

Projet RT0706

Table des matières

Projet RT0706.....	1
Access control vulnerabilities.....	2
Challenge 1 - Unprotected admin functionality.....	2
Challenge 2 - Unprotected admin functionality with unpredictable URL.....	3
Challenge 3 - User role controlled by request parameter.....	3
Business-logic-vulnerabilities	5
Challenge 1 - Excessive trust in client-side controls.....	5
Challenge 2 - High-level logic vulnerability	7
Challenge 3 - Inconsistent security controls.....	8

Access control vulnerabilities

Challenge 1 - Unprotected admin functionality

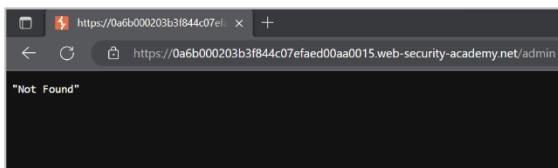
Contexte

Le site a une page d'administrateurs non protégée. Le but étant de supprimer l'utilisateur Carlos. Quand on lance le challenge, une page web de vente s'affiche.

Vulnérabilité / problème

D'après la documentation nous avons testé dans un premier temps l'escalade verticale des privilèges. L'escalade verticale des privilèges est quand les fonctions administratives ne sont pas seulement accessibles à partir de la page d'accueil d'un administrateur, mais également à partir de la page d'accueil d'un utilisateur.

Exploitation



Nous avons testé d'afficher une potentielle page d'administrateur avec l'url suivante :

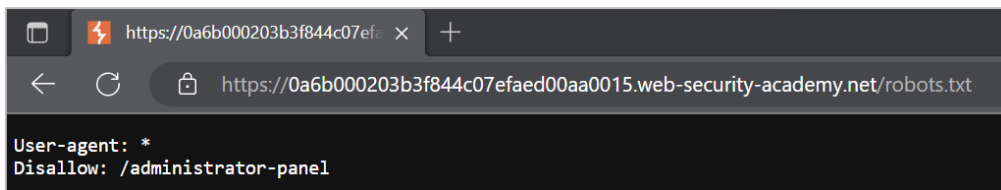
<https://0a6b000203b3f844c07efaed00aa0015.web-security-academy.net/admin>

Mais la page n'était pas référencée.

Nous avons donc regardé le contenu du fichier robots.txt. En effet, dans certains cas, l'URL d'administration peut être divulguée à d'autres emplacements comme ce dernier fichier. Nous avons rentré l'URL suivante :

<https://0a6b000203b3f844c07efaed00aa0015.web-security-academy.net/robots.txt>

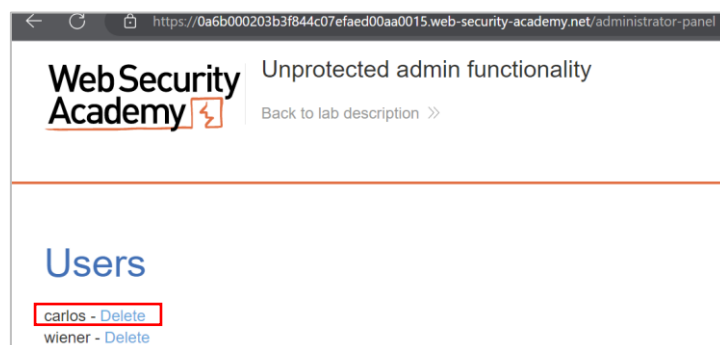
Ce fichier nous apprend qu'une page est non indexée :



Nous avons donc essayé d'accéder à cette dernière page avec l'URL :

<https://0a6b000203b3f844c07efaed00aa0015.web-security-academy.net/administrator-panel>

Cette page nous donne accès à des fonctionnalités d'administrateurs, tel que supprimer l'utilisateur Carlos.



Conseils de corrections

Il ne faut pas utiliser le fichier robots.txt pour empêcher l'affichage des informations sensibles (comme la possibilité de supprimer des utilisateurs en tant que simple utilisateur) dans les résultats

des moteurs de recherche. Il serait nécessaire d'utiliser une protection par mot de passe ou la directive `meta noindex`.

Challenge 2 - Unprotected admin functionality with unpredictable URL

Contexte

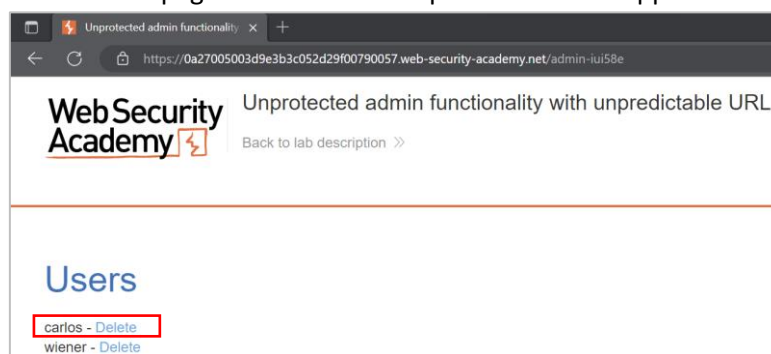
Le site a une page d'administrateurs non protégée. Le but étant de supprimer l'utilisateur Carlos. Quand on lance le challenge, une page web de vente s'affiche.

Vulnérabilité / problème

Dans certains cas, les fonctionnalités sensibles ne sont pas protégées de manière robuste mais sont dissimulées en leur donnant une URL moins prévisible : ce qu'on appelle la sécurité par obscurité. Le simple fait de masquer des fonctionnalités sensibles ne fournit pas un contrôle d'accès efficace, car les utilisateurs peuvent toujours découvrir l'URL obscurcie de différentes manières.

Exploitation

Après avoir ouvert les outils de développement du navigateur, dans l'onglet « Sources » le fichier « index » nous avons remarqué du code Javascript. En regardant dans la condition « `if (isAdmin)` » nous avons remarqué la ligne « `adminPanelTag.setAttribute('href', '/admin-iui58e');` ». Cette page devrait être accessible si la condition est `isAdmin` vérifiée, c'est-à-dire si nous étions administrateur. Nous avons donc rentré l'URL qui ne devrait être visible que pour les administrateurs : <https://0a27005003d9e3b3c052d29f00790057.web-security-academy.net/admin-iui58e>. Ainsi, nous avons eu accès à la page d'administration permettant de supprimer les utilisateurs :



Conseils de corrections

Il serait souhaitable de ne pas stocker des informations sensibles côté client, ainsi que de faire attention aux informations visibles dans les scripts. La page devrait être accessible par une authentification. Nous pouvons envisager un langage côté serveur plutôt que côté client.

Challenge 3 - User role controlled by request parameter

Contexte

Ce laboratoire dispose d'une page d'administration accessible en rajoutant « `/admin` » à la fin de l'URL du site web du challenge. Cette page identifie les administrateurs à l'aide d'un cookie falsifiable. Le but de ce défi est d'accéder au panneau d'administration et de supprimer l'utilisateur Carlos. Nous avons accès à notre propre compte (identifiant : `wiener`, mot de passe `peter`).

Vulnérabilité / problème

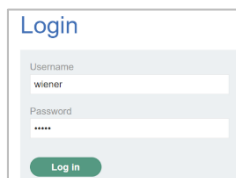
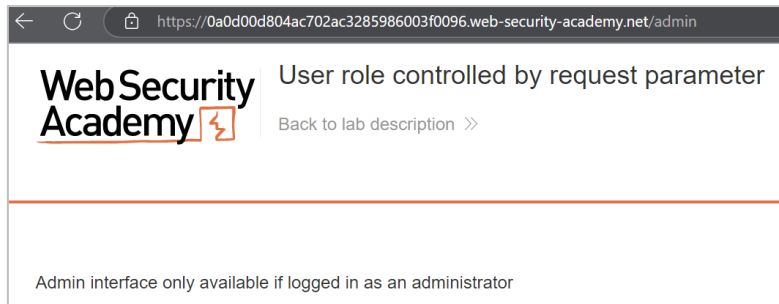
Certaines applications déterminent les droits d'accès ou le rôle de l'utilisateur lors de la connexion, puis stockent ces informations dans un emplacement contrôlable par l'utilisateur, tel qu'un champ masqué, un cookie ou un paramètre de chaîne de requête prédéfini. L'application prend des décisions de contrôle d'accès ultérieures en fonction de la valeur soumise. Cette approche est

fondamentalement non sécurisée car un utilisateur peut simplement modifier la valeur et accéder à des fonctionnalités auxquelles il n'est pas autorisé, telles que les fonctions administratives

Exploitation

Dans un premier temps nous avons, comme énoncé dans le sujet, essayé d'accéder à la page administrative avec l'URL : <https://0a0d00d804ac702ac3285986003f0096.web-security-academy.net/admin>

La page ne nous est pas accessible puisque nous ne sommes pas logués en tant qu'administrateur :



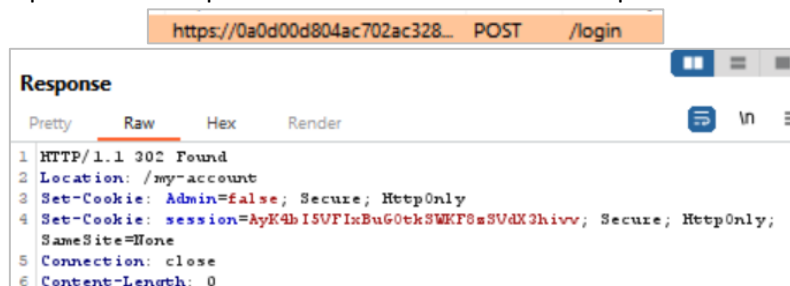
Nous avons donc essayé de nous connecter grâce au formulaire accessible depuis la page « My account ».

Une fois connectés l'URL est devenue :

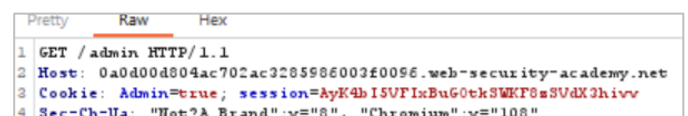
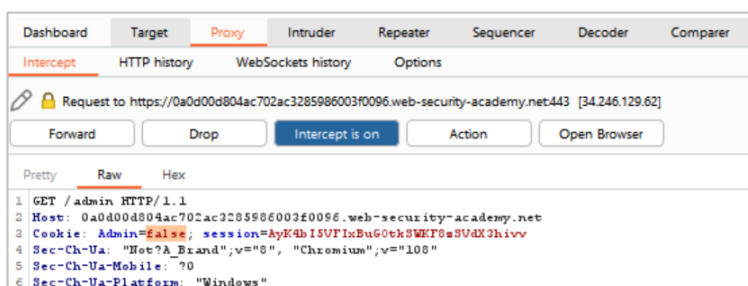
<https://0a0d00d804ac702ac3285986003f0096.web-security-academy.net/my-account?id=wiener>

Nous avons essayé de modifier la valeur du champ « id » avec « admin / administrator » mais sans grand succès.

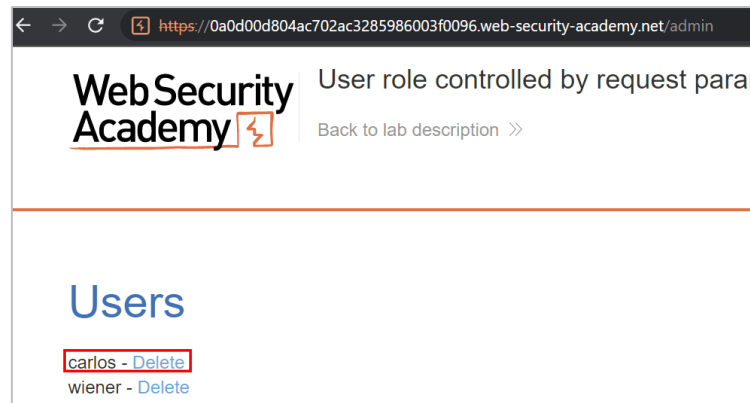
Nous avons donc utilisé Burp Suite Community pour intercepter les échanges et voir s'il était possible de modifier des valeurs afin de nous connecter en tant qu'administrateur. Une fois reconnectés dans le navigateur web de Burp nous avons lancé l'intercepteur et loader la page /admin. Nous avons remarqué que la réponse de la requête avait un cookie avec comme paramètre Admin=false.



Donc nous avons voulu changer la réponse de cette requête en changeant la valeur du paramètre Admin à vrai. Pour cela nous nous sommes délogués et relogués. Avant de passer l'intercepteur à OFF nous avons changé la valeur du cookie présent dans la réponse.



Cela nous a permis d'afficher la page d'admin avec l'URL donné précédemment.



Conseils de corrections

Il serait souhaitable de vérifier la valeur, côté serveur, de la session et non la valeur du paramètre Admin qui est trop facilement modifiable.

Business-logic-vulnerabilities

Challenge 1 - Excessive trust in client-side controls

Contexte

Cet atelier ne valide pas correctement les entrées des utilisateurs. Il faut donc exploiter une faille dans le flux d'achat pour acheter des articles à un prix non conventionnel. Pour résoudre ce challenge, il faut acheter l'article suivant : Lightweight l33t leather jacket. Comme pour le challenge précédent nous avons comme accès notre propre compte (identifiant : wiener, mot de passe peter).

Vulnérabilité / problème

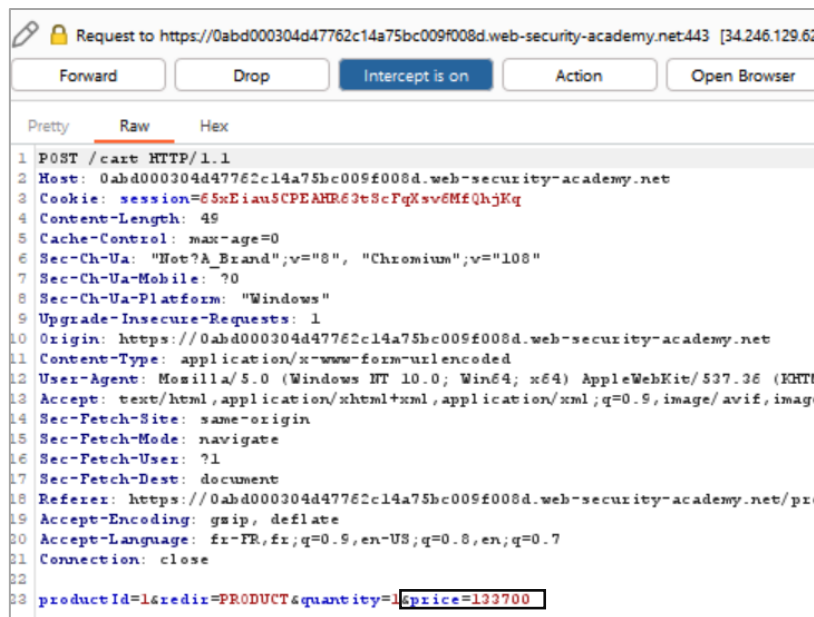
Des failles dans la logique peuvent permettre aux attaquants de contourner les règles. Par exemple, l'utilisateur peut être en mesure d'effectuer une transaction sans passer par le flux de travail d'achat prévu. Cela signifie que lorsqu'un attaquant s'écarte du comportement attendu de l'utilisateur, l'application ne prend pas les mesures appropriées pour empêcher cela et, par la suite, ne parvient pas à gérer la situation en toute sécurité.

Exploitation

Pour ce challenge nous avons utilisé Burp. Nous avons lancé le navigateur avec l'URL du challenge. Dans un premier temps nous nous sommes connectés avec les logins fournis. Nous avons remarqué

Store credit:	que nous avons 100€ de crédit. Nous avons ensuite
\$100.00	rajouté dans notre panier l'article demandé dans
Cart	l'énoncé. Cet article coûte 1337\$, nous n'avons donc
Not enough store credit for this purchase	pas assez d'argent sur notre compte pour poursuivre la transaction.

Nous avons enlevé l'article de notre panier et lancé l'intercepteur de Burp pour voir les requêtes. Nous avons rajouté de nouveau l'article en question dans notre panier et regarder la requête engendrée par cette action :



Nous avons remarqué que le prix était affiché dans la requête et qu'il était possible de le modifier par une valeur moindre.

productId=1&redirect=PRODUCT&quantity=1&price=4

Nous avons arrêté l'intercepteur sur Burp. Une fois retourné dans notre onglet panier nous avons remarqué que le prix retenu était bien celui que nous avions renvoyé.

Store credit:		
\$100.00		
Cart		
Name	Price	Quantity
Lightweight "I33t" Leather Jacket	\$0.04	- 1 + Remove

Nous avons pu valider le challenge en procédant à la transaction puisque nous avons ainsi assez de crédit.

Store credit:		
\$99.96		
Your order is on its way!		
Name	Price	Quantity
Lightweight "I33t" Leather Jacket	\$1337.00	1
Total: \$0.04		

Conseils de corrections

Le prix ne devrait pas être fixé/donné par la requête mais devrait être stocké et géré sur le serveur. En effet, la requête devrait seulement récupérer l'ID de l'article pour qu'une requête SQL puisse retrouver le prix associé dans la base de données. Ainsi il ne serait pas possible de modifier le prix.

Challenge 2 - High-level logic vulnerability

Contexte

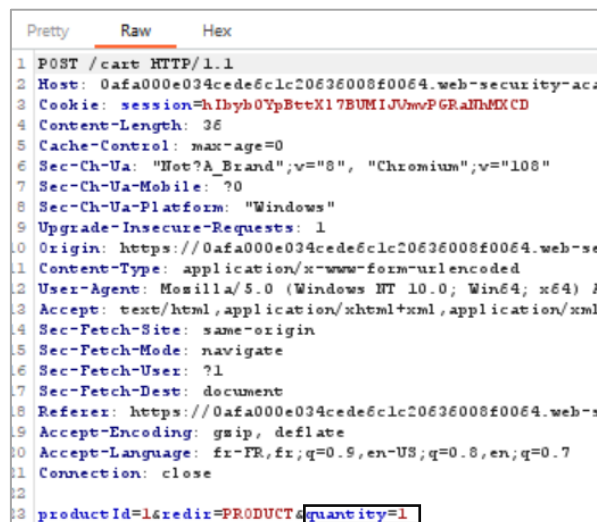
Cet atelier ne valide pas correctement les entrées des utilisateurs. Il faut donc exploiter une faille dans le flux d'achat pour acheter des articles à un prix non conventionnel. Pour résoudre ce challenge, il faut acheter l'article suivant : Lightweight I33t leather jacket. Comme pour le challenge précédent nous avons comme accès notre propre compte (identifiant : wiener , mot de passe peter).

Vulnérabilité / problème

Ce challenge a une faille semblable au challenge précédent. En effet, nous avons à faire à une faille de Flaw Logique également qui permettent aux attaquants de contourner les règles. Par exemple, l'utilisateur peut être en mesure d'effectuer une transaction sans passer par le flux de travail d'achat prévu.

Exploitation

Pour ce challenge nous avons également utilisé Burp. Nous avons lancé le navigateur avec l'URL du challenge. Dans un premier temps nous nous sommes connectés avec les logins fournis. Nous avons remarqué que nous avions 100€ de crédit. Avant de rajouter à notre panier l'article demandé nous avons lancé l'intercepteur de Burp. Nous n'avons plus accès au prix comme au challenge précédent. Cependant nous avons accès à la quantité.



```

1 POST /cart HTTP/1.1
2 Host: 0afa000e034cede6c1c20636008f0064.web-security-academy
3 Cookie: session=hIbyh0YpBttXl7EUMIJUmvPGRaJhMXCD
4 Content-Length: 36
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="108"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0afa000e034cede6c1c20636008f0064.web-security-academy
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0afa000e034cede6c1c20636008f0064.web-security-academy/
19 Accept-Encoding: gzip, deflate
20 Accept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7
21 Connection: close
22
23 productId=1&redir=PRODUCT&quantity=1
  
```

Une idée est de passer cette quantité en négatif. Nous l'avons donc passée à -4. Nous remarquons que dans notre panier le prix de la transaction est de -4011\$. Nous avons essayé de passer la transaction pour voir si l'argent nous serait « remboursé ». Mais une sécurité vérifie que la commande ne soit pas négative.

Store credit:
\$100.00

Cart

Cart total price cannot be less than zero

Cependant le fait qu'il y ait un nombre négatif ne semble pas poser de problème. Nous avons donc modulé nos achats afin d'avoir un total inférieur à 100\$:

- Nous avons ajouté 1X : Lightweight I33t leather jacket

- Nous avons ajoutés -13X : Your Virtual Journey Starts Here, en suivant la méthode testée juste au-dessus.

Name	Price	Quantity
Lightweight "l33t" Leather Jacket	\$1337.00	- 1 + Remove
Your Virtual Journey Starts Here	\$96.43	- -13 + Remove

Coupon:
 Apply

Total: \$83.41

Nous avons validé la commande et nous avons ainsi pu valider le challenge.

Store credit:
\$16.59

Your order is on its way!

Name	Price	Quantity
Lightweight "l33t" Leather Jacket	\$1337.00	1
Your Virtual Journey Starts Here	\$96.43	-13

Total: \$83.41

Conseils de corrections

La quantité ne devrait pas être fixée/donnée par la requête mais devrait être stockée et gérée sur le serveur. En effet, la requête devrait seulement récupérer l’ID de l’article pour qu’une requête SQL puisse retrouver le nombre d’article correspond à cet id. La quantité du stock devrait être vérifiée et jamais négative dans la base de données côté serveur. Cela implique qu’une vérification sur le serveur des quantités devrait avoir lieu. Il ne devrait pas être possible de commander un article en valeurs négatives. La page web ne devrait être qu’affichage et ne pas avoir autant d’impact sur les commandes qui devraient être gérées sur le serveur.

Challenge 3 - Inconsistent security controls

Contexte

Ce challenge nous permet de travailler sur une vulnérabilité logique qui permet aux utilisateurs d’accéder à des fonctionnalités administratives qui devraient seulement être disponibles pour les employés de l’entreprise. Pour résoudre ce challenge, il faut accéder à la page d’administration et supprimer Carlos.

Vulnérabilité / problème

Ce challenge a une faille semblable au challenge précédent. En effet, nous avons à faire à une faille de Flaw Logique également qui permettent aux attaquants de contourner les règles. Trop d’informations sont données qui permettent de comprendre et donc de contourner la sécurité mise en place sur le site.

Exploitation

Nous avons essayé dans un premier temps d'afficher la page :

<https://0ae30055048796bdc4d33fee001d001c.web-security-academy.net/admin>

Un message s'affiche indiquant qu'il faut être connecté en tant que l'utilisateur DontWannaCry.

Admin interface only available if logged in as a DontWannaCry user

Nous avons essayé de nous créer un utilisateur avec ce nom. Pour cela nous avons rentré le nom d'utilisateur que nous souhaitions (DontWannaCry) et utilisé l'adresse mail fourni, sur la page accessible par le bouton « Email client », pour créer notre compte. Nous ne pouvions pas directement utiliser @dontwannacry.com puisque nous n'avons pas accès à cette boîte mail et nous n'aurions pas pu valider notre compte.

Register

If you work for DontWannaCry, please use your @dontwannacry.com email address

Username
DontWannaCry

Email
attacker@exploit-0a1200640420964bc48d41fb01dd003a.exploit-server.net

Password

Register

Il nous était demandé de confirmer notre compte en cliquant sur le mail.

Your email address is attacker@exploit-0a1200640420964bc48d41fb01dd003a.exploit-server.net

Displaying all emails @exploit-0a1200640420964bc48d41fb01dd003a.exploit-server.net and all subdomains

Sent	From	Subject	Body	
2022-12-13 11:01:41 +0000	no-reply@0ae30055048796bdc4d33fee001d001c.web-security-academy.net	Account registration	<p>Hello!</p> <p>Please follow the link below to confirm your email and complete registration.</p> <p>https://0ae30055048796bdc4d33fee001d001c.web-security-academy.net/register?temp-registration-token=nL0kkI27Hz11DAhyBNFw9C9yFc0vsXw</p> <p>Thanks, Support team</p>	View raw

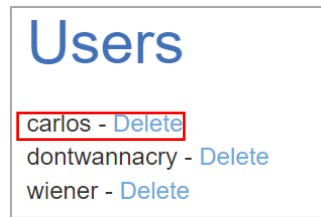
Nous avons suivi les indications données dans le mail. Ce qui a permis de valider notre compte. Nous nous sommes connectés avec ce nouveau compte. Une fois que nous étions connectés le site nous a proposé de mettre à jour notre adresse électronique. Ce que nous avons fait en remplaçant notre ancien email par celui-ci : test@dontwannacry.com, puisqu'il répond au format des mails des personnes travaillant pour l'entreprise du site.

My Account

Your username is: dontwannacry

Your email is: test@dontwannacry.com

Nous avons donc maintenant le nom d'utilisateur des administrateurs ainsi que leur adresse électronique. Une nouvelle page nous était maintenant accessible : « Admin panel ». Cette page permet de supprimer les utilisateurs comme il nous était demandé dans le challenge.



Conseils de corrections

Dans un premier temps il n'est pas conseillé de donner autant d'informations sur les conditions de connexions des administrateurs. Aussi, il serait nécessaire de vérifier qu'il ne soit pas possible de créer des Users avec le même nom plusieurs fois. Enfin, l'email est vérifié à la création du compte mais pas quand on la modifie. Ce qui n'est pas normal.