

1. Les attaques MITM : HTTPS
2. Le phishing : Eduquer les visiteurs
3. Les attaques CSRF : Token CSRF
4. Les modifications du DOM : Faire les vérifications au backend
5. Les injections : Figer la forme de la requete SQL et désactiver les balises HTML
6. Le brute force : Mettre une limite sur le nombre d'essais





Que peut-on vérifier au niveau  
du frontend en toute sécurité ?



Rien !

**RAPPEL**

1. Nous avons vu que si on vérifie les inputs au frontend (par exemple avec `type="email"`), le hacker peut **facilement modifier le DOM** et modifier le `type="email"` en `type="text"` et faire sauter la vérification du navigateur. En effet, ce qui est présenté à l'écran est le DOM et non pas le code HTML directement
2. Nous avons également dans notre TP de Happy Hacking que les vérifications d'identifiant et de mot de passe au frontend pouvait être contournés par le hacker en **regardant le code source de la page**

**CONCLUSION : toutes les vérifications  
au frontend peuvent être contournées  
par un hacker peu expérimenté !**



# Comment éviter le contournement des vérifications ?





**Il faut les faire au niveau  
du backend !**

1. Le code du backend se trouve au **niveau du serveur**

Le serveur ne permet que l'exécution des fichiers PHP mais ne permet en aucun l'accès du code.

En effet, lorsqu'un visiteur entre une URL dans son navigateur, une fonction du controller est exécuté et seulement le résultat de la vue ou des echo parviennent au visiteur.

Ainsi, le visiteur n'a accès qu'au résultat et jamais à l'entièreté du code source

En effectuant les vérifications au niveau du backend, le code est ainsi **inaccessible** au visiteur donc un hacker ne pourra ni le lire, ni le modifier !

Aucun accès pour le hacker en lecture



Si le hacker n'a aucun accès en lecture, que ne peut-il pas faire ?



Le hacker n'ayant pas accès au code sur le serveur, il ne pourra pas obtenir des pistes pour hacker le site internet. Il restera dans le flou et se trouvera dans une situation dans laquelle il devra attaquer une boîte noire.

Aucun accès pour le hacker en écriture





Si le hacker n'a aucun accès en écriture, que ne peut-il pas faire ?



Sans l'accès en écriture, le hacker ne pourra pas modifier les règles de vérifications pour les faire sauter. Par ailleurs, il ne pourra jamais modifier ou détourner le fonctionnement de l'application web (y compris des sites internet).



Si on utilise des frameworks ou des librairies frontend comme Reactjs, Angular ou Vue.js, peut-on faire des vérifications sécurisées ?



Non ! Cela n'a pas de lien avec la technologie utilisée !  
Même si Reactjs, Angular ou Vue.js sont utilisés, cela reste du développement frontend.

C'est donc du code qui sera transporté dans le navigateur et qui y sera exécuté, le hacker y a donc accès en lecture et en écriture !

Tout code relatif à la sécurité ne doit jamais être transporté au navigateur, il doit rester au niveau du serveur, donc il est être développé en backend

