mise en œuvre d'une tactique de sécurité architecturale. Lorsque le produit utilise l'épinglage de certificats, le développeur peut ne pas valider correctement tous les composants pertinents du certificat avant d'épingler ce dernier. Il peut alors être difficile ou coûteux de procéder à des tests une fois l'épinglage terminé. Cette faiblesse n'est spécifique à un langage de programmation. L'exploitation du CWE ID 296 peut conduire à la confiance dans des données qui peuvent provenir d'une source usurpée ; l'attaquant peut avoir des privilèges ou usurper l'identité et même exécuter des codes ou des commandes. Les données, les demandes ou les actions entreprises par l'entité attaquante peuvent être exécutées en tant qu'entité bénigne usurpée. Avec ces détails, nous avons la non-répudiation des données, l'intégrité, la confidentialité, la disponibilité et le contrôle d'accès qui sont impactées. Nous avons ci-dessous un exemple de code à titre d'exemple démonstratif. Ce code vérifie le certificat d'une pair connectée.

```
1    if ((cert = SSL_get_peer_certificate(ssl)) && host)
2    foo=SSL_get_verify_result(ssl);
3    if ((X509_V_OK==foo) || X509_V_ERR_SELF_SIGNED_CERT_IN_CHAIN==foo))
4    // certificate looks good, host can be trusted
```

Dans ce cas, comme le certificat est auto-signé, aucune autorité externe ne peut prouver l'identité de l'hôte. Le programme pourrait communiquer avec un autre système qui usurpe l'identité de l'hôte, par exemple en empoisonnant le cache DNS ou en utilisant une attaque AITM (Adversary-in-the-Middle) pour modifier le trafic entre le serveur et le client. Le CWE-297 qui est Improper Validation of Certificate with Host Mismatch quand l'application y est affectée, communique avec un hôte qui fournit un certificat, mais ne s'assure pas correctement que le certificat est effectivement associé à cet hôte. Même si un certificat est bien formé, signé et suit la chaîne de confiance, il peut simplement s'agir d'un certificat valide pour un site différent de celui avec lequel le produit interagit. Si les données spécifiques à l'hôte du certificat ne sont pas correctement vérifiées - telles que le nom commun (CN) dans le sujet ou l'extension Subject Alternative Name (SAN) d'un certificat X.509 - il est possible qu'une attaque de redirection ou d'usurpation permette à un hôte malveillant disposant d'un certificat valide de fournir des données, en se faisant passer pour un hôte de confiance. Pour

garantir l'intégrité des données, le certificat doit être valide et concerner le site auquel on accède. Même si le produit tente de vérifier le nom d'hôte, il est toujours possible de le faire à tort. Par exemple, des attaquants pourraient créer un certificat dont le nom commence par un nom de confiance suivi d'un octet NUL, ce qui pourrait amener certaines comparaisons basées sur des chaînes de caractères à n'examiner que la partie contenant le nom de confiance. Cette faiblesse peut se produire même lorsque le produit utilise l'épinglage de certificats, si le produit ne vérifie pas le nom d'hôte au moment de l'épinglage d'un certificat. Cette faiblesse est due à la mise en œuvre d'une tactique de sécurité architecturale. Lorsque l'application utilise l'épinglage de certificats, le développeur peut ne pas valider correctement tous les composants pertinents du certificat avant d'épingler ce dernier. Il peut alors être difficile ou coûteux de procéder à des tests une fois l'épinglage terminé. Cette vulnérabilité n'est spécifique a un langage de programmation. L'attaquant peut avoir des privilèges ou usurper une identité. Avec cette attaque les données lues à partir du système attesté par le certificat peuvent ne pas provenir du système attendu. La confiance accordée au système en question, sur la base du certificat malveillant peut permettre des attaques par usurpation d'identité ou par redirection. Ci-dessous un exemple de code pour démontrer.

```
cert = SSL_get_peer_certificate(ssl);
    if (cert && (SSL_get_verify_result(ssl)==X509_V_OK)) {
    // do secret things
}
```

Même si l'étape de vérification renvoie X509_V_OK, cette étape n'inclut pas la vérification du nom commun par rapport au nom de l'hôte. En d'autres termes, rien ne garantit que le certificat correspond à l'hôte souhaité. La connexion SSL a pu être établie avec un hôte malveillant qui a fourni un certificat valide. Improper Validation of Certificate Expiration qui est le CWE 298 permet que l'expiration d'un certificat ne soit pas validée ou l'est de manière incorrecte, de sorte que la confiance peut être attribuée à des certificats qui ont été abandonnés en raison de leur ancienneté. Lorsque l'expiration d'un certificat n'est pas prise en compte, aucune confiance n'a nécessairement été transmise par le biais de ce certificat. Par conséquent, la validité du certificat ne peut pas être vérifiée et tout le bénéfice du certificat est perdu. Cette faiblesse est due à la mise en œuvre d'une tactique de sécurité architecturale. Lorsque le logiciel utilise l'épinglage de certificats, le développeur peut ne pas valider correctement tous les composants pertinents du certificat avant d'épingler ce dernier. Il peut alors être difficile ou coûteux de procéder à des tests une fois l'épinglage terminé. Cette vulnerabilité n'est spécifique à un langage de programmation. Les données lues à partir du système garanti par le certificat expiré peuvent être erronées en raison d'une usurpation d'identité malveillante. La confiance accordée au système en question , sur la base du certificat expiré peut permettre des attaques par usurpation d'identité. Ci-dessous un exemple de code pour démontrer.

```
if (cert = SSL_get_peer(certificate(ssl)) {
foo=SSL_get_verify_result(ssl);
if ((X509_V_OK==foo) || (X509_V_ERR_CERT_HAS_EXPIRED==foo))
```

```
4 //do stuff
```

Ce code OpenSSL garantit l'existence d'un certificat et permet l'utilisation de certificats expirés. Si l'appel à SSL_get_verify_result() renvoie X509_V_ERR_CERT_HAS_EXPIRED, cela signifie que le certificat a expiré. Au fur et à mesure que le temps passe, les chances de compromettre le certificat augmentent pour les attaquants.

Le CWE-299 qui est Improper Check for Certificate Revocation, une fois une application en est vulnérable ne vérifie pas ou vérifie incorrectement l'état de révocation d'un certificat, ce qui peut l'amener à utiliser un certificat qui a été compromis. Une vérification incorrecte de la révocation d'un certificat est une faille bien plus grave que les défaillances de certificats qui en découlent. En effet, l'utilisation d'un certificat révoqué est presque certainement malveillante. La raison la plus courante de la révocation d'un certificat est la compromission du système en question, ce qui signifie qu'aucun serveur légitime n'utilisera un certificat révoqué, à moins qu'il ne soit gravement désynchronisé. Cette faiblesse est due à la mise en œuvre d'une tactique de sécurité architecturale. Lorsque le produit utilise l'épinglage de certificats, le développeur peut ne pas valider correctement tous les composants pertinents du certificat avant d'épingler ce dernier. Il peut alors être diffi- cile ou coûteux de procéder à des tests une fois l'épinglage terminé. Cette vulnérabilité est spécifique à aucun langage de programmation. Avec cette attaque, l'attaquant peut avoir des privilèges ou usurper l'identité ;la confiance peut être attribuée à une entité qui n'est pas celle qu'elle prétend être. Des données provenant d'une source non fiable (et éventuellement malveillante) peuvent être intégrées ; les données d'application peuvent être lues. Des données peuvent être divulguées à une entité usurpant l'identité d'une entité de confiance, ce qui entraîne la divulgation d'informations. Ici, le contrôle d'accès, l'intégrité et la confidentialité sont mise en cause. Ci-dessous un exemple de code openssl pour illustrer. Ce code OpenSSL s'assure de l'existence d'un certificat avant de poursuivre l'exécution.

```
1 if (cert = SSL_get_peer_certificate(ssl)) {
2     // got a certificate, do secret things
3 }
```

Comme ce code n'utilise pas SSL_get_verify_results() pour vérifier le certificat, il pourrait accepter des certificats qui ont été révoqués (X509_V_ERR_CERT_REVOKED). Le produit pourrait communiquer avec un hôte malveillant. Avec le CWE-599 qui est Missing Validation of OpenSSL Certificate, L'application utilise OpenSSL et fait confiance ou utilise un certificat sans utiliser la fonction SSL_get_verify_result() pour s'assurer que le certificat satisfait à toutes les exigences de sécurité nécessaires. Cela pourrait permettre à un attaquant d'utiliser un certificat non valide pour prétendre être un hôte de confiance, d'utiliser des certificats expirés ou de mener d'autres attaques qui pourraient être détectées si le certificat était correctement validé. Cette faiblesse est due à l'absence d'une tactique de sécurité lors de la phase d'architecture et de conception. Cette faiblesse permet à l'attaquant de lire les données d'application ; la confiance accordée au système en question peut permettre des attaques par

usurpation d'identité ou par redirection. Si le certificat n'est pas vérifié, il est possible qu'une attaque de redirection ou d'usurpation permette à un hôte malveillant disposant d'un certificat valide de fournir des données en se faisant passer pour un hôte de confiance. Bien que l'attaquant en question puisse avoir un certificat valide, il peut simplement s'agir d'un certificat valide pour un site différent. Afin de garantir l'intégrité des données, nous devons vérifier que le certificat est valide et qu'il concerne le site auquel nous souhaitons accéder. Nous remarquons que la confidentialité et le contrôl d'accès des données est en péril. Ci-dessous un exemple de code à titre démonstratif. Ce code OpenSSL permet de s'assurer que l'hôte dispose d'un certificat.

```
1 if (cert = SSL_get_peer_certificate(ssl)) {
2 // got certificate, host can be trusted
3 //foo=SSL_get_verify_result(ssl);
4 //if (X509_V_OK==foo) ...
5 }
```

Notons que le code n'appelle pas SSL_get_verify_result(ssl), ce qui désactive effectivement l'étape de validation qui vérifie le certificat. Le CWE-322 qui est Key Exchange without Entity Authentication, avec ce dernier l'application effectue un échange de clés avec un acteur sans vérifier l'identité de cet acteur. L'échange de clés permet de préserver l'intégrité des informations envoyées entre deux entités, mais ne garantit pas que les entités sont bien celles qu'elles prétendent être. Cela peut permettre à un attaquant de se faire passer pour un acteur en modifiant le trafic entre les deux entités. En règle générale, un client victime contacte un serveur malveillant qui se fait passer pour un serveur de confiance. Si le client saute l'authentification ou ignore un échec d'authentification, le serveur malveillant peut demander des informations d'authentification à l'utilisateur. Le serveur malveillant peut alors utiliser ces informations d'authentification pour se connecter au serveur de confiance en utilisant les informations d'identification de la victime, pour renifler le trafic entre la victime et le serveur de confiance. Cette faiblesse est due à l'absence d'une tactique de sécurité lors de la phase d'architecture et de conception ; elle n'est spécifique à un langage de programmation. Avec cette faiblesse, l'attaquant peut contourner certains mécanismes de protection. La communication cryptée entre un utilisateur et un hôte de confiance peut faire l'objet d'un reniflage par tout acteur se trouvant sur le chemin de la communication. Nous n'allons pas nous arrêter sur le CVE alpaca et les CWE qui s'y rapportent ; nous allons aussi voir avec le site https://capec.mitre.org/data/definitions/459.html ; https://capec.mitre.org qui est une plateforme de référence, CAPEC qui est Common Attack Pattern Enumeration and Classification, une ressource communautaire pour identifier et comprendre les attaques. Sur le site web de CAPEC, comprendre comment l'adversaire opère est essentiel pour une cybersécurité efficace. CAPEC y contribue en fournissant un dictionnaire complet des schémas d'attaque connus employés par les adversaires pour exploiter les faiblesses connues des capacités cybernétiques. Il peut être utilisé par les analystes, les développeurs, les testeurs et les éducateurs pour faire progresser la compréhension de la com-

munauté et améliorer les défenses. Avec l'attaque Partern ID 459 qui est Creating a Rogue Certification Authority Certificate liée à alpaca attack, Un adversaire exploite une faiblesse résultant de l'utilisation d'un algorithme de hachage ayant une faible résistance aux collisions pour générer des demandes de signature de certificat (CSR) qui contiennent des blocs de collision dans leurs parties "à signer". L'adversaire soumet une CSR à la signature d'une autorité de certification de confiance, puis utilise le bloc signé pour faire apparaître un second certificat signé par ladite autorité de certification. En raison de la collision de hachage, les deux certificats, bien que différents, se hachent à la même valeur et le blob signé fonctionne donc tout aussi bien dans le second certificat. L'effet net est que le second certificat X.509 de l'adversaire, que l'autorité de certification n'a jamais vu, est maintenant signé et validé par cette autorité de certification. Le second certificat peut également être un certificat de signature. Ainsi, l'adversaire est en mesure de créer sa propre autorité de certification, ancrée dans la racine de confiance de l'autorité de certification légitime qui a signé le premier certificat X.509 de l'attaquant. Si l'autorité de certification initiale a été acceptée par défaut par les navigateurs, il en ira de même pour l'autorité de certification mise en place par l'adversaire et pour tous les certificats qu'elle signera. Par conséquent, l'adversaire est en mesure de générer n'importe quel certificat SSL pour usurper l'identité de n'importe quel serveur web, et le navigateur de l'utilisateur n'émettra aucun avertissement à l'intention de la victime. Cela peut être utilisé pour compromettre les communications HTTPS et d'autres types de systèmes où l'ICP et les certificats X.509 peuvent être utilisés (par exemple, VPN, IPSec). [5] [6] [3] Comment atténuer l'attaque alpaca n'est pas une tâche aussi facile, nous avons utilisé les ressources disponibles sur alpaca à partir du CVE de alpaca tout en prenant en compte les CWE qui s'y rapportent comme nous l'avons décrit plus haut. [8]

2.2 Matériel

Pour notre travail, nous avons eu besoin des ressources suivantes que nous avons utilisé à savoir :

Un ordinateur Portatif
Système d'exploitation Linux (Debian)
La connexion internet
Un routeur
L'outil d'analyse de trafic réseau Wireshark

Table 2.1 – Liste de matériels

Chapitre 3

Résultats

Introduction

Dans le chapitre précédent consacré à la conception et matériels, nous avons parlé de l'attaque alpaca, qu'est-ce que c'est que alpaca à partir d'un lab ; nous avons aussi parlé de alpaca à travers son CVE et les CWE qui y sont liés, nous avons clarifié en détails avec des exemples de codes à l'appui comment les applications sont vulnérables à alpaca à travers TLS ou encore le SSL sans oublier les conséquences, les impacts de ces vulnérabilités sur les applications ; nous avons vu également comment on peut atténuer ces attaques. Dans ce chapitre, nous allons tout en nous basant sur la méthodologie, élargir notre travail sur les sites web en productions au Bénin pour voir les vulnérabilités alpaca sur ces sites en restant dans le cadre légal.

Résultats

Le Web utilise entre autre le protocole HTTPS qui est Hypertext Transfer Protocol Secure, une combinaison du HTTP avec une couche de chiffrement comme SSL ou TLS. Alpaca code étant une attaque de confusion de contenu exploitant des serveurs TLS implémentant différents protocoles mais en utilisant des certificats compatibles, à partir du CVE et des CWE qui s'y rapportent, en utilisant les détails nous avons trouvé des sites web au Bénin qui sont vulnérables à alpaca attack. Pour notre travail, nous avons utilisé l'application Wireshark pour faire le scan de ces site webs que nous avons analysé suivant le CVE de alpaca et les CWE qui s'y rapportent. Les différents sites web que nous avons analysé sont les site web `https://www.uacstartupvalley.com/`, `https://fondationuac.org/`. Comme nous l'avons évoqué plus haut, wireshark pour le rappeler est en résumé un analyseur de réseau ou de protocole (également connu sous le nom de renifleur de réseau) disponible gratuitement sur le

site Web de Wireshark. Il est utilisé pour analyser la structure de différents protocoles réseau et a la capacité de démontrer l'encapsulation. Nous allons avant de continuer rentrer dans les détails du protocole TCP/IP qui est la base du HTTPS pour le transport de données à travers internet. C'est une norme de communication qui contribue au fonctionnement d'Internet. TCP/IP est un protocole de liaison de données utilisé sur Internet pour permettre aux ordinateurs et autres appareils d'envoyer et de recevoir des données. L'acronyme TCP/IP signifie Transmission Control Protocol/Internet Protocol. Il permet aux appareils connectés à Internet de communiquer entre eux via les réseaux. Le protocole TCP/IP détermine la façon dont les ordinateurs transfèrent les données d'un appareil à un autre. Ces données doivent rester fiables afin que le destinataire reçoive les mêmes informations que celles qu'a envoyées l'expéditeur. Qu'est-ce que le protocole TCP/IP et comment fonctionne-t-il ? Pour s'assurer que chaque communication arrive intacte jusqu'à son destinataire, le modèle TCP/IP décompose les données en paquets et les réassemble pour former un message complet à l'autre extrémité. Le fait d'envoyer les données sous forme de petits paquets permet une plus grande fiabilité que si toutes les données étaient envoyées en même temps. Lorsqu'un message est décomposé en paquets, les différents paquets qui le composent peuvent prendre des voies différentes si l'un des itinéraires est encombré. C'est un peu comme envoyer par courrier plusieurs cartes d'anniversaire à la même adresse. Toutes les cartes commencent leurs parcours au même endroit, c'est-à-dire chez vous, mais vous pouvez déposer chaque carte dans une nouvelle boîte aux lettres et elles peuvent donc toutes prendre un chemin différent pour arriver jusqu'à l'adresse du destinataire. Les caractéristiques du protocole TCP TCP (qui signifie Transmission Control Protocol, soit en français : Protocole de Contrôle de Transmission) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP). Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 (Pour savoir que le protocole en amont est TCP.). TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission. Les caractéristiques principales du protocole TCP sont les suivantes :

- TCP permet de remettre en ordre les datagrammes en provenance du protocole IP
- TCP permet de vérifier le flot de données afin d'éviter une saturation du réseau
- TCP permet de vérifier le flot de données afin d'éviter une saturation du réseau
- TCP permet de formater les données en segments de longueur variable afin de les "remettre" au protocole IP
- TCP permet de multiplexer les données, c'est-à-dire de faire circuler simultanément des informations provenant de sources (applications par exemple) distinctes sur une même ligne
- TCP permet enfin l'initialisation et la fin d'une communication de manière courtoise

Le but de TCP Grâce au protocole TCP, les applications peuvent communiquer de façon sûre

(grâce au système d'accusés de réception du protocole TCP), indépendamment des couches inférieures. Cela signifie que les routeurs (qui travaillent dans la couche Internet) ont pour seul rôle l'acheminement des données sous forme de datagrammes, sans se préoccuper du contrôle des données, car celui-ci est réalisé par la couche transport (plus particulièrement par le protocole TCP). Lors d'une communication à travers le protocole TCP, les deux machines doivent établir une connexion. La machine émettrice (celle qui demande la connexion) est appelée client, tandis que la machine réceptrice est appelée serveur. On dit qu'on est alors dans un environnement Client-Serveur. Les machines dans un tel environnement communiquent en mode connecté, c'est-à-dire que la communication se fait dans les deux sens. Pour permettre le bon déroulement de la communication et de tous les contrôles qui l'accompagnent, les données sont encapsulées, c'est-à-dire qu'on ajoute aux paquets de données un en-tête qui va permettre de synchroniser les transmissions et d'assurer leur réception. Une autre particularité de TCP est de pouvoir réguler le débit des données grâce à sa capacité à émettre des messages de taille variable, ces messages sont appelés segments. La fonction de multiplexage TCP permet d'effectuer une tâche importante : le multiplexage/démultiplexage, c'est-à-dire faire transiter sur une même ligne des données provenant d'applications diverses ou en d'autres mots mettre en série des informations arrivant en parallèle. Ces opérations sont réalisées grâce au concept de ports (ou sockets), c'est-à-dire un numéro associé à un type d'application, qui, combiné à une adresse IP, permet de déterminer de façon unique une application qui tourne sur une machine donnée. Le format des données sous TCP Quand nous prenons un segement TCP, il est constitué des champs suivants :

- Port Source (16 bits) : Port relatif à l'application en cours sur la machine source
- Port Destination (16 bits) : Port relatif à l'application en cours sur la machine de destination
- Numéro d'ordre (32 bits) : Lorsque le drapeau SYN est à 0, le numéro d'ordre est celui du premier mot du segment en cours.
- Lorsque SYN est à 1, le numéro d'ordre est égal au numéro d'ordre initial utilisé pour synchroniser les numéros de séquence (ISN)
- Numéro d'accusé de réception (32 bits) : Le numéro d'accusé de réception également appelé numéro d'acquittement correspond au numéro (d'ordre) du prochain segment attendu, et non le numéro du dernier segment reçu.
- Décalage des données (4 bits) : il permet de repérer le début des données dans le paquet. Le décalage est ici essentiel car le champ d'options est de taille variable
- Réservé (6 bits) : Champ inutilisé actuellement mais prévu pour l'avenir
- Drapeaux (flags) (6x1 bit) : Les drapeaux représentent des informations supplémentaires :

– URG : si ce drapeau est à 1 le paquet doit être traité de façon urgente. – ACK : si ce drapeau est à 1 le paquet est un accusé de réception. – PSH (PUSH) : si ce drapeau est à 1, le paquet fonctionne suivant la méthode PUSH. – RST : si ce drapeau est à 1, la connexion est

réinitialisée. – SYN : Le Flag TCP SYN indique une demande d'établissement de connexion. – FIN : si ce drapeau est à 1 la connexion s'interrompt.

- Fenêtre (16 bits) : Champ permettant de connaître le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- Somme de contrôle (Checksum ou CRC) : La somme de contrôle est réalisée en faisant la somme des champs de données de l'en-tête, afin de pouvoir vérifier l'intégrité de l'en-tête
- Pointeur d'urgence (16 bits) : Indique le numéro d'ordre à partir duquel l'information devient urgente
- Options (Taille variable) : Des options diverses
- Remplissage : On remplit l'espace restant après les options avec des zéros pour avoir une longueur multiple de 32 bits

Fiabilité des transferts ; le protocole TCP permet d'assurer le transfert des données de façon fiable, bien qu'il utilise le protocole IP, qui n'intègre aucun contrôle de livraison de datagramme. En réalité, le protocole TCP possède un système d'accusé de réception permettant au client et au serveur de s'assurer de la bonne réception mutuelle des données. Lors de l'émission d'un segment, un numéro d'ordre (appelé aussi numéro de séquence) est associé. A réception d'un segment de donnée, la machine réceptrice va retourner un segment de donnée dont le drapeau ACK est à 1 (afin de signaler qu'il s'agit d'un accusé de réception) accompagné d'un numéro d'accusé de réception égal au numéro d'ordre précédent. De plus, grâce à une minuterie déclenchée dès réception d'un segment au niveau de la machine émettrice, le segment est réexpédié dès que le temps imparti est écoulé, car dans ce cas la machine émettrice considère que le segment est perdu. Toutefois, si le segment n'est pas perdu et qu'il arrive tout de même à destination, la machine réceptrice saura grâce au numéro d'ordre qu'il s'agit d'un doublon et ne conservera que le dernier segment arrivé à destination.
Etablissement d'une connexion

Etant donné que ce processus de communication, qui se fait grâce à une émission de données et d'un accusé de réception, est basé sur un numéro d'ordre (appelé généralement numéro de séquence), il faut que les machines émettrices et réceptrices (client et serveur) connaîssent le numéro d'ordre initial de l'autre machine. L'établissement de la connexion entre deux applications se fait souvent selon le schéma suivant :

- Les ports TCP doivent être ouverts
- L'application sur le serveur est passive, c'est-à-dire que l'application est à l'écoute, en attente d'une connexion
- L'application sur le client fait une requête de connexion sur le serveur dont l'application est en ouverture passive. L'application du client est dite "en ouverture active" Les deux machines doivent donc synchroniser leurs séquences grâce à un mécanisme communément appelé three ways handshake (poignée de main en trois temps), que l'on retrouve aussi lors de la clôture de session.

Ce dialogue permet d'initier la communication, il se déroule en trois temps, comme sa dénomination l'indique. Dans un premier temps la machine émettrice (le client) transmet un segment dont le drapeau SYN est à 1 (pour signaler qu'il s'agit d'un segment de synchronisation), avec un numéro d'ordre N, que l'on appelle numéro d'ordre initial du client. Dans un second temps la machine réceptrice (le serveur) reçoit le segment initial provenant du client, puis lui envoie un accusé de réception, c'est-à-dire un segment dont le drapeau ACK est à 1 et le drapeau SYN est à 1 (car il s'agit là encore d'une synchronisation).Ce segment contient le numéro d'ordre de cette machine (du serveur) qui est le numéro d'ordre initial du client. Le champ le plus important de ce segment est le champ accusé de réception qui contient le numéro d'ordre initial du client, incrémenté de 1. Enfin, le client transmet au serveur un accusé de réception, c'est-à-dire un segment dont le drapeau ACK est à 1, dont le drapeau SYN est à zéro (il ne s'agit plus d'un segment de synchronisation). Son numéro d'ordre est incrémenté et le numéro d'accusé de réception représente le numéro d'ordre initial du serveur incrémenté de 1 Suite à cette séquence comportant trois échanges les deux machines sont synchronisées et la communication peut commencer ! Le principal défaut de TCP/IP est que les données qui circulent à l'aide de ce protocole ne sont pas cryptées. Il devient donc possible de renifler les données qui transigent sur le réseau et d'en lire le contenu. Le reniflage peut avoir lieu autant sur un réseau filaire que sur un réseau sans fil. Les utilisateurs malveillants utiliseront des logiciels spécialisés qui interagissent avec la carte réseau. Si le reniflage sur un réseau sans fil est limité à la portée du réseau, il en va autrement pour les réseau filaires, et de façon plus large, pour Internet. Et voici pourquoi. La force de TCP/IP, c'est sa souplesse. Les paquets de données arriveront à destination en empruntant différents chemins selon la disponibilité ou non des différents noeuds du réseau. La souplesse de TCP/IP a cependant un coût : puisque les paquets peuvent emprunter différents noeuds pour arriver à destination, il est possible de trafiquer les tables de routage afin de leur faire faire un détour qui permettra à un utilisateur malveillant de renifler les données qu'ils contiennent. C'est pourquoi les sites Web quelque soit leur sensibilité, comme le Web utilise le HTTP qui ne fonctionne pas sans le TCP, doivent absolument protéger les données qu'ils manipulent. [7] Il est donc important de sécuriser le protocole TCP/IP avec SSL (Secure Socket Layer).

En analysant les sites web suivant évoqué plus haut avec wireshark que nous avons décrit, nous avons les résultats suivant qui prouvent que ces sites sont vulnérables à l'attaque alpaca. Ci-dessous la présentation des résultats pour les analyses concernant le site https ://www.uacstartupvalley.com/. Avec l'outil Wireshark qui est un renifleur nous avons applique le filtre suivant afin d'avoir ce qui se passe quand on scan l'hote uacstartupvalley.com, http.host == uacstartupvalley.com and tcp.port == 80. Que fait en réalité ce filtre ? Ce filtre dit à l'outil Wireshark de faire le scan, avoir ce qui ce passe sur le port 80 (port HTTP) de l'hote uacstartupvalley.com ; avoir tout le trafic de l'hôte uacstartupvalley.com sur ce port. Dans les résultats nous avons les informations suivantes :

```
1 Frame 2: 852 bytes on wire (6816 bits), 852 bytes captured (6816 bits) on interface enol, id 0
```

```
Ethernet II, Src: Dell_84:2e:c9 (5c:26:0a:84:2e:c9), Dst: Tp-LinkT_52:76:c0 (28:ee:52:52:76:c0)
Internet Protocol Version 4, Src: 192.168.1.103, Dst: 162.247.243.29
Transmission Control Protocol, Src Port: 47594, Dst Port: 443, Seq: 1, Ack: 1, Len: 786
Source Port: 47594
Destination Port: 443
[Stream index: 0]
[TCP Segment Len: 786]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 667848260
[Next Sequence Number: 787 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 78964628
1000 .... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window: 501
[Calculated window size: 501]
[Window size scaling factor: -1 (unknown)]
Checksum: 0x5b5d [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (786 bytes)
Transport Layer Security
TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 781
Encrypted Application Data: 56920cb11063d5a177ffb5ea9c8d3bbfd46b49317c5501e6b519705c25a14
[Application Data Protocol: http-over-tls]
================================================================================
Frame 13: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eno1, id 0
Ethernet II, Src: Dell_84:2e:c9 (5c:26:0a:84:2e:c9), Dst: Tp-LinkT_52:76:c0 (28:ee:52:52:76:c0)
Destination: Tp-LinkT_52:76:c0 (28:ee:52:52:76:c0)
Source: Dell_84:2e:c9 (5c:26:0a:84:2e:c9)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.1.103, Dst: 108.177.15.188
0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 52
Identification: 0x8af6 (35574)
Flags: 0x40, 'Dont fragment
Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x7151 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.103
Destination Address: 108.177.15.188
Transmission Control Protocol, Src Port: 55330, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
Source Port: 55330
Destination Port: 443
[Stream index: 2]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2356695638
```

```
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 2602722185
1000 .... = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window: 501
[Calculated window size: 501]
[Window size scaling factor: -1 (unknown)]
Checksum: 0x3ea3 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
```

Nous savons avec ces informations et vu comment le protocole TCP/IP fonctionne que le site web https : //www.uacstartupvalley.com/ est vulnérable a alpaca attaque dont le CVE est CVE-2021-3618 aussi au CWE qui s'y rapportent.

De ces informations reçu, nous avons fait des analyses suivants comment fonctionne le protocole TCP/IP en ce qui concerne la fiabilité des données, l'autenticité et aussi les session (le temps imparti pour un packet ou une connexion sans oublier les données en état de transmission). Avec wireshark, nous avons au cours de la session, au cours de notre capture, plusieurs packets de divers types, prouvant déja que l'hôte reniflé est vulnérable à l'attaque alpaca. En effet, il est possible grâce à l'outil scarpy, disponible sur github mais aussi avec metasploitable disponible sur kali linux de créer des packets et plusieurs packets reçu au cours de la capture par ces derniers forgés à chaud ; ce qui compromettra les données tant au niveau de la couche application qu'au niveau réseau et matériel ; ce qui engendrera comme conséquence la lenteur des réseaux, des pannes inhabituelles, des conséquences grâves qui ne trouveront solutions que grâce à l'intervention d'experts internationnaux. En plus de cette perception, de ces failles à travers comment fonctionne le TCP ; dans le cas précedent il s'agit de l'ordre dans les packet ; nous avons d'autres perceptions qui suivent l'analyse que nous avons effectué. Entre autre ces perceptions, nous avons : Frame 2 ; ce qui nous a montré le numéro d'ordre du packet dans la liste ; connaissant de tel numéro d'ordre, il est très facile de compromettre la position des packets dans la liste de la session en cours par d'autres packets peut être bien de même numéro d'ordre, voir les ajouter ou simplement les supprimer ; tout ceci dans le but entre autre de ralentir le réseau ou de concourir à de vol de données voir des traffic bancaires. Nous avons aussi comme informations dans le packet, les adresses mac, ip de sources et de destinations, nous avons les ports sources et destinations qui met déja en évidence les services ou applications en communication avec aussi les sources et destinations identifiée tant physique que logique (IP). Avec ces détails, en se conformant au fonctionnement du protocole TCP/IP décrit plus haut, nous avons la possibilité en faisant ces captures de proche en proche étendre l'étendue réseau jusqu'à avoir avec sureté la topologie réseaux ainsi que les services en cours voir les applications sur les machines ainsi que leurs locations ; un grand danger pour les entreprises et personnes qui en sont vulnérable. Notre analyse ne s'est pas arrêté à ces niveaux, nous avons étendu notre analyse au niveau des sessions ; avec

les informations que nous avons des packets, il peut y avoir très facilement la compromission des certificats, l'utilisation de certificat volé d'un serveur pour un autre ; le vol de session, par exemple utiliser la session smtp pour un serveur FTP ou encore HTTP voir HTTPS. Tout ceci car nous avons les informations de sessions dans les packets reçus comme le TTL, les infos de connexion, le handshake (syn, ack, FIN et bien d'autres ; syn, ack déja suffisant pour avoir des sessions valides).

Les résultats du scan du second site web qui est https ://fondationuac.org suivant le filtre http.host == fondationuac.org and tcp.port == 80 avec wireshark sont ci-dessous.

```
Frame 14: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eno1, id 0
Ethernet II, Src: Dell_84:2e:c9 (5c:26:0a:84:2e:c9), Dst: Tp-LinkT_52:76:c0 (28:ee:52:52:76:c0)
Destination: Tp-LinkT_52:76:c0 (28:ee:52:52:76:c0)
Source: Dell_84:2e:c9 (5c:26:0a:84:2e:c9)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.1.103, Dst: 140.82.113.26
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 52

Identification: 0xc372 (50034)
Flags: 0x40, 'Dont fragment
Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0xb7d5 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.103
Destination Address: 140.82.113.26
Transmission Control Protocol, Src Port: 40016, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
Source Port: 40016
Destination Port: 443
[Stream index: 1]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2109117784
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 4109161943
1000 .... = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window: 501
[Calculated window size: 501]
[Window size scaling factor: -1 (unknown)]
Checksum: 0xbfa2 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[Timestamps]
================================================================================
Frame 44: 151 bytes on wire (1208 bits), 151 bytes captured (1208 bits) on interface eno1, id 0
Ethernet II, Src: Tp-LinkT_52:76:c0 (28:ee:52:52:76:c0), Dst: Dell_84:2e:c9 (5c:26:0a:84:2e:c9)
Destination: Dell_84:2e:c9 (5c:26:0a:84:2e:c9)
Source: Tp-LinkT_52:76:c0 (28:ee:52:52:76:c0)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 172.217.168.170, Dst: 192.168.1.103
```

```
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 137
Identification: 0x2419 (9241)
Flags: 0x00
Fragment Offset: 0
Time to Live: 49
Protocol: TCP (6)
Header Checksum: 0x4dc3 [validation disabled]
[Header checksum status: Unverified]
Source Address: 172.217.168.170
Destination Address: 192.168.1.103

Transmission Control Protocol, Src Port: 443, Dst Port: 53920, Seq: 1, Ack: 1, Len: 85
Source Port: 443
Destination Port: 53920
[Stream index: 3]
[TCP Segment Len: 85]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2458125721
[Next Sequence Number: 86 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 3500710368
1000 .... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window: 535
[Calculated window size: 535]
[Window size scaling factor: -1 (unknown)]
Checksum: 0x3587 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
TCP Option - No-Operation (NOP)
TCP Option - No-Operation (NOP)
TCP Option - Timestamps: TSval 1252589007, TSecr 3039408087
Kind: Time Stamp Option (8)
Length: 10
Timestamp value: 1252589007
Timestamp echo reply: 3039408087
[SEQ/ACK analysis]
[Timestamps]
TCP payload (85 bytes)
Transport Layer Security
TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 80
Encrypted Application Data: f0efe4c9e56242bb1960fdb4f2d1ed1d2b9cce5277ed7c35e3fcd4a8fb7c1
[Application Data Protocol: http-over-tls]
```

Le site fondationuac.org suit le même shéma vu les informations que nous avons recueillie ; comme pour le cas précedent, en rentrant dans les détails du fonctionnement du protocole TCP/IP , les attaquants peuvent se servir de ces informations disponibles pour voler des informations, faire des attaques Man in the Middle ; amener les gens à payer des rançons car ayant les adresses physique et logiques des machines, pouvant s'introduire facilement dans les réseaux des entreprises et créer des dégats, ces attaquants peuvent amener les gens

Chapitre 3. Résultats

à payer des rançons afin de les libérer ; des cas qui ne sont pas facile à gérer car très difficile ne connaissant pas celui avec qui l'on à faire.

Avec ces résultats, il est clair que le site fondationuac.org est vulnérable à l'attaque alpaca et au CWE qui s'y rapportent.

Chapitre 4

Discussion

Introduction

Dans le présent chapitre, il est question eu égard des résultats de nos analyses voir l'impact de l'attaque ALPACA sur les site web en production au Bénin, donner des approches de solutions et faire des recommandations.

Discussion

La sécurité des systèmes d'information et la cybercriminalité sont des problématiques de plus en plus importantes au Bénin, à mesure que l'Internet continue de se développer et que de nombreux sites web voient le jour. Les attaques ALPACA, en particulier, posent un risque significatif pour les sites web vulnérables, permettant aux utilisateurs malveillants de mener des attaques en usurpant des identités, volant des sessions TCP et perturbant les réseaux. Ces menaces sont d'autant plus préoccupantes étant donné que de nombreux réseaux au Bénin souffrent de négligence et de déficiences en matière de sécurité. Il est indéniable que les formations en informatique jouent un rôle crucial dans la lutte contre la cybercriminalité. Au Bénin, un grand nombre de professionnels et d'étudiants s'engagent dans des domaines liés à l'informatique, tels que l'analyse-programmation, la conception de sites web, la sécurité informatique et bien d'autres. Cependant, il est essentiel de se pencher sur la qualité de ces formations pour qu'elles soient reconnues internationalement et répondent aux normes de sécurité nécessaires. Les acteurs impliqués dans la qualification de ces formations doivent s'assurer de leur pertinence et de leur adéquation avec les enjeux actuels de la cybersécurité. De plus, des infrastructures telles que les cybercafés et les réseaux mal entretenus présentent des vulnérabilités importantes aux attaques ALPACA et autres formes de cyberattaques. La

négligence dans la maintenance de ces réseaux conduit souvent à des défaillances de sécurité, exposant les utilisateurs à des risques accrus. Pour contrer ces menaces, il est nécessaire de sensibiliser les utilisateurs et de les éduquer sur les bonnes pratiques en matière de sécurité informatique. Voici quelques idées générales qui pourraient être intégrées dans les campagnes de sensibilisation et de formation pour les populations béninoises :

- Utilisation sécurisée du Cyber Café : Les utilisateurs devraient être informés des précautions à prendre lorsqu'ils utilisent un Cyber Café pour accéder à Internet et effectuer des tâches sensibles, comme la gestion de comptes bancaires en ligne ou l'accès à des informations confidentielles.
- Sélection de logiciels fiables : Les utilisateurs doivent être guidés sur le choix de logiciels légitimes provenant de sources fiables pour éviter l'installation d'applications malveillantes.
- Protection du système d'exploitation : Sensibiliser les utilisateurs sur l'importance de maintenir leur système d'exploitation à jour et d'utiliser des logiciels de sécurité tels que des antivirus pour protéger leurs appareils contre les menaces potentielles.
- Licences logicielles : Insister sur l'acquisition légale de logiciels en payant pour des licences appropriées, afin de ne pas encourager l'utilisation de versions piratées contenant des vulnérabilités.
- Utilisation d'un VPN : Informer les utilisateurs sur l'utilisation d'un réseau privé virtuel (VPN) pour sécuriser leurs connexions Internet, en particulier lorsqu'ils utilisent des réseaux Wi-Fi publics.
- Compréhension des réseaux : Expliquer en termes simples ce qu'est un réseau et comment identifier des réseaux légitimes pour éviter de tomber entre les mains de pirates malveillants.
- Précautions avec les réseaux sans fil : Conseiller les utilisateurs sur les bonnes pratiques pour sécuriser leur réseau sans fil à domicile ou au bureau, en utilisant des mots de passe forts et en activant des fonctionnalités de sécurité telles que le chiffrement.

Il est essentiel d'impliquer les organismes gouvernementaux tels que l'Agence Nationale du Numérique, l'autorité de régulation des communications électroniques, des postes et de la distribution de la presse (ARCEP) ainsi que l'Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI) dans ces campagnes de sensibilisation et de formation. Leur soutien et leurs ressources peuvent contribuer à renforcer l'efficacité des initiatives de lutte contre la cybercriminalité. Cependant, malgré les efforts déployés par les organismes de sécurité, il est regrettable que les cas de piratage et d'autres crimes liés à Internet continuent de se produire. Il est donc primordial de renforcer la coopération entre les forces de l'ordre et les autres acteurs impliqués dans la sécurité des systèmes d'information pour une action plus coordonnée et efficace contre la cybercriminalité. Outre la sensibilisation du grand public, il est également important de mettre l'accent sur la responsabilité des entreprises et des organisations en matière de sécurité informatique. Les entreprises opérant au Bénin, qu'elles

Chapitre 4. Discussion

soient petites, moyennes ou grandes, doivent reconnaître leur rôle crucial dans la protection des données de leurs clients et employés. Les fuites de données, les violations de la vie privée et les cyberattaques peuvent avoir des conséquences désastreuses sur la réputation et la pérennité d'une entreprise. Il est donc impératif pour les organisations de mettre en œuvre des politiques de sécurité solides, de former leur personnel à la sécurité informatique et de mettre en place des mécanismes de prévention et de détection des cybermenaces. Un autre aspect essentiel à considérer est la collaboration entre le secteur public et le secteur privé pour renforcer la sécurité informatique. Les échanges d'informations sur les menaces potentielles, les vulnérabilités et les meilleures pratiques peuvent être bénéfiques pour tous les acteurs impliqués. Les entreprises peuvent bénéficier des connaissances et de l'expertise des organismes gouvernementaux en matière de sécurité, tandis que les organismes gouvernementaux peuvent mieux comprendre les défis et les besoins du secteur privé. Cette collaboration peut également contribuer à la création d'un écosystème plus sûr pour les entreprises qui envisagent d'investir au Bénin, renforçant ainsi la confiance dans le pays en tant que destination numérique fiable. En outre, le développement de talents dans le domaine de la cybersécurité est essentiel pour renforcer la posture de sécurité du Bénin. Les efforts de sensibilisation et de formation doivent s'étendre au-delà des utilisateurs finaux pour inclure la formation de professionnels de la sécurité informatique et des experts en cyberdéfense. Les institutions académiques, en collaboration avec le secteur privé et les organismes gouvernementaux, devraient concevoir des programmes de formation spécialisés pour répondre aux besoins spécifiques du marché du travail en matière de cybersécurité. En parallèle, le Bénin devrait encourager la recherche et le développement en matière de sécurité informatique. Investir dans des projets de recherche et des initiatives novatrices aidera à créer une communauté d'experts en sécurité capable de faire face aux défis émergents et aux menaces futures. Ces efforts pourraient également conduire à l'émergence de solutions technologiques locales pour renforcer la sécurité des infrastructures et des réseaux béninois. En ce qui concerne la coopération internationale, le Bénin devrait chercher à renforcer ses partenariats avec d'autres pays et organisations régionales et internationales œuvrant dans le domaine de la cybersécurité. Cela permettrait d'accéder à un partage d'expertise, de meilleures pratiques et de ressources pour contrer les menaces transfrontalières. Le Bénin pourrait également participer activement aux discussions internationales sur la sécurité informatique et contribuer à l'élaboration de normes et de politiques mondiales visant à promouvoir la cybersécurité. Outre les mesures préventives, il est également essentiel de développer des capacités de réponse rapide aux incidents de sécurité. La mise en place d'une équipe nationale de réponse aux incidents informatiques permettrait de coordonner efficacement les efforts de réponse en cas d'attaque cybernétique. Cette équipe serait chargée de détecter les incidents, d'analyser les menaces, de fournir des conseils aux victimes et de collaborer avec les forces de l'ordre pour enquêter sur les crimes numériques. Une telle initiative renforcerait considérablement la résilience du Bénin face aux attaques informatiques. Enfin, la protection des infrastructures critiques doit

Chapitre 4. Discussion

être une priorité absolue. Les secteurs stratégiques tels que les services de santé, l'énergie, les transports et les communications sont souvent la cible privilégiée des attaques informatiques. Le Bénin devrait mettre en place des mesures de sécurité spécifiques pour protéger ces infrastructures vitales contre les cybermenaces. Les audits de sécurité réguliers, la mise en œuvre de normes de sécurité rigoureuses et la sensibilisation du personnel à la sécurité informatique sont des éléments clés pour assurer la protection de ces secteurs sensibles. En tant qu'ingénieurs informaticiens, nous avons un rôle de premier plan à jouer dans cette lutte contre la cybercriminalité. Nous devons nous tenir informés des dernières avancées technologiques et des nouvelles tendances en matière de cyberattaques. En améliorant continuellement nos compétences en sécurité informatique, nous pouvons devenir des experts capables de concevoir et de mettre en œuvre des systèmes résilients et sécurisés. En tant que professionnel de l'informatique au Bénin, nous nous engageons à contribuer activement à la promotion de la sécurité informatique dans mon pays. Nous participerons aux formations spécialisées en sécurité informatique pour renforcer nos compétences et partagerons nos connaissances avec nos pairs et les membres de notre communauté. Nous soutiendrons les initiatives de sensibilisation à la sécurité informatique auprès du grand public, en mettant l'accent sur les bonnes pratiques et les mesures de prévention. De plus, nous travaillerons en étroite collaboration avec les organismes de régulation et les entreprises pour promouvoir l'adoption de normes de sécurité et encourager la mise en place de politiques de sécurité informatique solides. Ensemble, nous pouvons renforcer la résilience du cyberespace béninois et réduire les risques de cyberattaques. En conclusion, la cybersécurité est un défi majeur auquel le Bénin est confronté à mesure que l'Internet continue de se développer dans le pays. Les attaques ALPACA et d'autres formes de cybercriminalité menacent la confidentialité, l'intégrité et la disponibilité des données, ainsi que la stabilité des infrastructures numériques. Pour faire face à ces menaces, le Bénin doit adopter une approche globale et collaborative, impliquant tous les acteurs de la société. La sensibilisation du public, la formation des professionnels de l'informatique, l'amélioration des infrastructures, la coopération entre le secteur public et privé, la recherche et le développement, ainsi que la coopération internationale sont autant de piliers essentiels pour renforcer la cybersécurité du pays. Il est temps pour le Bénin de prendre des mesures concrètes et d'investir dans la sécurité informatique pour protéger ses citoyens, ses entreprises et ses infrastructures critiques des menaces en ligne. En agissant dès maintenant, le Bénin peut non seulement se prémunir contre les cyberattaques actuelles, mais aussi se positionner comme un leader en matière de cybersécurité dans la région et sur la scène internationale. La cybersécurité est une responsabilité partagée, et chacun doit jouer son rôle pour assurer un environnement numérique sûr et fiable. En unissant leurs forces, les autorités, les entreprises, les institutions académiques et la société civile peuvent créer une culture de sécurité informatique qui bénéficie à tous. En tant qu'ingénieurs informaticiens, nous avons la responsabilité de jouer un rôle actif dans cette lutte pour renforcer la sécurité informatique au Bénin. En investissant dans nos compétences, en partageant nos connaissances et en col-

Chapitre 4. Discussion

laborant avec d'autres acteurs, nous pouvons contribuer à créer un cyberespace béninois plus sûr et plus résilient. Seule une approche collective et proactive permettra de protéger efficacement les citoyens, les entreprises et les infrastructures contre les menaces en ligne. En nous engageant dans cette voie, nous pouvons promouvoir une culture de la sécurité informatique, favoriser l'innovation technologique et renforcer la confiance dans l'utilisation de l'Internet au Bénin. Ensemble, nous pouvons faire du Bénin un acteur solide et responsable dans le paysage numérique mondial, tout en préservant la confidentialité, l'intégrité et la disponibilité des données de nos concitoyens. La sécurité informatique est un enjeu critique pour l'avenir du Bénin, et en travaillant ensemble, nous pouvons relever ce défi avec succès.

Conclusion générale et perspectives

Le présent mémoire a mis en évidence les graves conséquences de l'attaque ALPACA, une menace inquiétante exploitant des vulnérabilités de protocoles de couche application dans les serveurs TLS. En profitant des certificats compatibles, les attaquants peuvent utiliser des attaques de type "Man In The Middle" pour rediriger le trafic d'un sous-domaine vers un autre, perturbant ainsi le mécanisme d'authentification TLS. Cette compromission de la sécurité peut conduire à des situations critiques, où des données sensibles peuvent être dérobées, des sessions volées, et des opérations illégales réalisées, menaçant gravement la confidentialité et l'intégrité des échanges. Les conséquences désastreuses de l'attaque ALPACA ne doivent pas être sous-estimées, d'autant plus que de nombreuses vulnérabilités connexes ont été recensées, reflétées par les CVE (Common Vulnerabilities and Exposures) associés à cette menace. La faille d'authentification qui en découle peut permettre aux attaquants de compromettre plusieurs protocoles au niveau de la couche application, créant ainsi un risque important pour les systèmes utilisant TLS. Au Bénin, où l'informatique est en plein essor, il est essentiel de prendre en compte ces vulnérabilités, en particulier lorsqu'il s'agit d'infrastructures informatiques en production. Des négligences peuvent malheureusement être observées dans certaines configurations, et l'accès à Internet sans la sécurité adéquate expose les systèmes et les utilisateurs à des risques considérables. Cependant, il est rassurant de constater que des efforts significatifs ont été déployés pour lutter contre ces menaces. La police et les organes gouvernementaux du Bénin sont engagés dans la protection des infrastructures informatiques et la sensibilisation à la sécurité. Néanmoins, face à la sophistication croissante des attaques, il est primordial de maintenir une vigilance constante et de promouvoir des pratiques exemplaires en matière de sécurité informatique. Ce mémoire met également en lumière l'importance de la sensibilisation à la sécurité chez les utilisateurs et les administrateurs. Une bonne compréhension des risques liés à l'attaque ALPACA et aux vulnérabilités de TLS peut aider à prévenir les erreurs humaines et à renforcer la protection des données et des communications. En conclusion, la lutte contre l'attaque ALPACA et les vulnérabilités d'authentification TLS est un enjeu majeur pour garantir la sécurité des échanges en ligne. En appliquant des mesures de sécurité appropriées, en maintenant les systèmes à jour, et en développant une culture de sécurité informatique, il est possible de réduire les risques associés

à cette menace et de renforcer la cybersécurité dans un contexte numérique en perpétuelle évolution.

Bibliographie

[1] M. Brinkmann. Alpaca : Application layer protocol confusion - analyzing and mitigating cracks in tls authentication. 2012.

[2] Z. Zhong. Alpaca : A large-scale privacy-aware collaborative anti-spam system. 2010.

Webographie

[3] CAPEC - CAPEC-459 : Creating a Rogue Certification Authority Certificate (Version 3.9). https://capec.mitre.org/data/definitions/459.html. Consulté en Février 2023.

[4] Certificats X509 v3 - cryptosec. https://www.cryptosec.org/?Certificats-X509-v3. Consulté en Avril 2023.

[5] CVE-2021-3618 : ALPACA is an application layer protocol content confusion attack, exploiting TLS servers implementing different protocol. https://www.cvedetails.com/cve/CVE-2021-3618/. Consulté en Février 2023.

[6] CWE - CWE-295 : Improper Certificate Validation (4.11). https://cwe.mitre.org/data/definitions/295.html. Consulté en Février 2023.

[7] Le protocole TCP. https://web.maths.unsw.edu.au/~lafaye/CCM/internet/tcp.htm. Consulté en Mars 2023.

[8] Protéger les données qui transigent sur le réseau (https - certificat numérique SSL). https://apical.xyz/fiches/securiser_un_site_wordpress_002/Proteger_les_donnees_qui_transigent_sur_le_reseau. Consulté en Mars 2023.

[9] The Transport Layer Security (TLS) Protocol Version 1.2. https://datatracker.ietf.org/doc/html/rfc5246. Consulté en Juillet 2023.

Table des matières

Dédicace ii

Remerciements iii

Liste des abréviations iv

Liste des tableaux v

Table des figures vi

Résumé ix

Abstract xi

Introduction 1

0.1 Problématique . . . 1
0.2 Contexte et justification . . . 2
0.3 Objectifs . . . 3
0.4 Organisation du mémoire . . . 3

1 Généralités et concepts de base 4

GÉNÉRALITÉS . . . 4
1.1 TLS et certificats X.509 . . . 4
1.1.1 STARTTLS . . . 4
1.1.2 Certificats de serveur . . . 4
1.1.3 Indication du nom du serveur (SNI) . . . 5
1.1.4 Négociation du protocole de la couche application (ALPN) . . . 5
1.1.5 Certificats X509 v3 . . . 5
1.1.6 Rôles des certificats X.509 dans l'authentification et la confidentialité des communications . . . 6
1.2 Protocoles de la couche d'application . . . 6
1.3 Attaques inter-protocoles basées sur TLS . . . 8
1.4 Attaques inter-protocoles sur HTTPS . . . 8
1.4.1 Méthodes d'attaque . . . 9

1.4.2 Attaques par déni de service (DoS) et par déni de service distribué (DDoS) . . . 9
1.4.3 Attaque teardrop . . . 10
1.4.4 Attaque Smurf . . . 10
1.4.5 Attaque par mot de passe . . . 10
1.4.6 Tolérance des navigateurs Web . . . 11
1.4.7 Configuration du laboratoire . . . 12

2 Conception et matériels **13**
Conception et matériels . . . 13
2.1 Méthodologie . . . 13
2.1.1 Analyse critique des attaques interprotocoles . . . 13
2.1.2 Présentation de l'attaque ALPACA . . . 14
2.1.3 Atténuation de l'attaque ALPACA . . . 21
2.2 Matériel . . . 30

3 Résultats **31**

4 Discussion **41**

Conclusion générale et perspectives **46**

Bibliographie **48**

Webographie **49**

Table des matières **50**