NOM 1

CODEPERMANENT

NOM2

CODEPERMANENT

Sécurité des logiciels et exploitation de vulnérabilités

INF600C

Hiver 2019

Travail Pratique 2

Write Ups

Présenté à

M. Philippe Pépos-Petitclerc

Université du Québec à Montréal

Assiette

Il nous est donné le programme avec bit setuid, *assiette*, qui semble nous donner une liste de plats, dont seul flag semble inaccessible. Nous n'avons pas non plus de permissions d'écriture dans l'arborescence du défi. Bon, ça fait rien, mais on est en train de jouer la paix du royaume...

Étant donné qu'os.realpath résout les liens symboliques, le programme utilise le vrai chemin absolu et peut donc lister les différents plats. Par contre, lorsqu'on rentre le nom du plat en lancant le programme en dehors du dossier où il se trouve, il nous répond que notre répertoire plats n'existe pas, en effet, le programme utilise un chemin relatif à partir du dossier courant vers le dossier *plats* pour afficher le contenu des fichiers.

Qu'à cela ne tienne, on va régler la question du traité au jeu. On crée un dossier *plats* dans un répertoire que l'on contrôle. Dans ce même dossier, nous créons un lien symbolique nommé *fruits*, puisque le programme python l'accepte, et que le mot *flag* n'est pas accepté par le programme.

Ce lien pointe vers */quetes/02-sys/assiette/plats/flag*. Ainsi, le programme va suivre le lien symbolique nommé *fruits* et grâce à son setuid, devrait pouvoir ouvrir pour nous le fichier *flag*. Visiblement, ça lui fait plaisir et il nous sert le délicieux plat *flag* :

Flag

FLAG{assiette_fromage}

CWE Associés

- CWE-610: Externally Controlled Reference to a Resource in Another Sphere
  - Le CWE 610 couvre la manipulation de chemin et tout type de ressources, y compris les symbolic links.
- CWE-59: Improper Link Resolution Before File Access ('Link Following') et CWE-61: UNIX Symbolic Link (Symlink) Following
  - Ces deux CWE s'appliquent, un faux fichier dans un faux répertoire, qui est un symlink vers le vrai fichier et pas de vérification à l'ouverture.

Références

- https://stackoverflow.com/questions/19643223/launching-a-python-script-via-a-symbolic-link ainsi que https://docs.python.org/3/library/os.path.html
  - Ceci nous permet de comprendre comment la fonction os.realpath fonctionne.
- https://mail.python.org/pipermail/python-list/2013-May/647035.html
  - Permet de comprendre la relation de la fonction avec les symlinks
- http://vimeo.com/66810725, pour les citations

Corrections

- Le programme devrait enlever le setuid si lancé par un lien symbolique ou devrait arrêter son exécution si tel est le cas.
- Le programme pourrait s'assurer de son emplacement, en faisant un cd dans le bon répertoire au démarrage, par exemple.

Web1 - Casino De Drummond

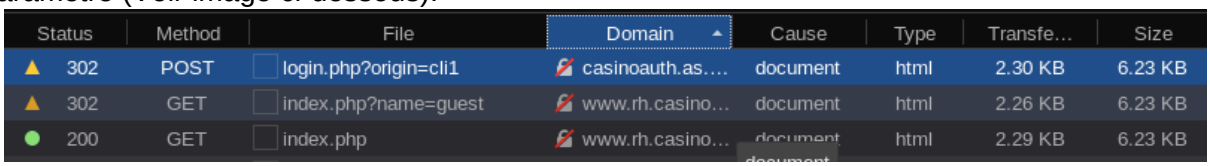
Il nous est présenté la magnifique page d'accueil des ressources humaines du Casino de Drummond. En inspectant le code de la page, nous remarquons que seul le lien de '*Login*' mène vers une page, tous les autres liens ont un attribut href vide.

Ce lien nous amène vers une page de connexion avec un formulaire prenant un mot de passe et un nom d'utilisateur. Nous savons qu'il est possible de se connecter avec les crédentiels '*guest:guest*', ce que nous faisons. Nous sommes redirigés vers la page précédente, mais identifié en tant que '*guest*'

Un cookie 'PHPSESSID' est mis par la page. Nous savons donc que l'application est probablement écrite en PHP. Nous avons d'abord tenté de décoder le cookie, qui est en base64, afin de voir s'il contiendrait des informations intéressantes. Ce n'est pas le cas, il est donc probable que l'implémentation par défaut des identifiants de session soit utilisée ici.

Nous sommes ensuite retournés au formulaire pour tenter de l'injection SQL. Celui-ci semble protégé contre l'injection, car la seule erreur obtenue est que le nom d'utilisateur ou le mot de passe sont invalides.

En nous identifiant à nouveau en tant que '*guest*' et en observant le trafic résultant, nous avons remarqué que le formulaire de connexion envoie les données en POST pour valider le mot de passe et le nom d'utilisateur, puis redirige vers la page d'accueil avec seulement un paramètre '*name*' contenant le nom d'utilisateur qui redirige vers elle-même en enlevant le paramètre (Voir image ci-dessous).



Status	Method	File	Domain	Cause	Type	Transfe…	Size
302	POST	login.php?origin=cli1	casinoauth.as…	document	html	2.30 KB	6.23 KB
302	GET	index.php?name=guest	www.rh.casino…	document	html	2.26 KB	6.23 KB
200	GET	index.php	www.rh.casino…	document	html	2.29 KB	6.23 KB

Nous avons simplement envoyé une nouvelle requête vers la page d'accueil avec la valeur '*admin*' pour le paramètre '*name*' sans passer par le formulaire de connexion et nous étions identifiés en tant que '*admin*'.

Flag

FLAG{AtDrummondCasinoYouReNotANumberButAName}

Au casino de Drummond, nous sommes identifiés par un nom (sans réelle validation) et uniquement un nom plutôt que par un numéro, comme un identifiant de session par exemple.
Il s'agit probablement du casino qui a le plus confiance en ses clients dans le monde entier.

CWE Associés

- CWE-287: Improper Authentication, CWE-288: Authentication Bypass Using an Alternate Path or Channel et CWE-304: Missing Critical Step in Authentication
  - Il est possible de s'identifier sans avoir à passer par le formulaire de connexion en envoyant les bons paramètres dans une requête vers la page d'accueil. Ceci est une route alternative qui permet de s'authentifier sans passer par la validation du mot de passe.
- CWE-306: Missing Authentication for Critical Function
  - L'authentification permettant d'accéder au flag en mode administrateur est brisée.

Corrections

- Utiliser les identifiants de session pour identifier l’utilisateur après la connexion plutôt que d’utiliser des paramètres dans la requête.
- Ne pas permettre de se connecter sans passer par le formulaire de connexion.
- Supprimer l’identifiant de session à la déconnexion.