

YAVUZLAR WEB GÜVENLİĞİ & YAZILIM TAKIMI

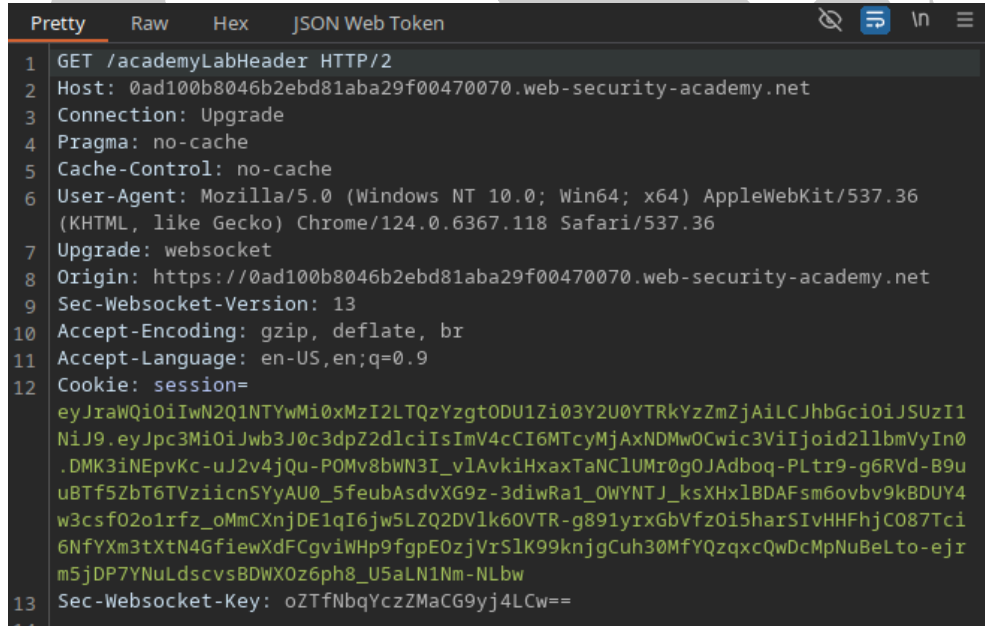
CAPTCHA BYPASS ÖDEV RAPORU

1. JWT Authentication Bypass Via Flawed Signature Verification

Laboratuvarı

Verilen üç adet laboratuvarda da hedef olarak “/admin” sayfasındaki admin paneline erişim sağlayıp “carlos” adlı kullanıcıyı silmemiz isteniyor. Laboratuvar açıklamasında web uygulaması server’ının imzasız JWT’leri kabul ettiği belirtilmiş.

Laboratuvarı başlattığımızda karşımıza bir blog sayfası karşılıyor. Blog sayfasından “my-account” sayfasına geçtiğimizde laboratuvarın başlangıcında verilen kullanıcı bilgileri ile giriş yapıyoruz. Giriş yaptığımız anda gönderilen isteği dinlediğimizde cookie kısmında, oluşturulan JWT’yi görüyoruz.

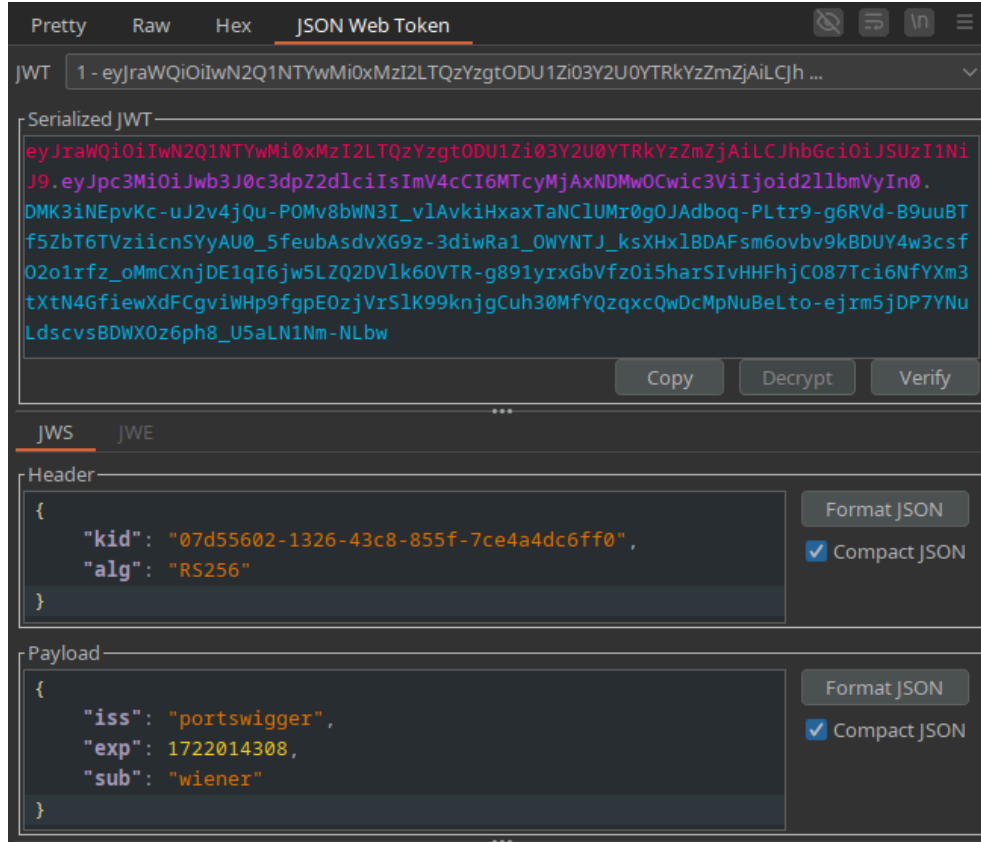


```
1 GET /academyLabHeader HTTP/2
2 Host: 0ad100b8046b2ebd81aba29f00470070.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
7 Upgrade: websocket
8 Origin: https://0ad100b8046b2ebd81aba29f00470070.web-security-academy.net
9 Sec-WebSocket-Version: 13
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: en-US,en;q=0.9
12 Cookie: session=
  eyJraWQ1OiIwN2Q1NTYwMi0xMzI2LTQzYzgtODU1Zi03Y2U0YTRkYzZmZjAiLCJhbGciOiJSUzI1
  NiJ9.eyJpc3MiOiJwb3J0c3dpZ2Z2d1ciIsImV4cCI6MTcyMjYxNjYxNjYxNjYxNjYxNjYxNjYx
  .DMK3iNEpvKc-uJ2v4jQu-POMv8bWN3I_vlAvkiHxaxTaNC1UMr0g0JAdboq-PLtr9-g6RVd-B9u
  uBTf5ZbT6TVziicnSYyAU0_5feubAsdvXG9z-3diwRa1_OWYNTJ_ksXHx1BDAFsm6ovbv9k8DUY4
  w3csf02o1rfz_oMmCXnjDE1qI6jw5LZQ2DV1k6OVTR-g891yrxGbVfz0i5harSiVHHFhjC087Tci
  6NfYXm3tXtN4GfiewXdFCgviWHp9fgpEOzjVrSlK99knjgCuh30MfYQzqxcQwDcMpNuBeLto-ejr
  m5jDP7YNuLdscvsBDWX0z6ph8_U5aLN1Nm-NLbw
13 Sec-WebSocket-Key: oZTfNbqYczZMaCG9yj4LCw==
```

Şekil 1.1. Oluşturulan JWT’nin Ekran Görüntüsü

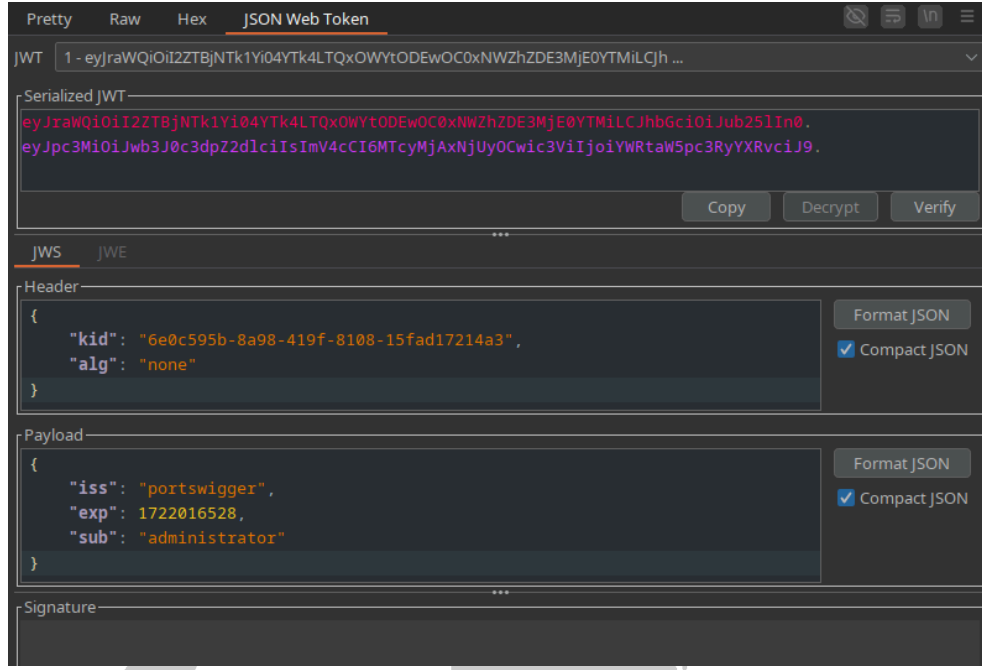
JWT’ler header, payload ve signature olarak üç kısımdan oluşmaktadır. Payload kısmında görüntülediğimiz “sub” parametresi kullanıcıyı temsil eden bir kimliktir. “/admin” sayfasına erişebilmemiz için öncelikle admin olmamız gerekiyor. Bunu da “sub” parametresindeki değeri admin ya da administrator olarak değiştirip JWT’yi kullanarak admin

paneline erişebiliriz. Fakat JWT'lerin yapısı gereği payload kısmında yapacağımız herhangi bir değişiklik JWT'nin bütünlüğünü bozar ve signature geçersiz olur.



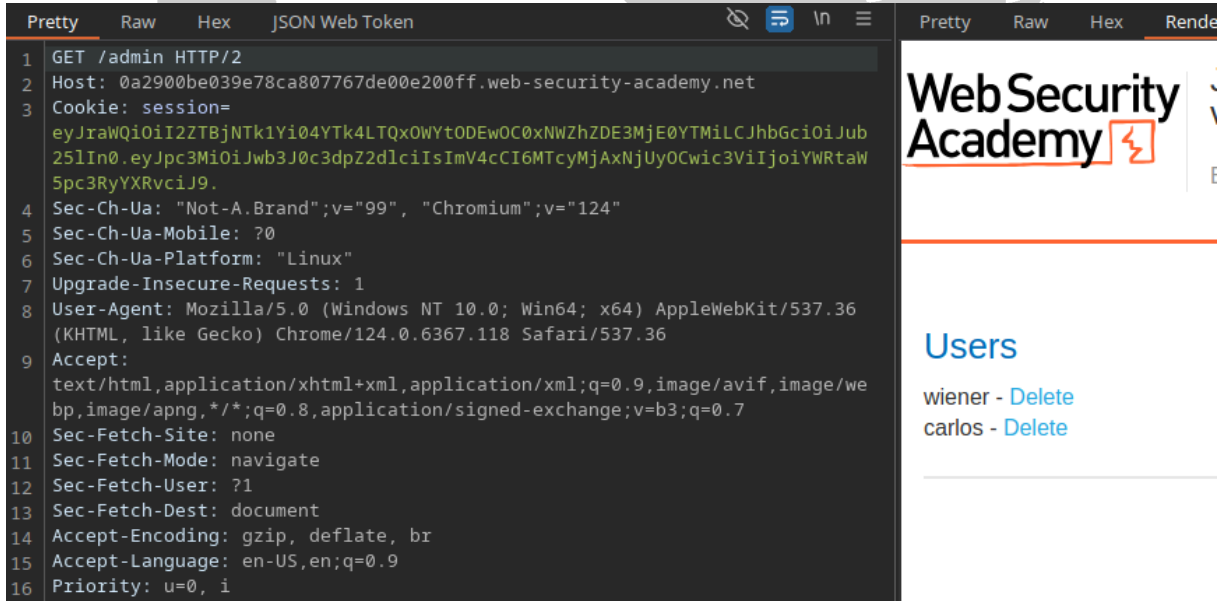
Şekil 1.2. JWT İçeriğinin Ekran Görüntüsü

Signature kısmını oluşturan headerdaki “alg” parametresini “none” olarak değiştirip beraberindeki signature’ı silersek bozuk JWT ile kimlik doğrulamasını aşabiliriz.



Şekil 1.3. Değiştirilen JWT İçeriği

Değiştirdiğimiz JWT ile “/admin” sayfasına gitmeye çalıştığımızda ise sayfaya erişebildiğimizi görüyoruz.



Şekil 1.4. Gönderilen İsteğin ve Sonucunun Ekran Görüntüsü

Admin panelinin kaynak kodlarını incelediğimizde kullanıcı silme işlemi için “/admin/delete?username=carlos” şeklinde başka bir sayfaya yönlendirme yapıldığını görüyoruz. Bu uzantıya mevcut JWT’miz ile istek gönderdiğimizde ise carlos kullanıcısının silindiğini ve laboratuvarı tamamladığımızı görüyoruz.

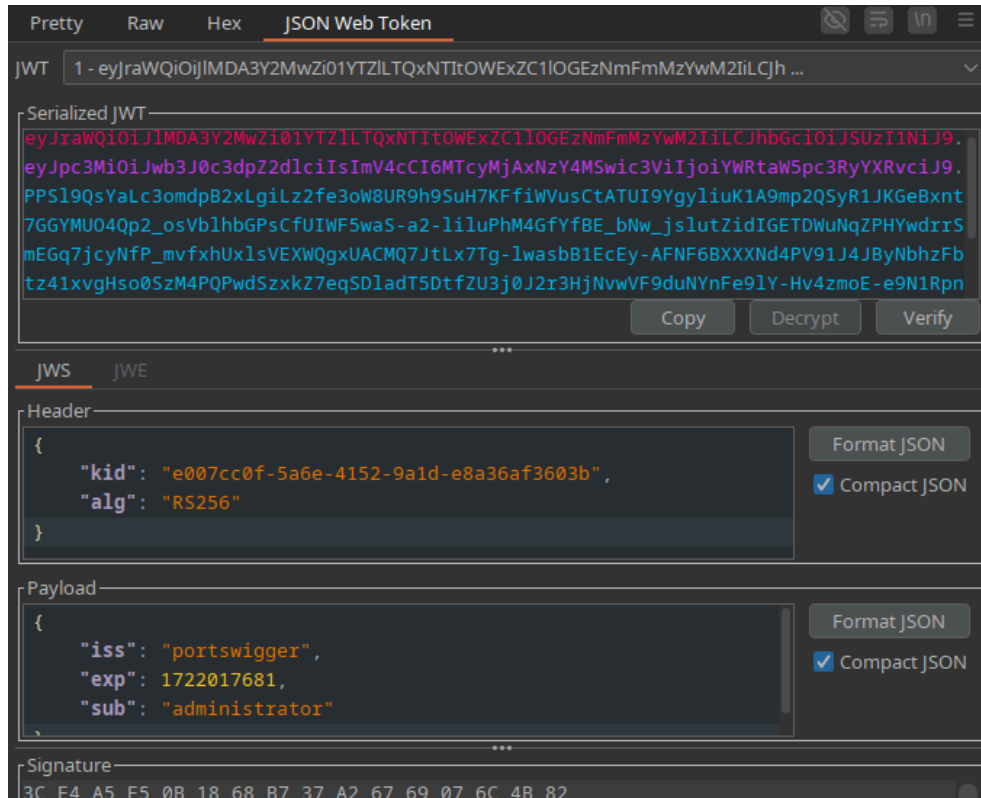


Şekil 1.5. Tamamlanan Laboratuvarın Ekran Görüntüsü

2. JWT Authentication Bypass Via Unverified Signature Laboratuvarı

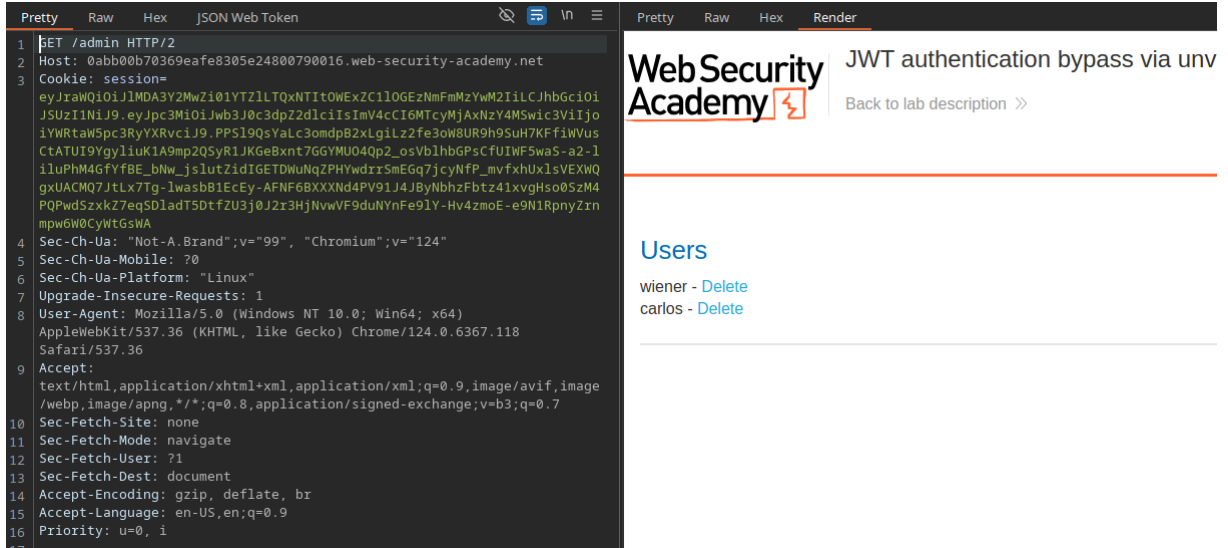
Laboratuvar açıklamasında ipucu olarak web uygulamasının server'ının herhangi bir şekilde JWT'lerin signature'larını kontrol etmediği belirtilmiş.

Önceki laboratuvardaki gibi kullanıcı girişi yapıp JWT üretilmesini sağlıyoruz. Aynı şekilde JWT'nin payload kısmındaki “sub” parametresini administrator olarak değiştiriyoruz.



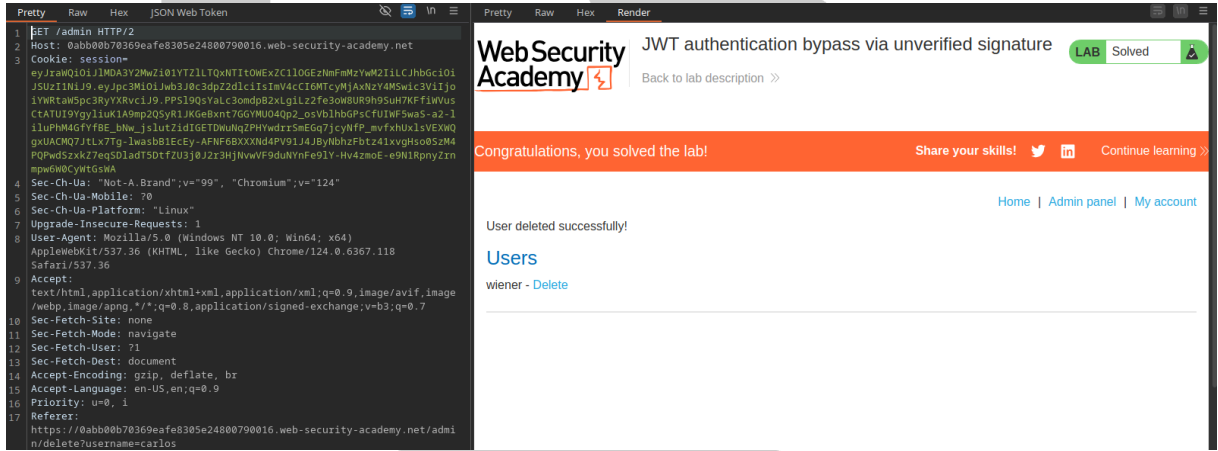
Şekil 2.1. Değiştirilen JWT Ekran Görüntüsü

Değiştirdiğimiz JWT ile “/admin” sayfasına istek attığımızda admin paneline erişimimizin olduğunu görüyoruz. Önceki laboratuvara kıyasla JWT signature'larının kontrolü sağlanmadığı için headerda bulunan “alg” parametresi ve signature üzerinde herhangi bir değişiklik yapmamıza gerek kalmadı. Sadece payload kısmındaki “sub” parametresinde yaptığımız değişiklik ile admin paneline erişim sağladık.



Şekil 2.2. Gönderilen İsteğin ve Sonucunun Ekran Görüntüsü

Kullanıcı silme işlemi için önceki laboratuvar da kullandığımız “/admin/delete?username=carlos” uzantısına mevcut JWT’imiz ile istek attığımızda ise carlos kullanıcıasını silmiş ve laboratuvarı tamamlamış oluyoruz.

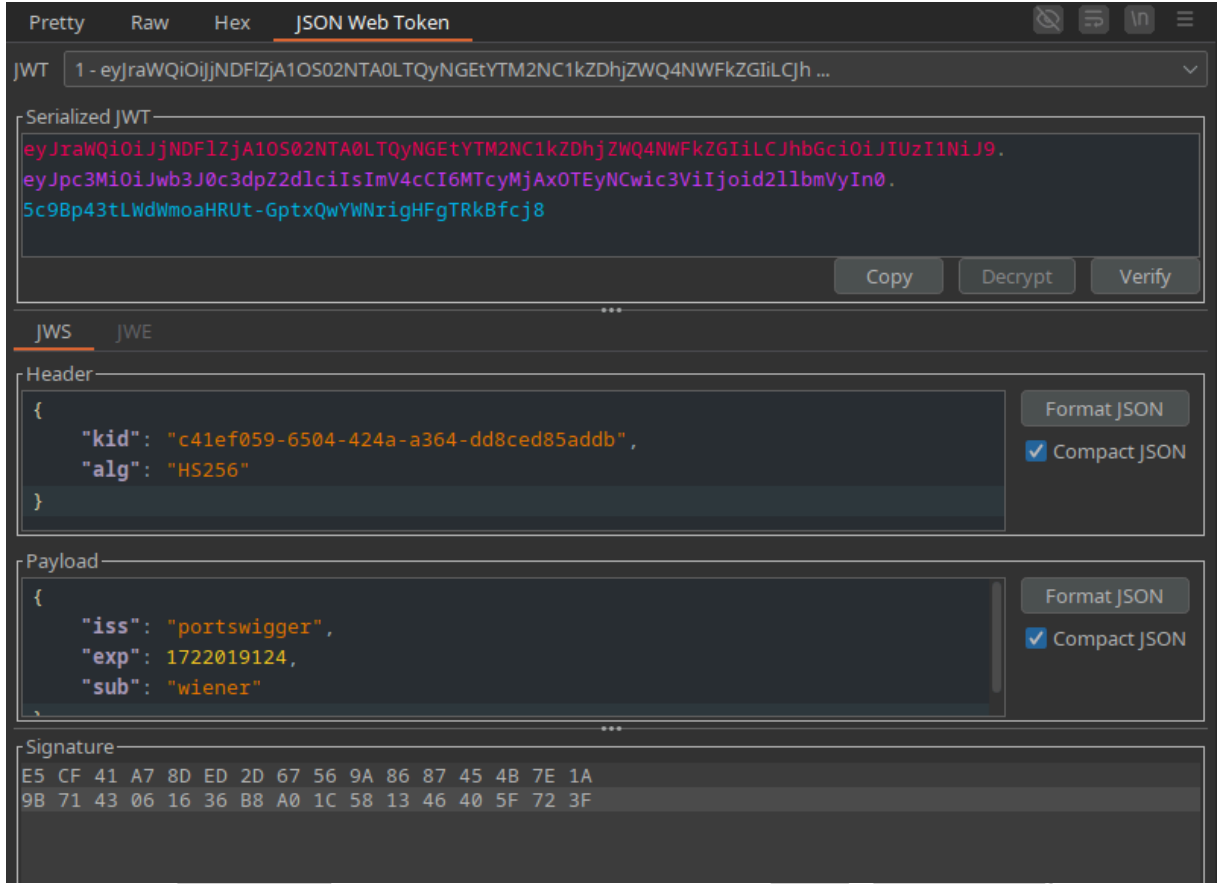


Şekil 2.3. Tamamlanan Laboratuvarın Ekran Görüntüsü

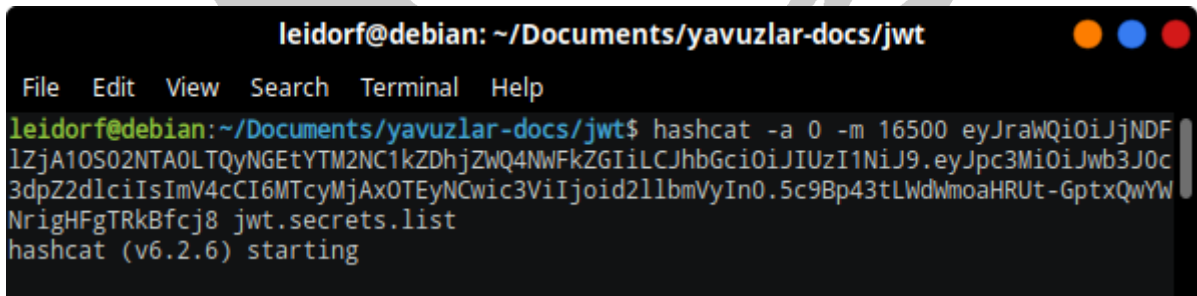
3. JWT Authentication Bypass Via Weak Signing Key Laboratuvarı

Laboratuvar açıklamasında JWT’leri hem imzalamak hem de doğrulamak için son derece zayıf bir anahtar kullanıldığı, bu anahtarın basit bir sözlükle brute force kullanarak ortaya çıkacağı belirtilmiş. Brute force için kullanılacak sözlüğün uzantısı, laboratuvar açıklamasında belirtilmiş.

Laboratuvar da kullanıcı girişi yapıp oluşturulan JWT'yi görüntülediğimizde diğer laboratuvarlardaki JWT'lere göre zayıf bir signature'a sahip olduğunu gözlemliyoruz.



Şekil 3.1. Oluşturulan JWT'nin Ekran Görüntüsü



Şekil 3.2. Hashcat ile JWT Brute Force Kullanımı

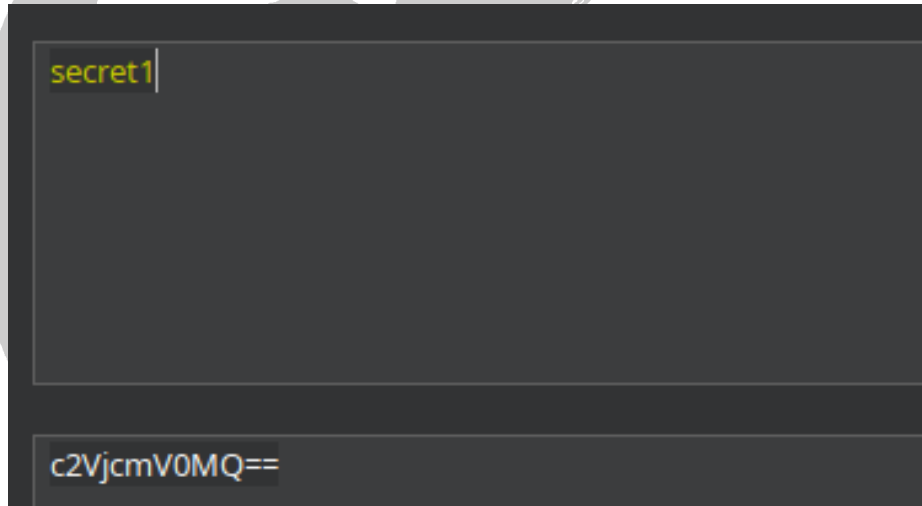
Laboratuvar açıklamasında verilen sözlüğü hashcat aracıyla birlikte kullanarak oluşturulan JWT üzerinde brute force gerçekleştiriyoruz.

```
leidorf@debian: ~/Documents/yavuzlar-docs/jwt
File Edit View Search Terminal Help
eyJraWQiOiJjNDFlZjA1OS02NTA0LTQyNGEtYTM2NC1kZDhjZWQ4NWZkZGIiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJwb3J0c3dpZ2dldciIsImV4cCI6MTcyMjAxOTYyNCwic3ViIjoid2llbmVyIn0.5c9Bp43tLWdWm
oaHRUt-GptxQwYWNrighFgTRkBfcj8:secret1

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 16500 (JWT (JSON Web Token))
Hash.Target.....: eyJraWQiOiJjNDFlZjA1OS02NTA0LTQyNGEtYTM2NC1kZDhjZWQ4NWZkZGIiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJwb3J0c3dpZ2dldciIsImV4cCI6MTcyMjAxOTYyNCwic3ViIjoid2llbmVyIn0.5c9Bp43tLWdWm
oaHRUt-GptxQwYWNrighFgTRkBfcj8
```

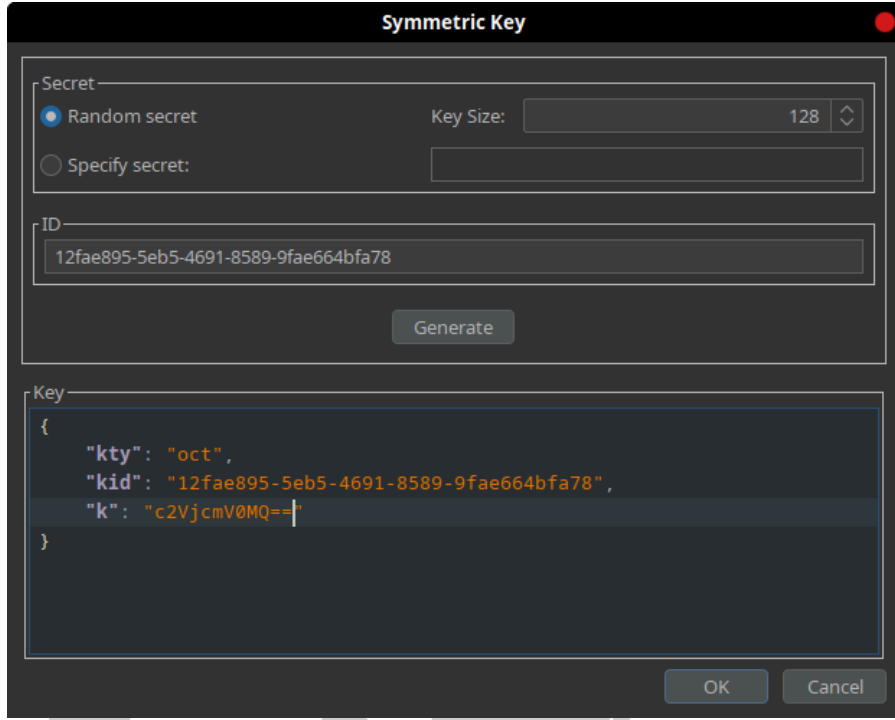
Şekil 3.3. JWT Anahtarının Tespiti

Gerçekleştirdiğimiz brute force sonucu hashcat bize oluşturulan JWT gizli anahtarının “secret1” olduğunu gösterdi. Bu gizli anahtar ile administrator kullanıcısı için JWT oluşturup admin paneline erişiminde kullanıcaz. Bunun için bu anahtarı en başta Base64 algoritması ile şifreliyoruz. Bu şifreleme için Burp Suite’in Decoder sekmesini kullanabiliriz.



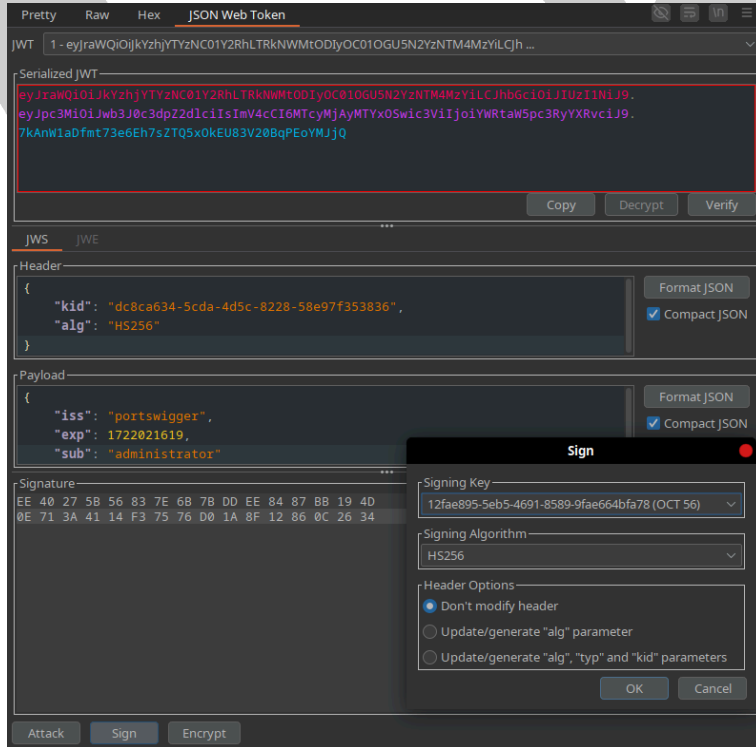
Şekil 3.4. Gizli Anahtarın Base64 ile Şifrenmesi

Şifrelediğimiz değeri yeni bir JWT oluşturmak için kullanıyoruz. Bunun için Burp Suite’in JWT Editor’ü üzerinden boş bir simetrik anahtar oluşturuyoruz. Oluşturduğumuz simetrik anahtarın “k” parametresi gizli anahtarı temsil ettiğinden, şifrelediğimiz değeri buradaki değerle değiştiriyoruz.

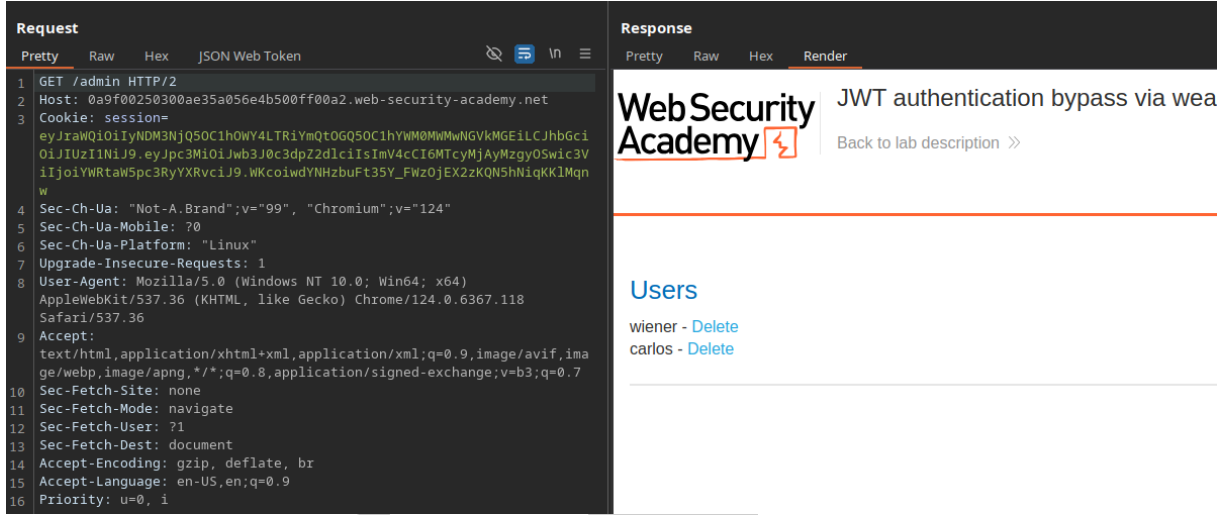


Şekil 3.5. Simetrik Anahtar Oluşturma

Mevcut JWT payloadındaki “sub” parametresindeki değeri administrator olarak değiştirip, oluşturduğumuz simetrik anahtar ile imzalıyoruz ve administrator kullanıcısına ait sahte bir JWT oluşturmuş oluyoruz.

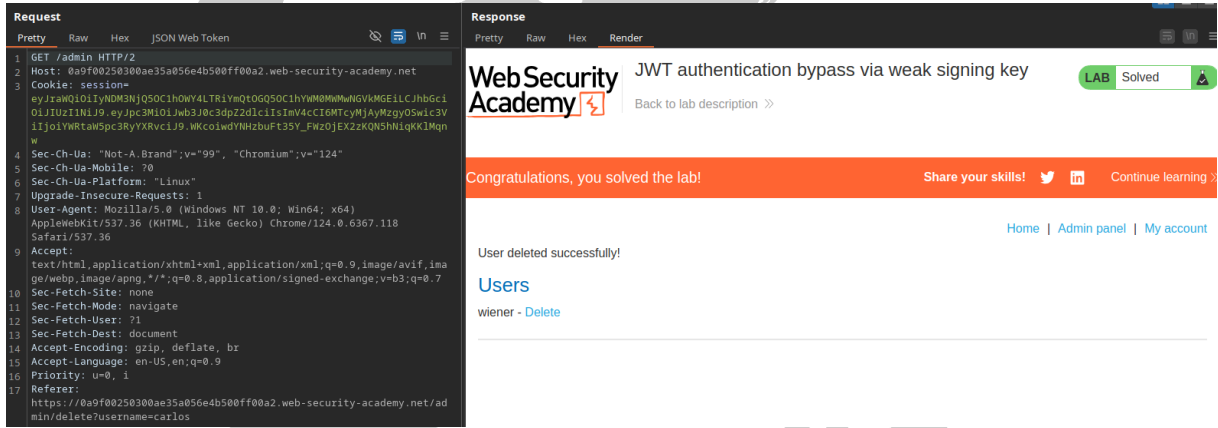


Şekil 3.6. Değiştirilen JWT'nin İmzalanması



Şekil 3.7. Yeni JWT ile Gönderilen İsteğin ve Sonucunun Ekran Görüntüsü

Yeni oluşturduğumuz JWT ile “/admin” sayfasına istek attığımızda erişimimizin olduğunu görüyoruz.



Şekil 3.8. Tamamlanan Laboratuvarın Ekran Görüntüsü

Diğer laboratuvarlarda da kullanıcı silmek için kullandığımız “/admin/delete?username=carlos” uzantısına mevcut JWT’miz ile istek attığımızda carlos kullanıcıasını silmiş ve laboratuvarı tamamlamış oluyoruz.