Burp ile CSRF Token Login Panellere Brute Force Yapma Adımları

a. Arkaplan

Burp ile yapılan brute force ve dictionary saldırıları çeşitli önlemler ile engellenebilmektedir. Bunlar arasında;

i) sleep() v.b. fonksiyonlarla yanlış giriş denemelerine olan yanıtı bekletme
ii) login panele hidden alan olarak sürekli rastgele verilen csrf token değeri eklenmesi
iii) fazla sayıda yanlış giriş denemelerinde istemci public IP'sinin bloklanması
iv) fazla sayıda belirli bir hesabın şifresi yanlış girildiğinde saldırıya hedef olan hesabın bloklanması
v) captcha ile elle müdahale mecburiyeti getirilmesi
vi) login panel tasarımının dinamikleştirilmesi ve iki katmana (username sayfası ve şifre sayfası olarak) bölüştürülmesi (AJAX ile iki katmanlı login panel arayüzü yapılması)
vii) login panele OTP katmanının eklenmesi (yani username ve password katmanlarına ilaveten bir de mobil telefondaki bir uygulamadan veya gelen sms mesajından elde edilen geçici şifrenin girileceği ekstra ekranın eklenmesi)

gelmektedir.

Bu önlemler arasında csrf token önlemi esasında csrf saldırılarında kurbanların korunaksız html form'larını uzaktan saldırgan eliyle fark etmeden submit'lemesin diye oluşturulmuş bir çözümdür. Fakat aynı çözüm, html login form'larına saldırganların doğrudan, fakat brute force ve dictionary saldırıları uygulamak için yapacakları teşebbüslere de engel olmaktadır. Çünkü standard brute force ve dictionary saldırısı yapan araçlar her yapılan http talebine karşılık gelen http yanıtındaki yeni csrf token değerini alıp deneyeceği yeni kullanıcı adı şifre ikilisi paketine eklemedikleri için, yani sadece ilk tanımlanan http talebini yeni kullanıcı adı şifre ikilileri ile tekrarlayıp durdukları için csrf token ile korunan login form'larda işe yaramamaktadır. Bu, html login form'ların güvenliğini sağlayan bir çözümdür. Fakat dinamik metotlar sunan brute force ve dictionary saldırı tool'ları ile bu güvenlik önlemi aşılabilir. Örneğin Burp yazılımı tıpkı diğer araçlar gibi normal koşullarda (normal kullanımda) csrf token ile korunan bir html login sayfasına brute force ve dictionary saldırısı yapamaz. Fakat Burp'ün makro seçeneği devreye alındığında ilave olarak çalıştırılan kurallar sonucu dinamik olarak her gelen yanıttaki yeni csrf token değeri, ardından gönderilecek yeni kullanıcı adı, şifre ikilisinin yer aldığı http talebine yerleştirilip gönderilebilmektedir. Böylelikle sunucu gelen http talep paketlerini geçerli sayıp yetkilendirme kıyaslamalarını olması gerektiği gibi yapıp doğru yanıtları (örneğin şifre yanlışsa yanlış, doğruysa doğru yanıtını) gönderecektir.

Burp normal koşullarda kullanıcı adı ve şifreyi, esas aldığı bir http talep paketine değiştire değiştire koyup tekrarlamaktadır. Yani aynı paketi tekrarlamaktadır ve sadece bu paket içerisindeki kullanıcı adı ve şifre değerlerini statik bir kaynaktan (txt dosyasından veya matematikten) yararlanarak güncellemektedir. Makro ise ekstradan bir kural ilave ederek dinamik bir şekilde her gelen http yanıtında farklı değerde gelen token değerini tekrarlanan paketin token parametresine arguman olarak koyarak paketin bozuk gitmemesini sağlar ve karşı tarafta saldırı sürdürülebilir olur. Kısaca Burp'le yapılan brute force ve dictionary saldırılarına makro ilave ederek korunaklı html formlara halen brute force ve dictionary saldırısı yapabiliriz.

b. [CSRF Token] Korunaklı Login Formuna Brute Force Adımları

1. Burpsuite'i başlat.
   -> java -jar burpsuite_community_v1.7.36.jar
2. Burpsuite'in Intercept sekmesinde Intercept Off yap.
3. Ardından Burpsuite sekmelerden Project options->Sessions sekmesine gel.
   -> Session Handling Rules kısmındaki Add butonuna tıkla.
   -> Ekrana gelen penceredeki Rule Actions kısmında yer alan Add butonuna tıkla ve Run Macro seçeneğini seç.
   -> Ekrana gelen penceredeki Select Macro: kısmına yer alan Add butonuna tıkla
   -> Ekrana gelen Macro Recorder penceresindeki sağ üstte yer alan Intercept Off'u Intercept On yap ve http://localhost./DVWA-master/vulnerabilities/brute/ adresini tarayıcıdan gir.
   -> Macro Recorder penceresindeki ekrana düşen http://localhost. GET /DVWA-master/vul nerabilities/"brute"/ satırını seç ve OK butonuna tıkla.
   -> Daha sonra ekrana gelen Macro Editor penceresinde http://localhost. GET /DVWA-master/vulnerabilities/brute/ satırı seçili haldeyken Configure Item butonuna tıkla.
   -> Ekrana gelen Configure Macro Item penceresindeki Add butonuna tıkla.
   -> Ekrana gelen Define Custom Parameter penceresindeki alt bölümde yer alan paket içeriği görüntüleme kısmı boş ise hemen üzerindeki Refetch response butonuna tıkla. Paket içeriği görünüyorsa veya Refetch response sonrası geldiğinde en alttaki arama kutucuğuna csrf token parametre isminin bir kısmını yaz ve paket içerisinde arat. Örn; "token". Daha sonra token'ın bulunduğu satırdaki token parametre değerini farenle seç. Böylece token parametre değeri burp tarafından paket parse edilerek delimiter'lar ile cımbızlanabilir hale gelecektir. CSRF token değerini farenle seçtiğinde yukarıda Start After Expression ve End At Delimiter bölümleri otomatikmen başlangıç ve son ayraçlarını edinecektir. Eğer daha sağlam bir cımbızlama istersen hemen sağındaki Extract From Regexp group seçeneğine tick atarak burp'ün otomatikmen regexp e göre CSRF token değerini cımbızlamasını sağlayabilirsin. Son olarak en üstteki Parameter name textBox'ına şifre kırma saldırısı sırasında gönderilen paketin içerisinde CSRF token parametresi hangi ismi alıyorsa o konur. Böylece az önce delimiter ile burp'e koyduğumuz kural sayesinde response'lardan gelen csrf token değeri request'teki belirtilen csrf token parametre ismine arguman olarak yerleşecektir. Son olarak OK butonu na tıkla.
   -> Ekrana gelen Configure Macro Item pencersine de OK de.
   -> Ekrana gele Macro Editor penceresine de OK de.
   -> Ekrana gelen Session Handling Action Editor sayfasındaki ortalarda yer alan Tolerate URL Mismatch When Matching Parameters (use for URL-agnostic CRSF tokens) seçeneğine tick at ve OK de.
   -> Ekrana gelen Session Handling Rule Editor penceresindeki Scope sekmesine gel. Tools Scope kısmındaki Intruder seçeneğinin tick'li olduğundan emin ol. Hemen aşağısındaki URL Scope kısmındaki Use suite scope [defined in Target tab] seçeneğinin tick'li olduğundan emin ol. Son olarak OK butonuna tıkla.
   -> Proxy -> Intercept sekmesine gel ve Intercept is On yap.
   -> http://localhost/.DVWA-master/vulnerabilities/brute/ adresindeki login panele rastgele bilgiler gir ve Burp Proxy->Intercept sekmesine paket geldiğinde paketi sağ tık Intruder'a gönder.
   -> Intruder->Positions sekmesinde paketteki $'ları sil ve şifreyi Add $ ile işaretle.
   -> Intruder->Payloads sekmesinde Payload Options kısmından bir wordlist ekle.
   -> Intruder->Options sekmesinde Grep-Match kısmından Add butonu ile login başarılı olduğunda gelen "Welcome to the password protected area" mesajını ekle. Eklediğin bu ifadenin hemen üzerindeki "Flag result items ….'ın tick'li olduğundan emin ol".
   -> Saldırı öncesi son olarak Target->Sitemap sekmesi altındaki http://localhost. 'a sağ tık ya pıp Add to Scope seçeneğine tıkla. Eklendiğinden emin olmak için ise Target->Scope sek

mesinde http://localhost. satırı eklenmiş mi teyit edebilirsin.
-> Ve Intruder->Options->Start Attack ile saldırıyı başlat.

[!] OLASI HATALAR ve ÇÖZÜMLERİ

i) Macro Recorder Penceresinde Remove From Scope Yapma Hk.

Macro Recorder penceresinde sadece macro'nun çalıştırılacağı login sayfası url'sini barındıran satır seçilir ve o seçiliyken pencere aşağısındaki OK butonuna tıklanır. Diğer satırlar (örn; macro'nun çalışltırılacağı login sayfası url'si, ama misal GET ile parametreleri eklenmiş hali veya başka başka adresler) seçilmemelidir. Diğer türm adresleri Delete Selected Items ile silebilirsin.

Olay:

Örneğin Macro Recorder penceresinde sıralı satırlardan http://localhost./DVWA-master/vulner abilities/brute/ satırını seçecektin. Ama diğer satırların kalabalıklığı dolayısıyla onları temizlemek istedin. Bunun için örneğin http://localhost./DVWA-master/ satırını Remove From Scope dedin. Bunu demen sıkıntı doğuracaktır. Çünkü saldırıya başlamadan önceki Session Handling Rule Editor penceresinde Scope sekmesinde yer alan URL Scope kısmına "Use suite scope [defined in Target tab]" demiştin. Yani Target->Scope'da ekli duran adreslere sadece Makro'yu uygula demiş oldun. Fakat saldırı öncesi son maddede bir yandan Target->Sitemap de http://localhost./ 'u Add to Scope diyerek scope'a ekleyerek brute/ dizinini de kapsarken, bir yandan daha önceki maddelerde yaptığın http://localhost./DVWA-master/ adresini Remove From Scope ile scope dışı bıraktın. Yani böylece localhost scope'da oldu, ama DVWA uygulaması scope dışı oldu. Bu nedenle Macro Recorder penceresinde Delete'i kullanman daha güvenli duruyor.

c. Korunaklı Login Formuna Brute Force Uygulaması

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

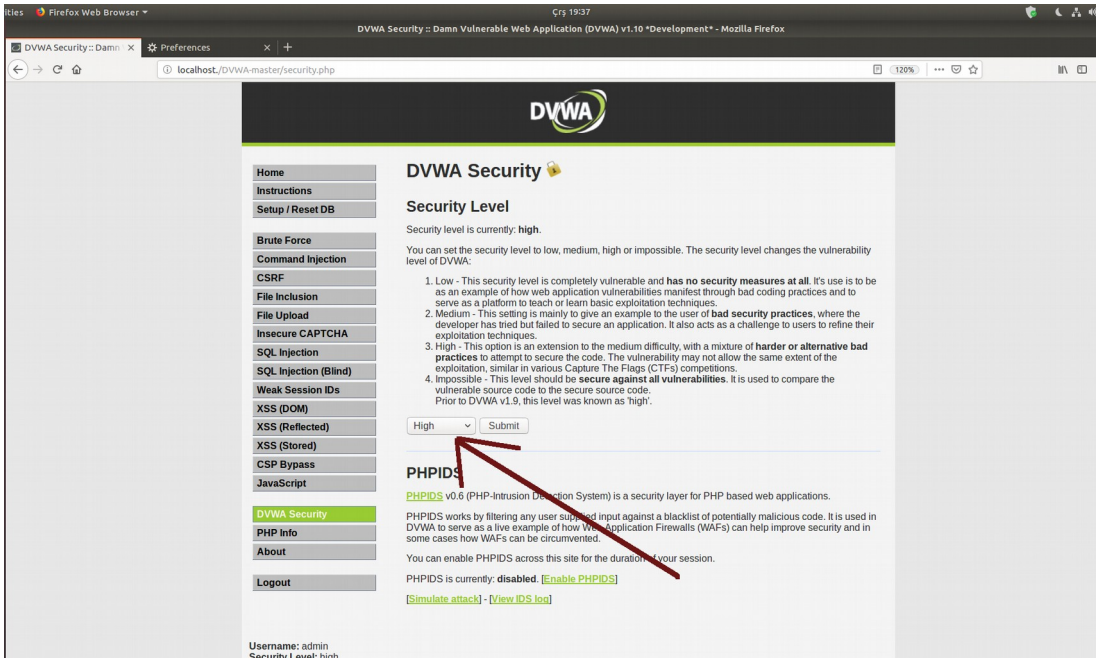
Gereksinimler

Ubuntu 18.04 LTS	// Ana Makina
DVWA-master/	// Yerel Apache Sunucudaki Dvwa Son Sürümü
Burpsuite	// Ana Makina'daki Burpsuite Aracı

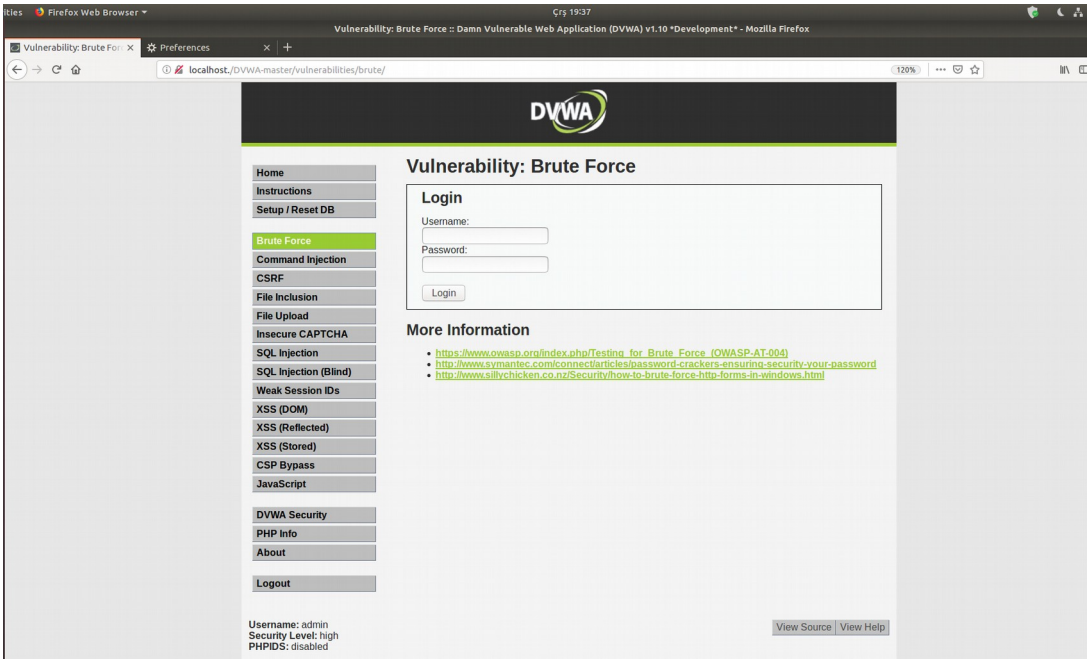
DVWA'nın high seviyesindeyken Brute Force ekranına şifre kırma saldırısı düzenleyebilmek için csrf token korunağını aşmamız gerekmektedir. Bunun için tarayıcıdan dvwa'ya gidelim.

http://localhost./DVWA-master/	// Not: localhost'un sonunda nokta var.
	// Diğer türlü güncel linux firefox'larda
	// burp localhost trafiğini kesememekte.

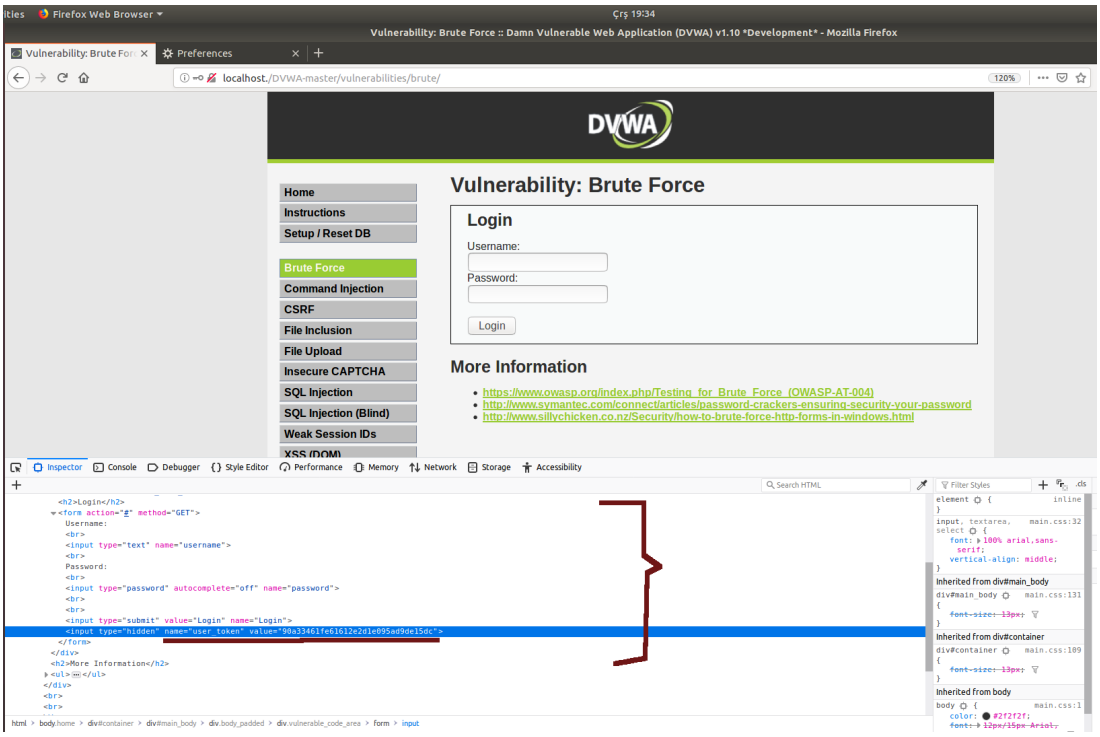
Ardından security sayfasından Low'u High yapalım.



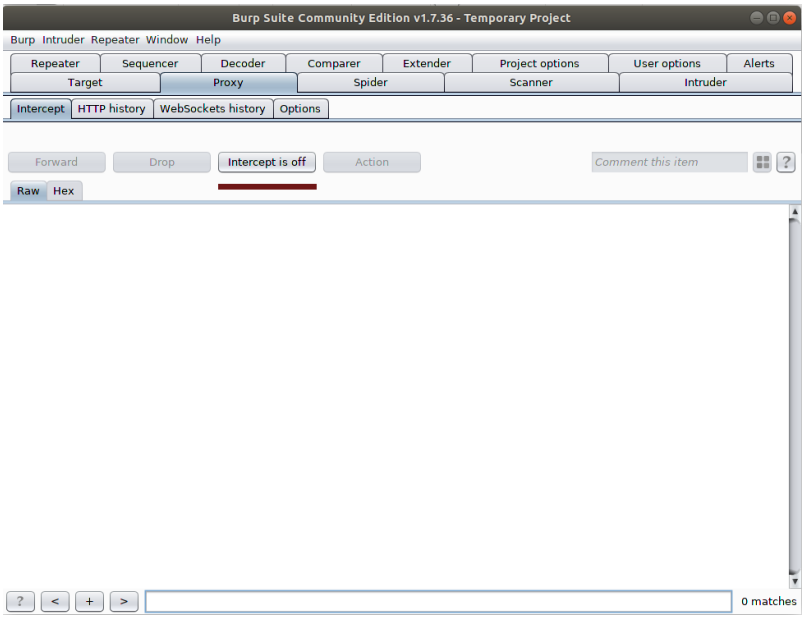
Ve dvwa brute force ekranına geçelim.



Dvwa high seviyesindeki brute force dersinde yer alan login form'una bakıldığında csrf token görülecektir.

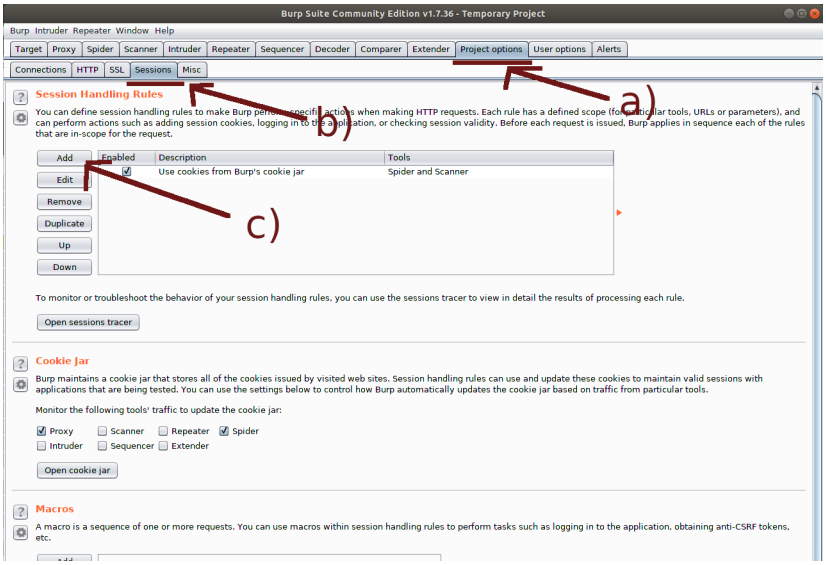


Ve Burpsuite "Intercept Off" olarak kalsın.

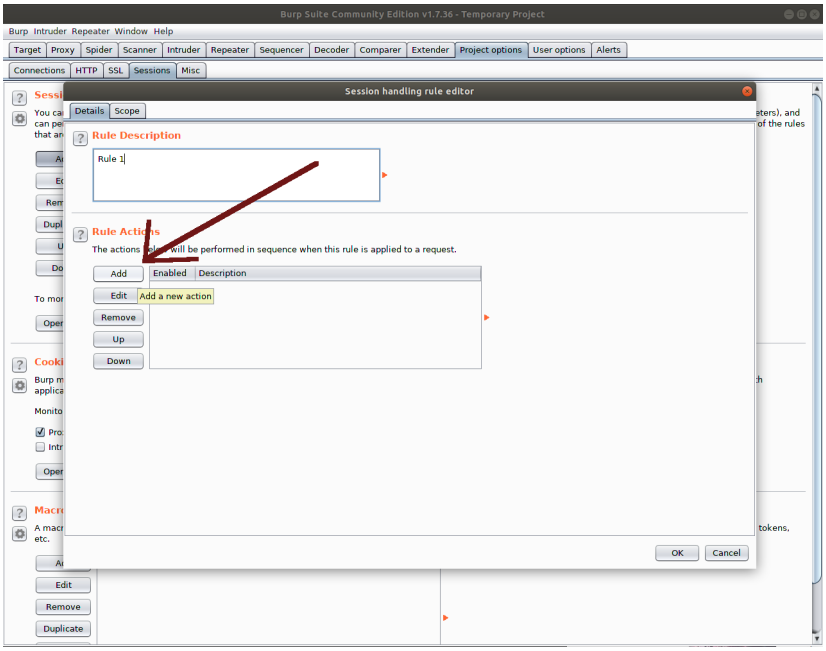


Öncelikle sözlük saldırısı düzenleyeceğimiz login form'undaki korunağı sağlayan csrf token hidden alanı için gerekli makro tanımlamasını yapalım.

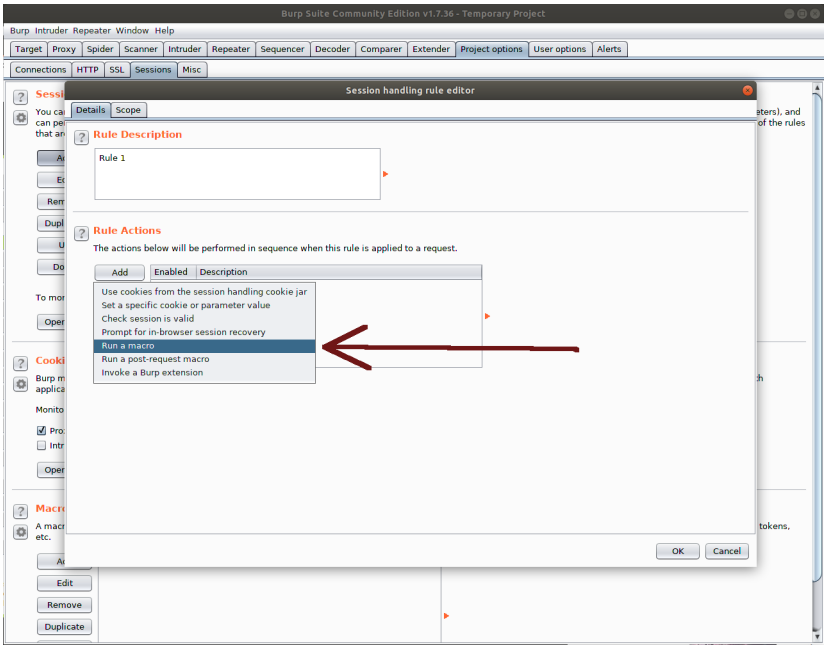
Burp'te Project Options -> Sessions sekmelerinden Session Handling Rules'a gidip Add butonuna tıklayalım.



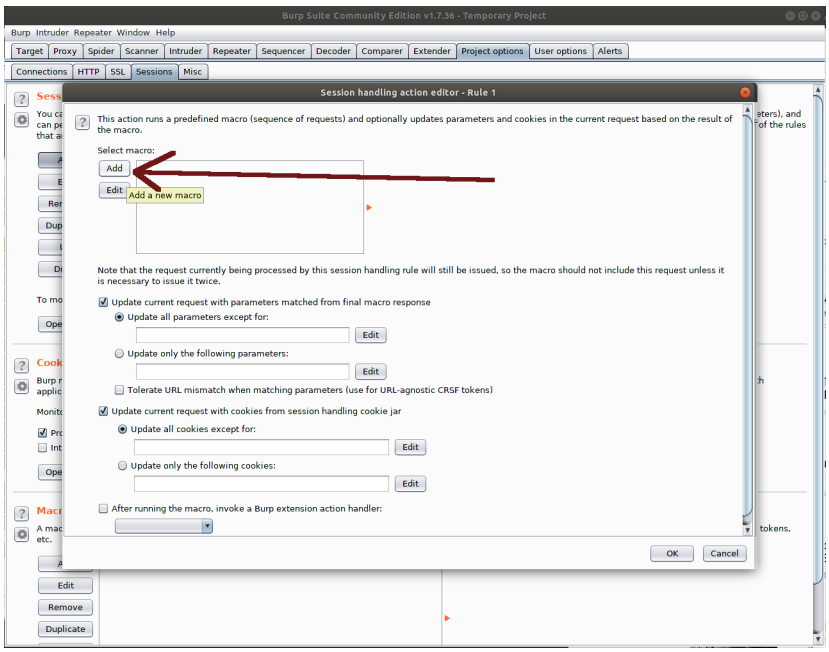
Gelen ekrandaki Rules Actions kısmında yer alan Add butonuna tıklayalım.



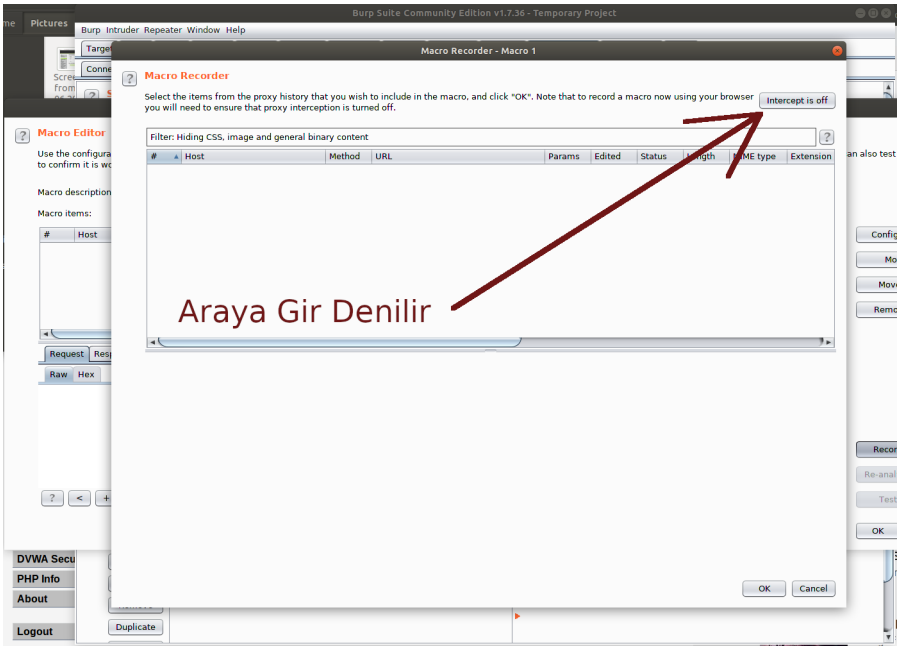
Run a Macro seçeneğine gidelim.



Gelen Session Handling Action Editor ekranındaki Select Macro kısmında yer alan Add butonuna tıklayalım.



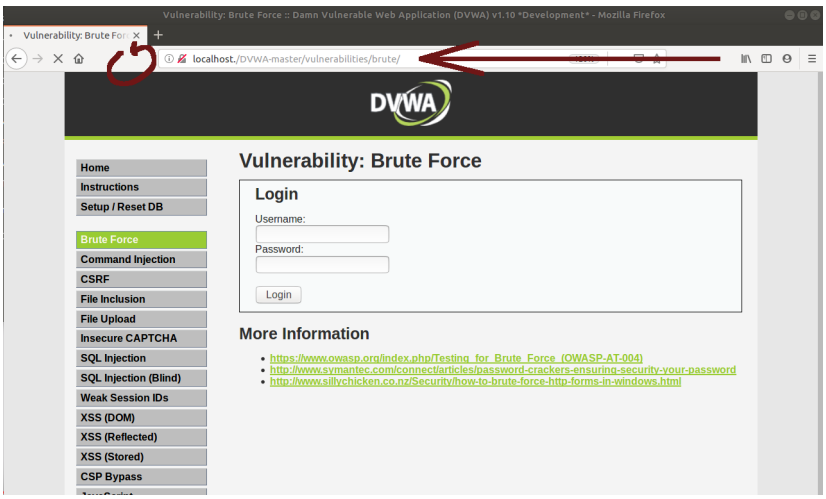
Ekrana gelen Macro Recorder sayfasındaki Intercept is off ‘u On yapalım.



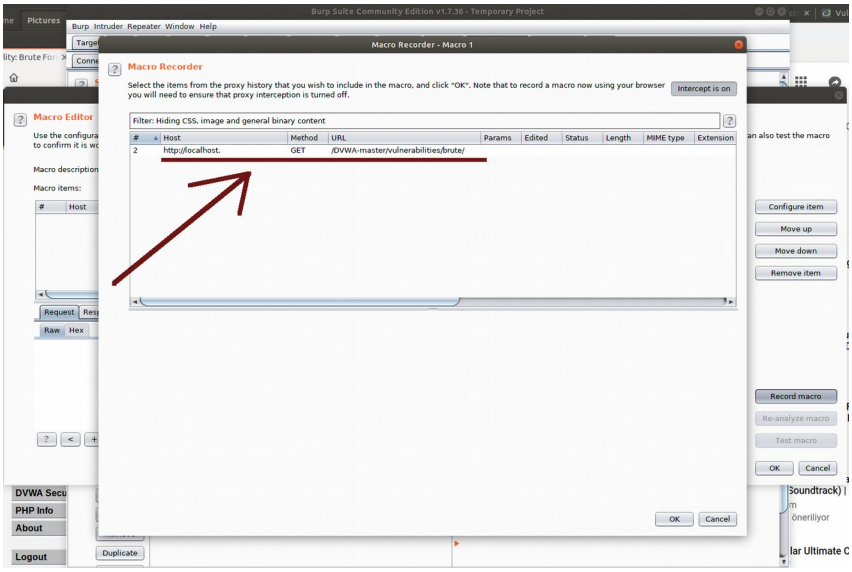
Ardından csrf token korunaklı login sayfasını refresh’leyelim.

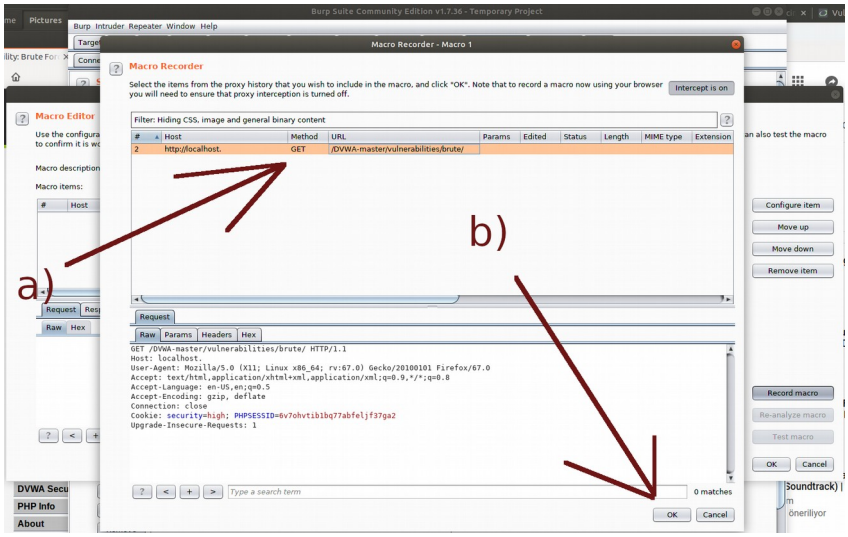
[!] Uyarı:

Yapılması gereken şey csrf token korunaklı login form’unu submit’leme değil, o form’un bulunduğu sayfayı refreshlemektir. Bu şekilde sunucudan gelen yanıttaki csrf token değerini tutan alanın name attribute değerini ve value attribute değerini alarak üzerinde işlemler yürütebileceğiz (not: Diğer türlü dvwa sapıtıyor).

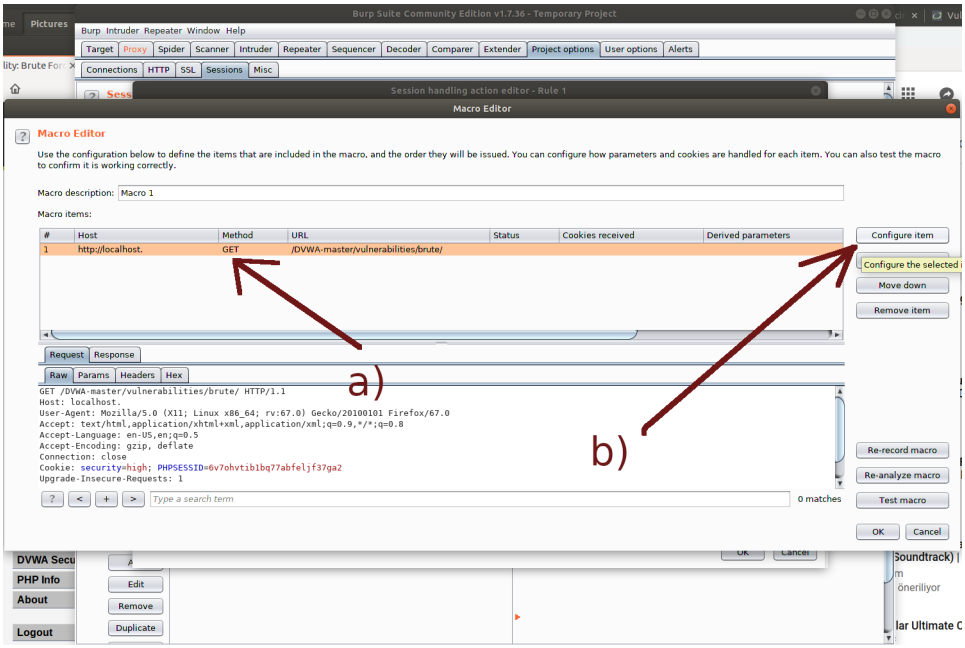


Mevcut Macro Recorder penceresinde ekrana düşen refresh'lenmiş sayfa seçilir ve OK butonuna basılır.

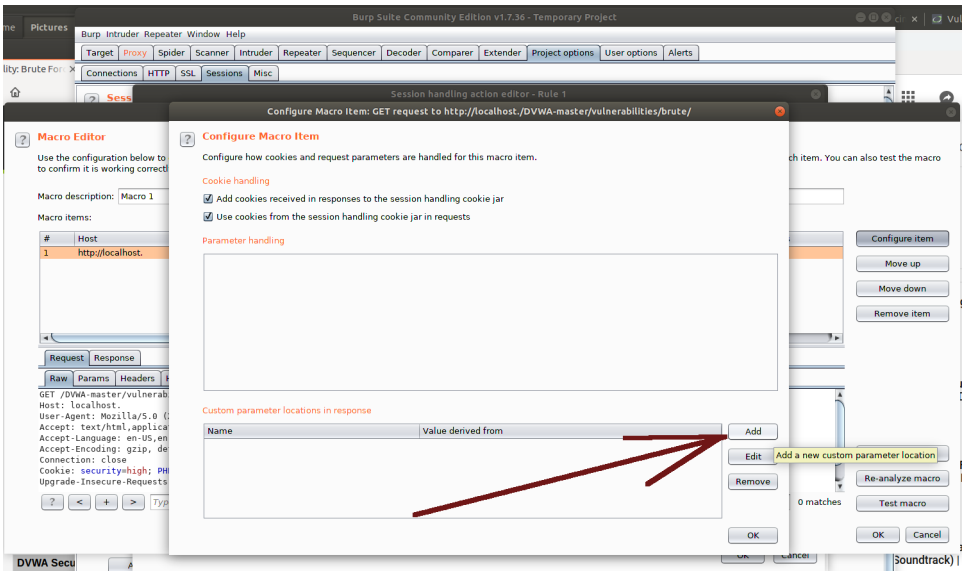




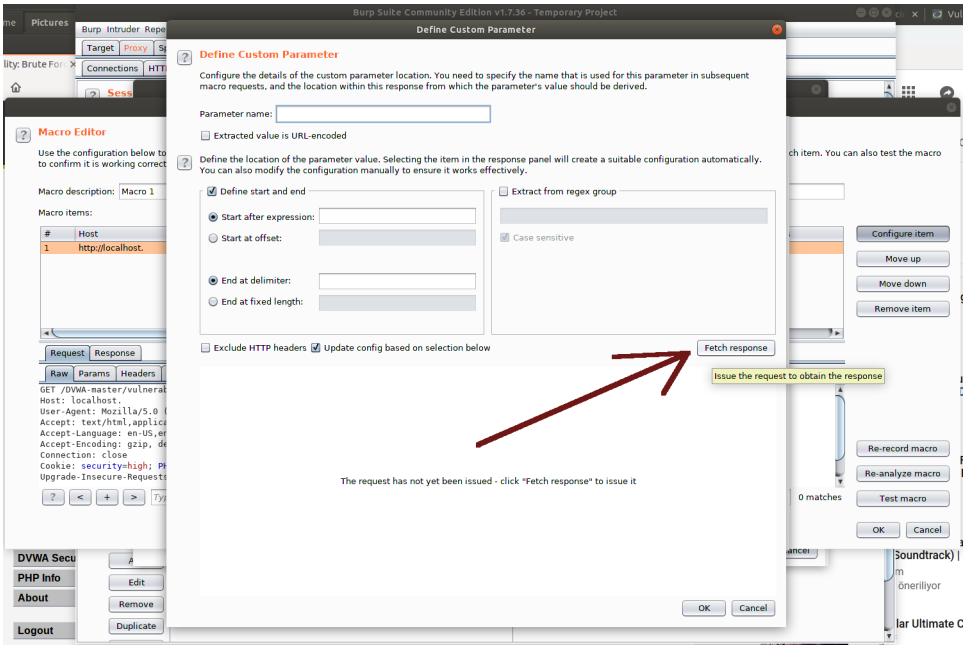
Gelen Macro Editor ekranında daha aynı http talebi tekrar seçilir ve Configure item butonuna tıklanarak kural tanımlama sayfasına gidilir.

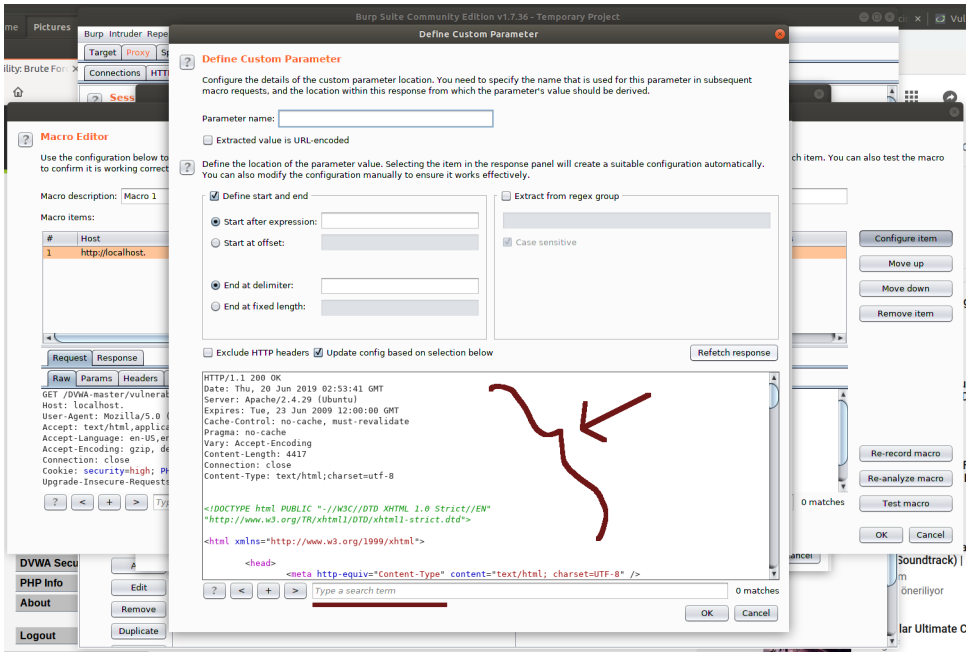


Gelen ekrandaki “Custom Parameter….” (yani kişiselleştireceğimiz bir parametre…..) bölümünde yer alan Add butonuna tıklanır. Bu kısım bizi sunucudan dönen csrf token değerini tutan hidden alanın üzerine yapılacak kuralları girebileceğimiz ekrana götürecektir.

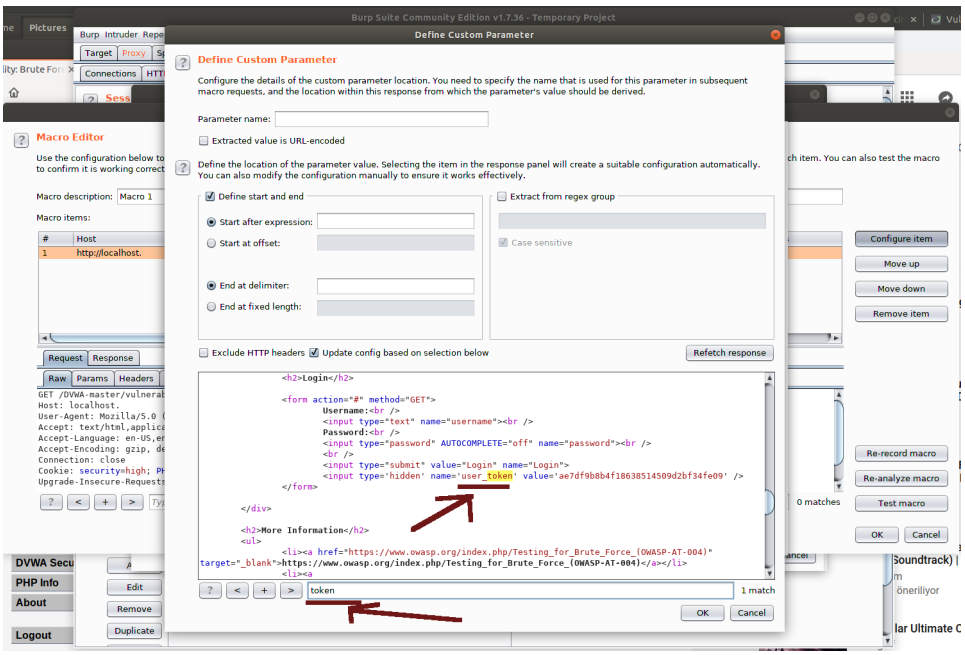


Gelen ekranda seçtiğimiz http talebinin yanıt paket içeriği yoksa Fetch Response butonuna tıklanır.



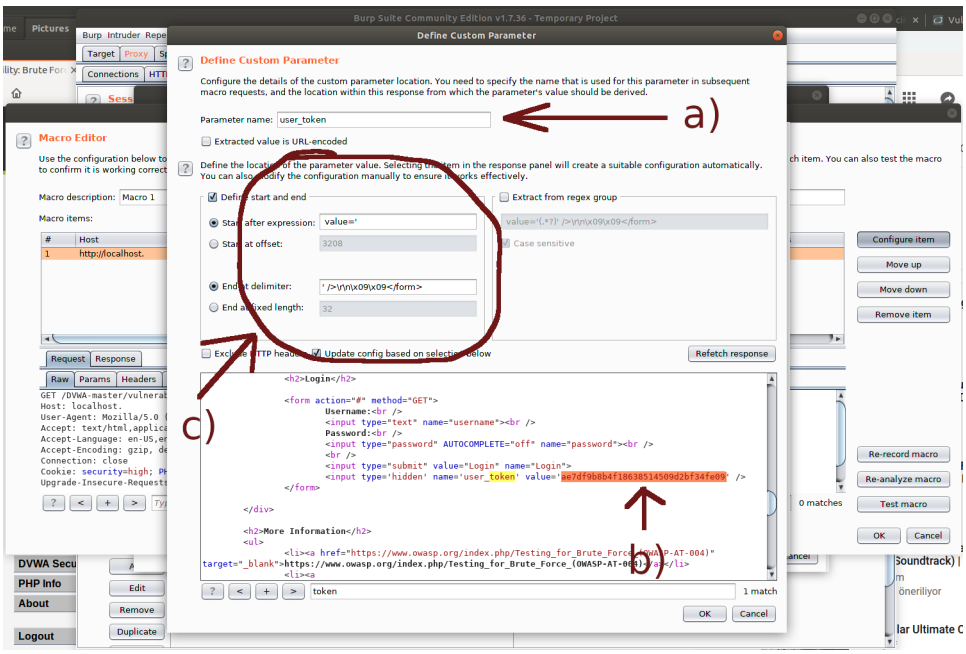


Gelen http yanıtında csrf token olan html düğümünü bulmak için arama kutucuğuna token v.b. ifade girilir.



Csrf token değerini tutan alan bulunduktan sonra http talep paketinde kişiselleştireceğimiz parametrenin ismini alıp ekrandaki Parametre Name textBox'ına koyalım. Bu, http taleplerinde hangi parametre'nin işaretleneceğini ve makronun üzerinde çalıştırılacağı bilgisini sağlar (not: Bir nevi http talep paketindeki csrf token'ı $ 'lama).

Sonra ekrandaki http yanıt paketi üzerinden csrf token değerini komple fare ile seçelim. Burp'te tanımlamakta olduğumuz makro, burp'ün sözlük saldırısı esnasında http talep paket gönderimlerinde karşılık olarak alacağı http yanıt paketlerinde nereye bakacağı bilgisini görmüş olacaktır. Bu sayede her gelen http yanıt paketlerindeki belirtilen bölgeyi alıp sonraki http talebinde paketteki işaretlenen parametreye ekleyerek http talep paketini gönderecektir.

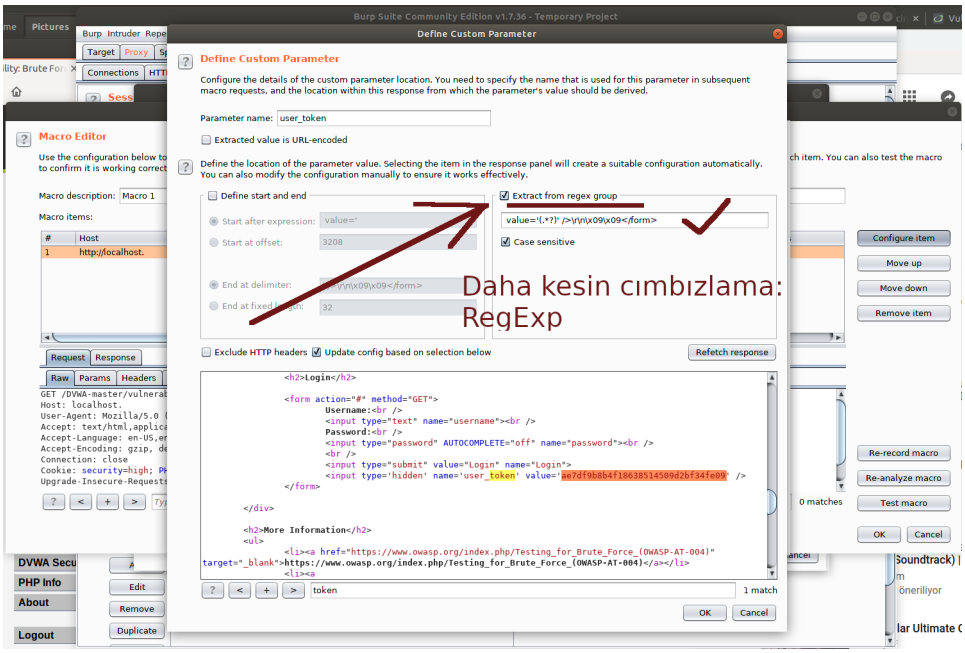


Burp makrosu artık gönderilen http talebine karşılık gelen http yanıtındaki belirli bölgeden gelen değeri http talebindeki işaretlenen parametreye ekleme kuralına sahiptir.

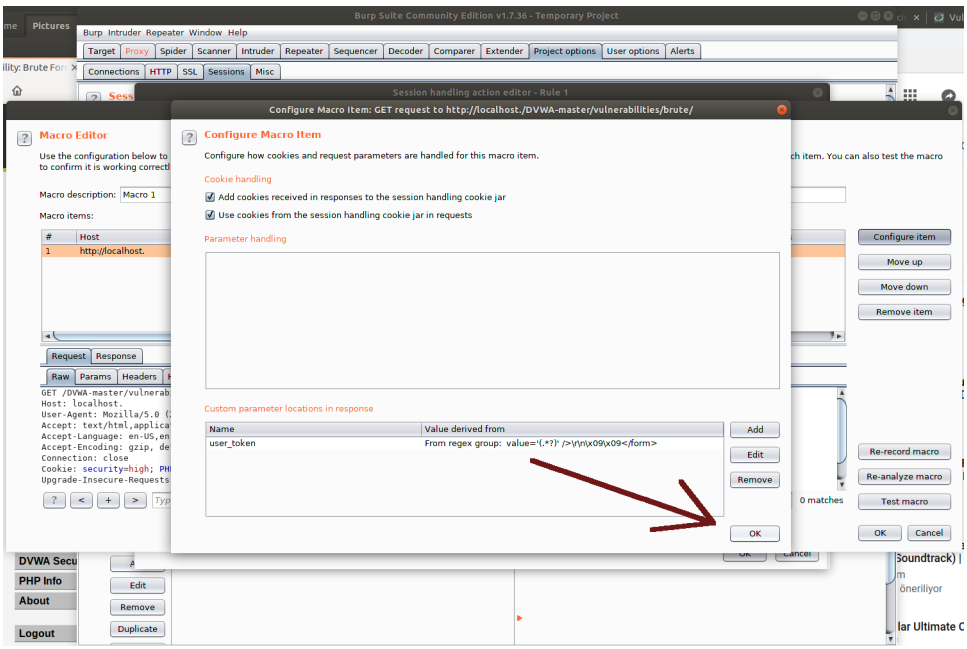
[*] Bilgi:

Yukarıdaki makro kural girme penceresinde “Parameter Name” metin kutusuna http yanıtında yer alan user_token parametre isminin girilmesi işlemi yapılmamaktadır. Şayet öyle olsaydı parametre değerini çekecek kural oluşturma (fareyle seç ve bölge koordinatını göster) işlemine gerek kalmazdı. “Parameter Name” metin kutusuna http taleplerindeki bir parametrenin ismi girilir. Alt tarafta ise bu parametrenin, değerini nereden alacağı kuralı girilir.

Burada değinilebilecek bir başka nokta daha vardır: Makro kural girme penceresinde gelen http yanıtlarındaki token field’ının değerini çekme kuralı fare hareketlerimize göre oluştuğundan gerçekten bölge seçme kuralı doğru mu oluşmuş net görebilmek için aynı ekranın sağ tarafındaki Extract From Regexp Group seçeneğine tick atılabilir. Böylece burp, fare seçim hareketimize göre oluşturduğu otomatik kuralı bu sefer Regexp Pattern karşılığı olarak (yani daha okunur bir formatta) ekrana verecektir ve eğer pattern’ı hatalı bulursak elimizle düzeltmeye gidebileceğiz.

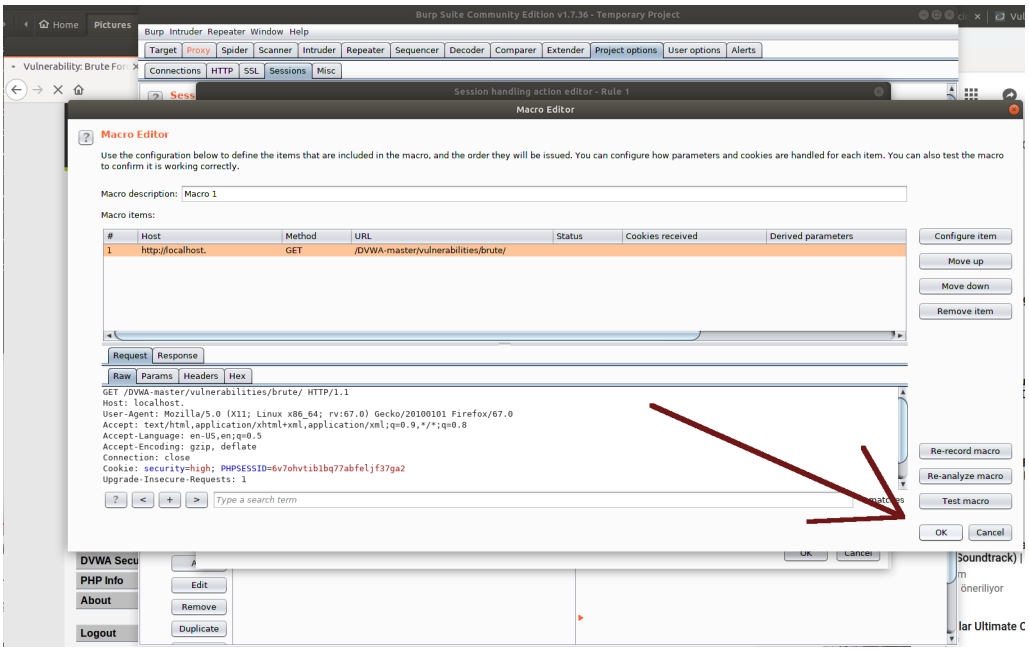


Ardından OK butonuna tıklanır. Ekrana bir önceki ekran “Configure Macro Item” ekranı gelir.

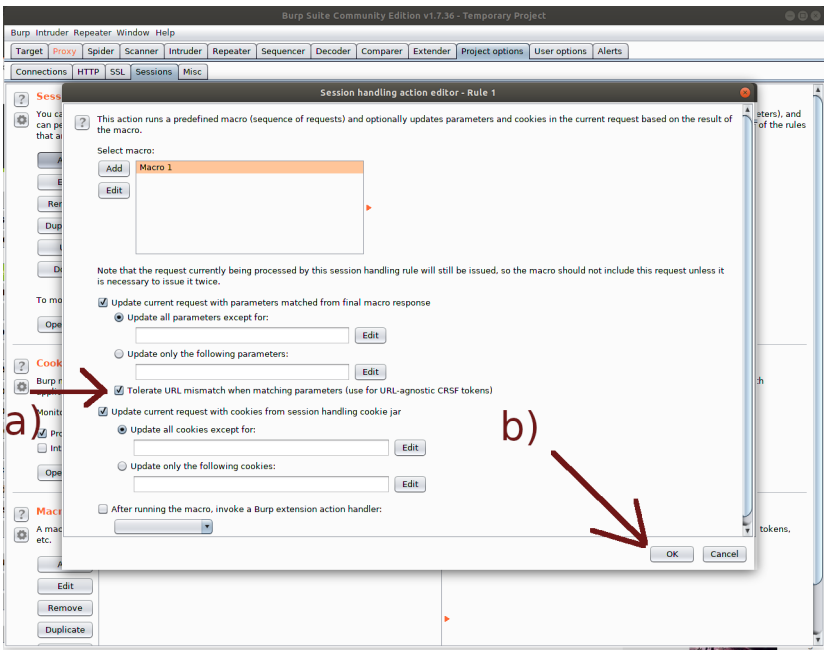


Bu ekranda görülebileceği üzere kişiselleştireceğimiz parametrenin ismi (yani http taleplerinde gönderilen bir parametrenin ismi), o parametrenin değerinin de nereden türetileceği bilgisi (yani http response'dan gelen bir değerden türetileceği bilgisi) yer alacaktır. Bu, bizim login form'una submit denemesi yaptığımızda yanlış deneme sonrası karşı tarafın http response'u içerisinde gelen csrf token random değerini alma kuralımızdır.

Bu ekranı OK diyerek geçelim. Ardından gelen ekrana da OK diyelim.



Sonra, ekranda yer alan Tolerate URL mismatch when matching parameters (use for URL-agnostic CSRF token) seçeneğine tick atılır ve OK butonuna basılır.

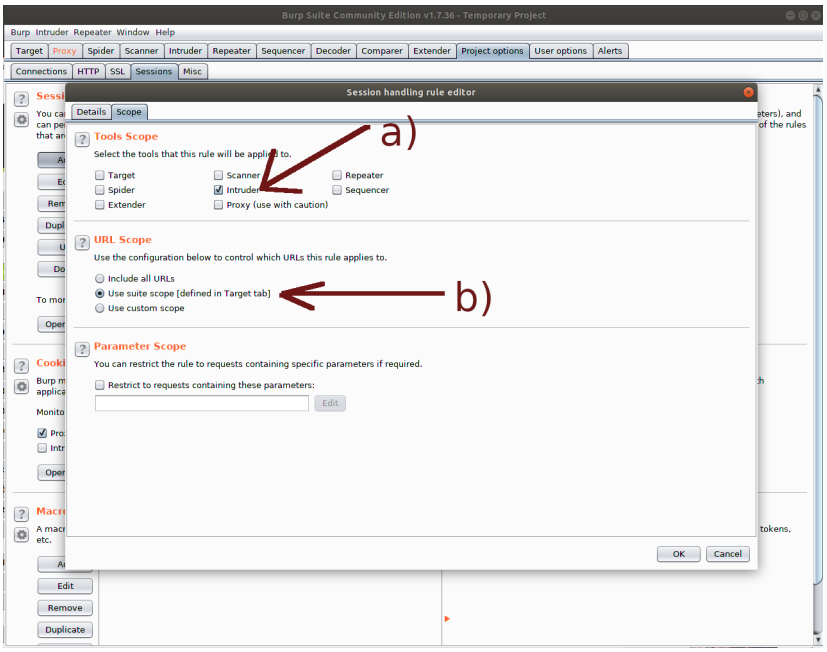


Ve son olarak gelen bir önceki ekranda yer alan Scope sekmesine geçilir. Bu sekme altında burp aksiyonu olarak sadece Intruder'u seçelim. Böylece oluşturduğumuz makronun burp'ün sadece Intruder sekmesi altında etkin olmasını sağlarız. Diğer sekmelerinde belki aynı anda farklı işlemler yürütüyor olabiliriz ve bu makronun araya girip çorba olmasını istemeyebiliriz.

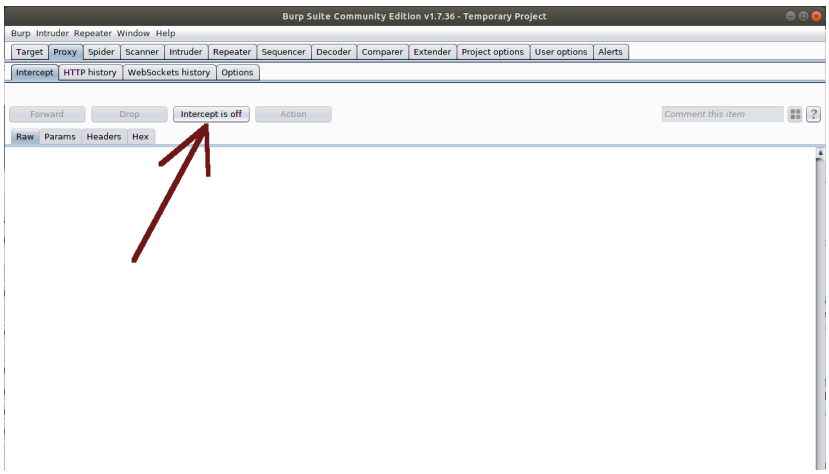
Aynı ekrandaki URL Scope kısmına ise Use suite scope diyerek daha sonra tanımlayacağımız target url scope kısmındaki url bilgisi dışında bir url eğer işleme tabi tutuluyorsa ona makroyu uygulama demiş oluruz.

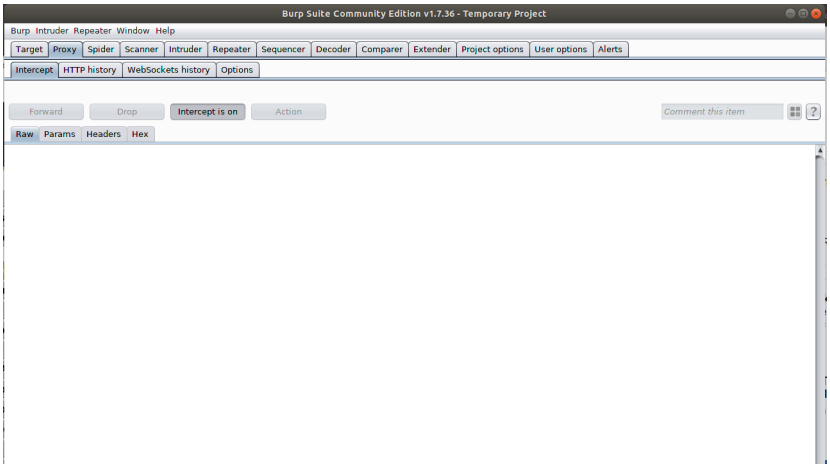
Böylelikle makro Burp'ün sadece Intruder sekmesi altında ve bu sekme altında da sadece belirli url'e yapılan işlemlerde kullanılabilir olacaktır.

Bu işlemler sonrası ekrana OK diyelim.

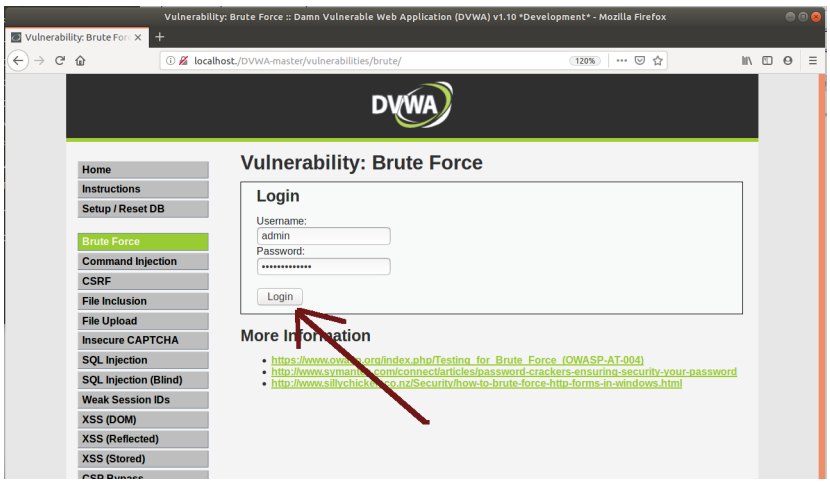


Burp proxy’sini Intercept On yapalım.

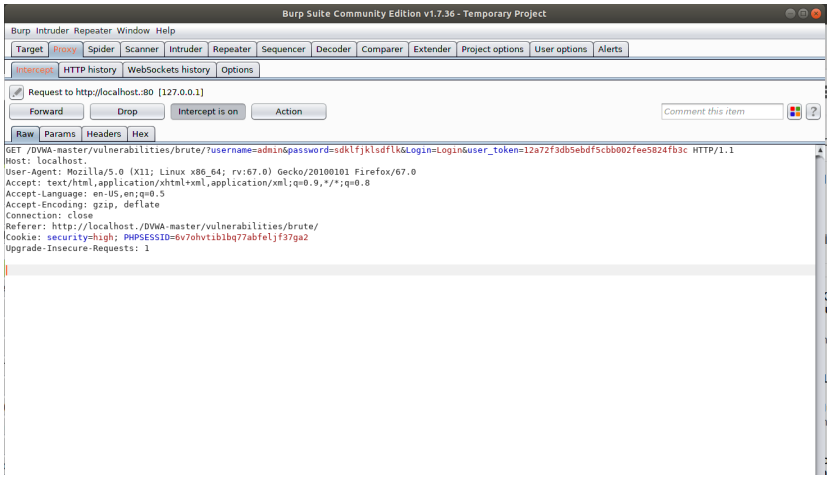




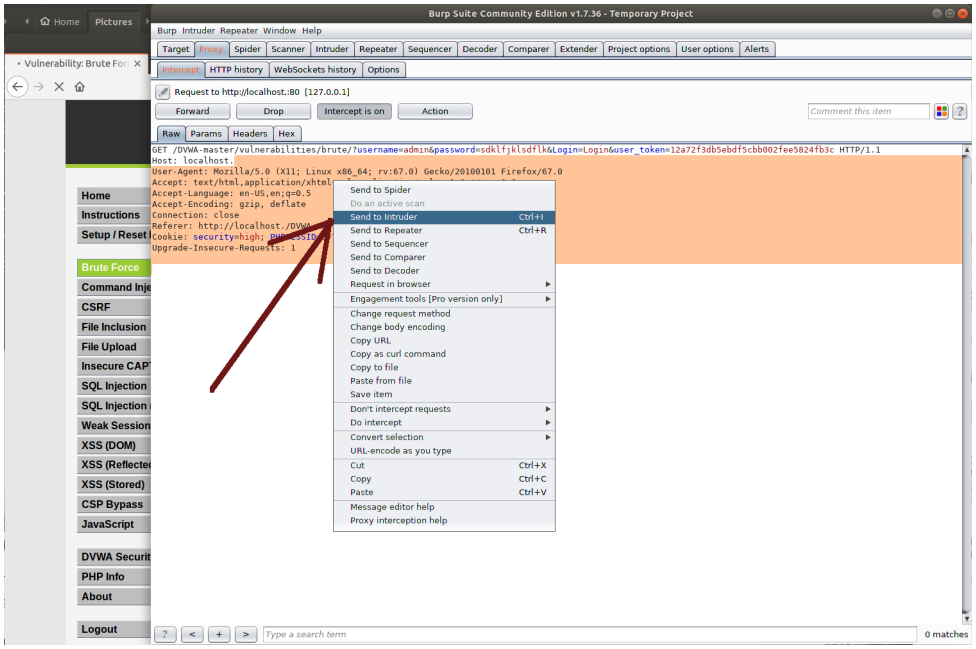
Dvwa’nın csrf token korunaklı html login form’una submit yapalım.



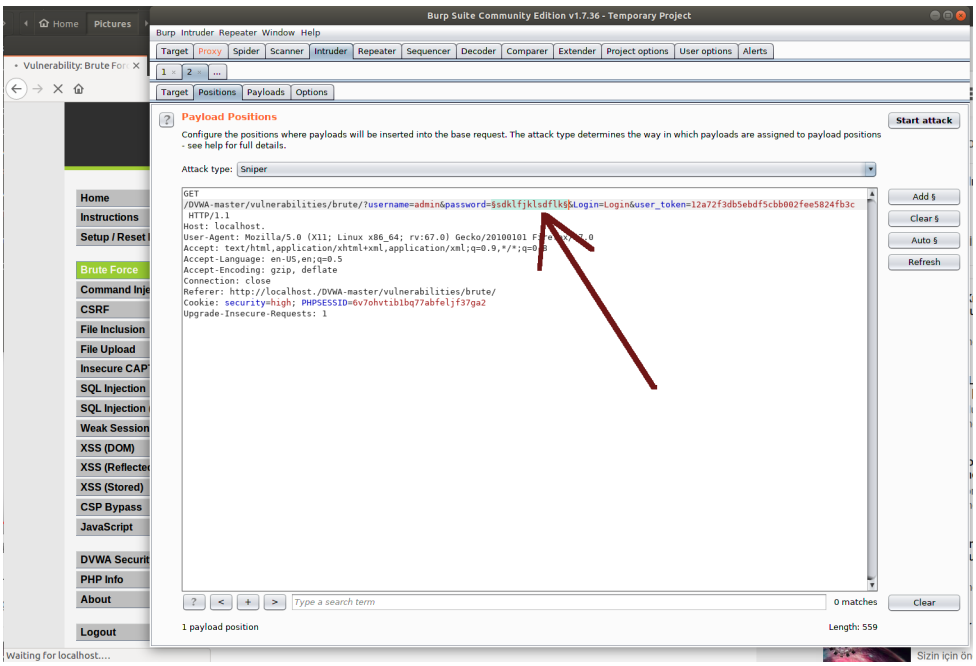
Burp paketi yakalayacaktır ve bekletecektir.



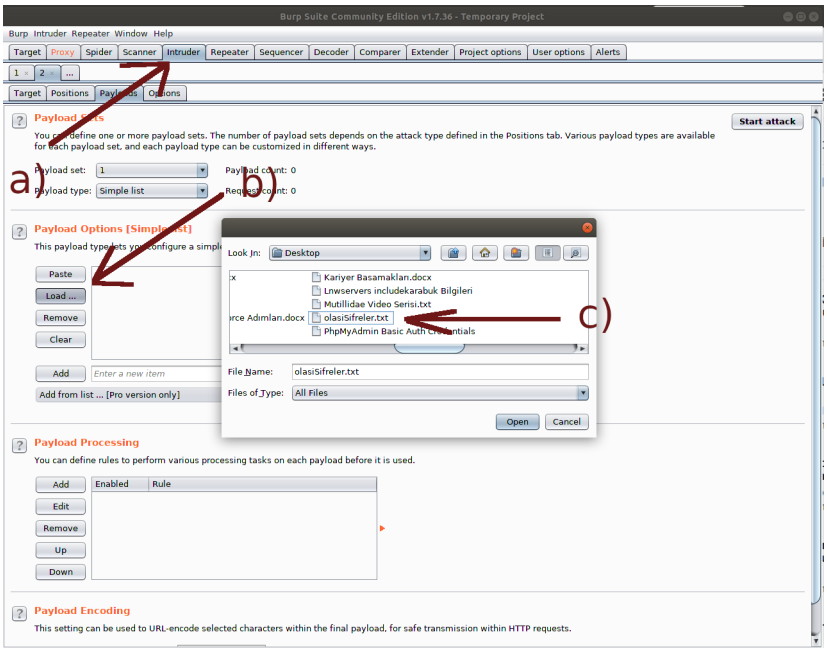
Paketi Intruder sekmesine yollayalım.

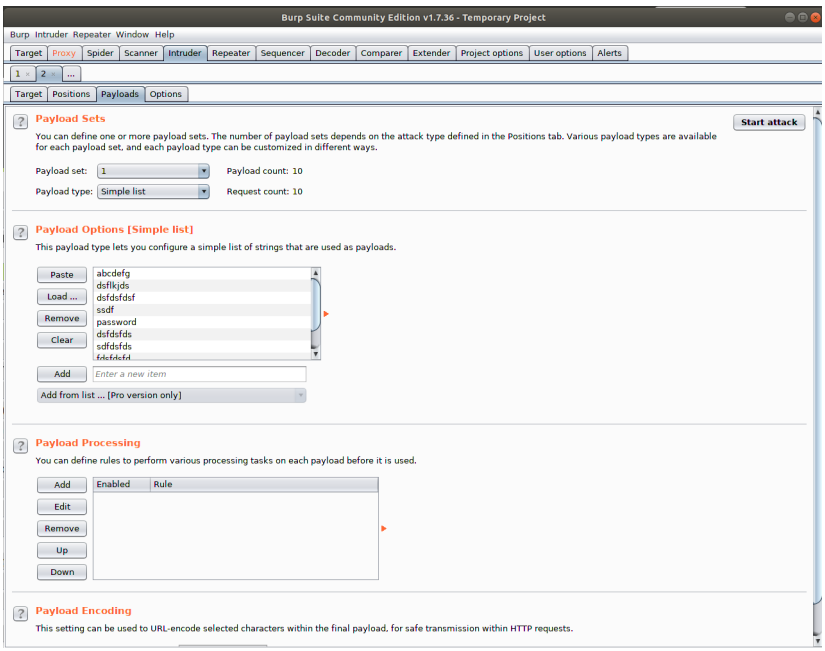


Intruder -> Positions sekmesi altında paketteki gönderilen parametrelerden username'e admin bilgisini girelim ve password kısmını ise değişken yapmak için $'la işaretleyim. CSRF token parametresini ise bırakalım. Zaten o parametreyi tanımladığımız makro dinamik bir şekilde güncelleyip duracaktır.

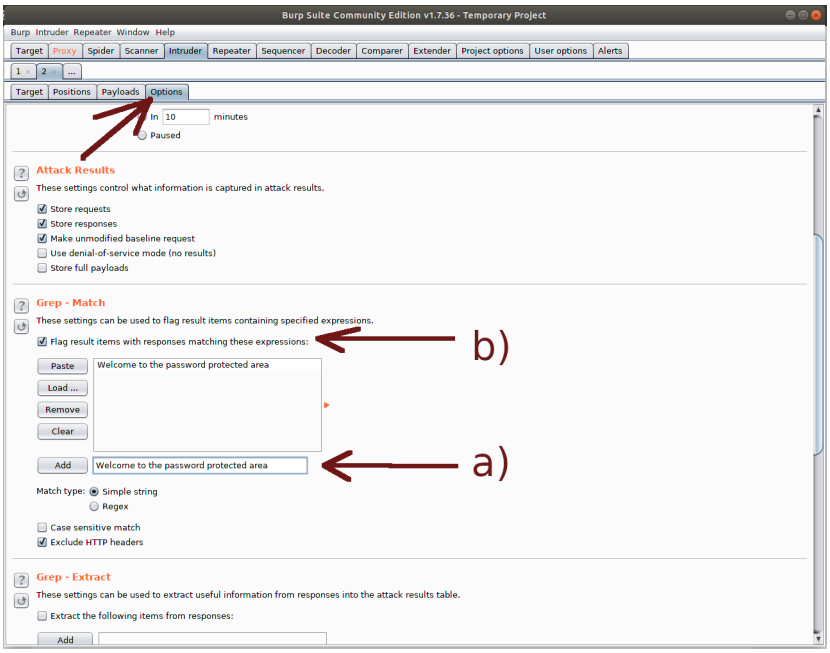


Intruder -> Payloads kısmından değişken yaptığımız parametreye bir sözlük dosyası atayalım.

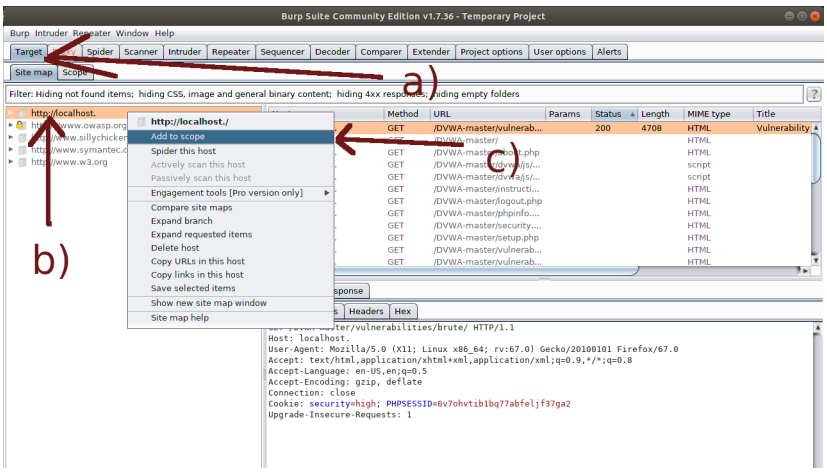




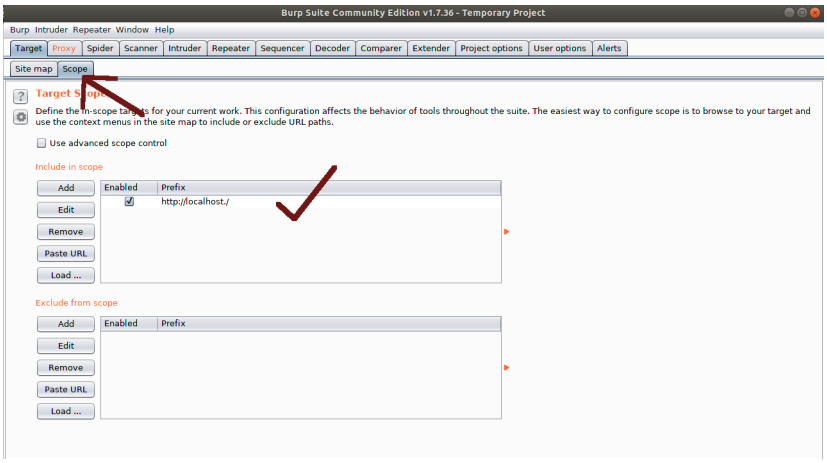
Ardından Intruder -> Options sekmesine gelip Grep - Match bölümüne sadece oturum açıkken görülebilecek Hoşgeldiniz vari bir cümleyi Add ile ekle ve Flag kutucuğunu tick'le işlemlerini yapalım. Böylece oturum açıldığında Hoşgeldiniz vari cümle tick işaretli olacaktır ve şifreyi kırdığımızı görebileceğiz.



Son olarak tanımladığımız makro'ya url scope olarak "Use suited scope" dediğimiz için burp'ün Target sekmesine gidelim ve üzerinde işlem yapacağımız url'i (DVWA-master localhost'ta olduğu için localhost.'u) scope olarak belirleyelim.

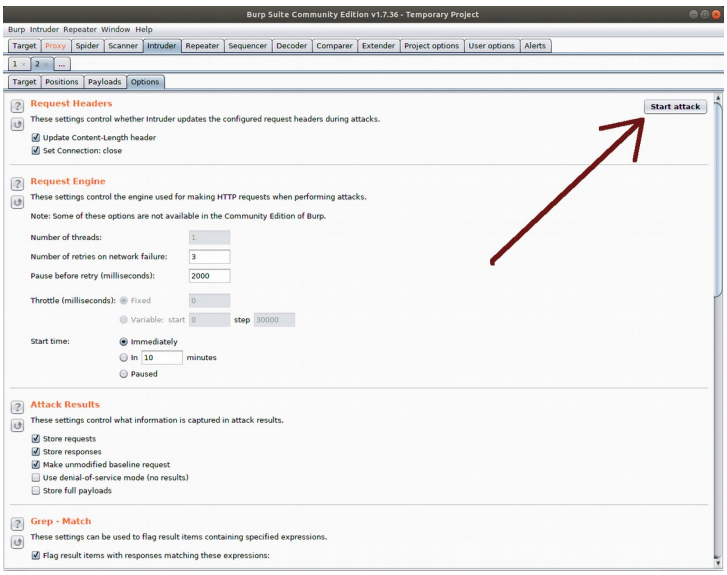


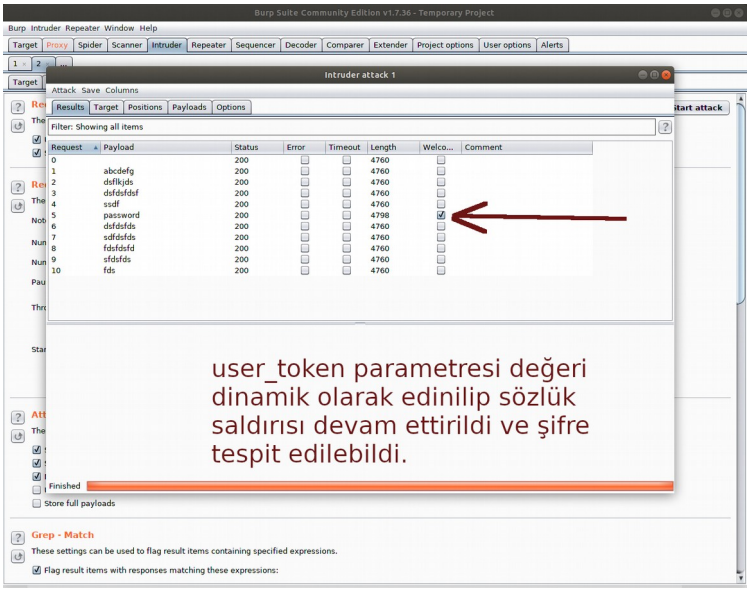
Scope eklemesi sorunsuz mu Target -> Scope'dan teyit edelim.



Not: localhost'un sonunda nokta olmalıdır. Diğer türlü güncel linux firefox'larda burp, localhost trafiğini kesememekte.

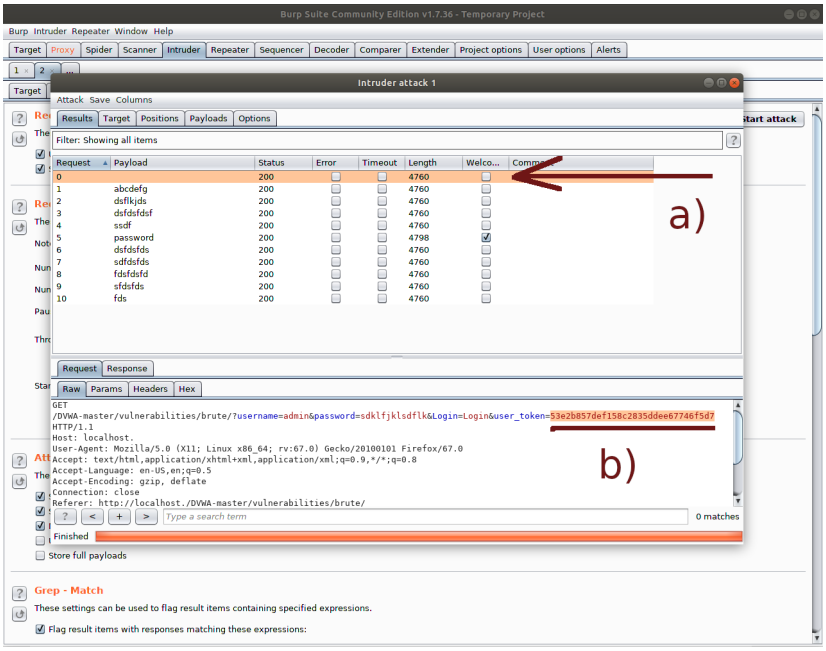
Ve saldırıyı Intruder -> Options -> Attack ile başlatalım.

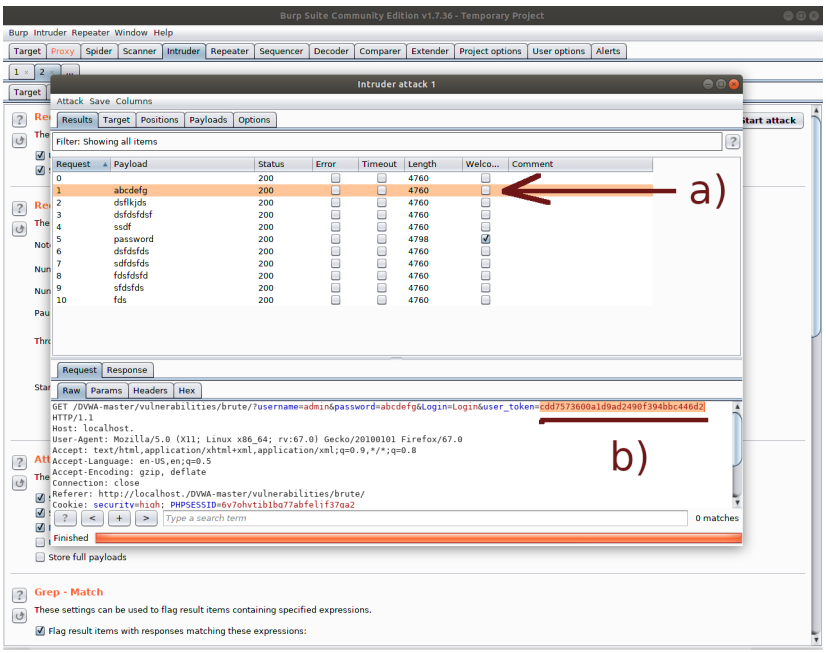


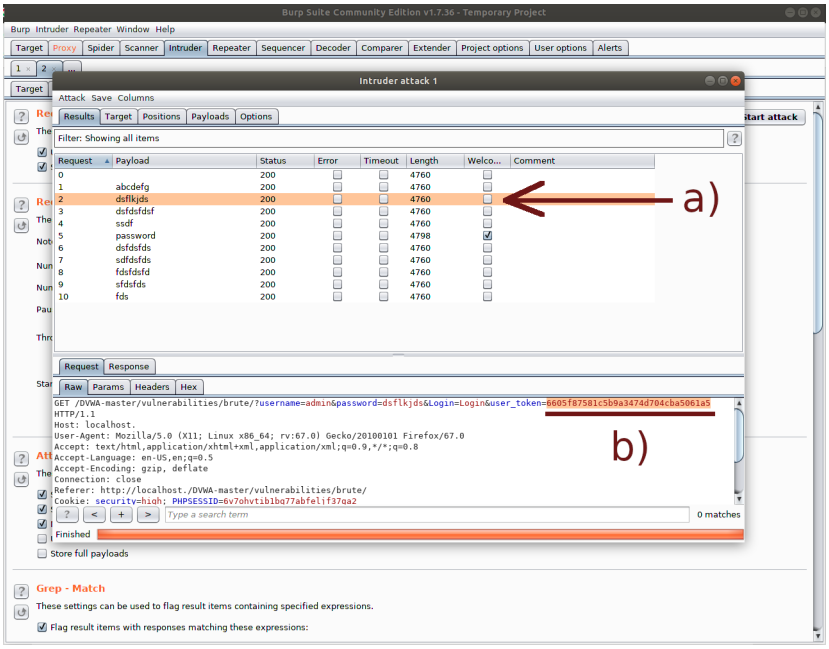


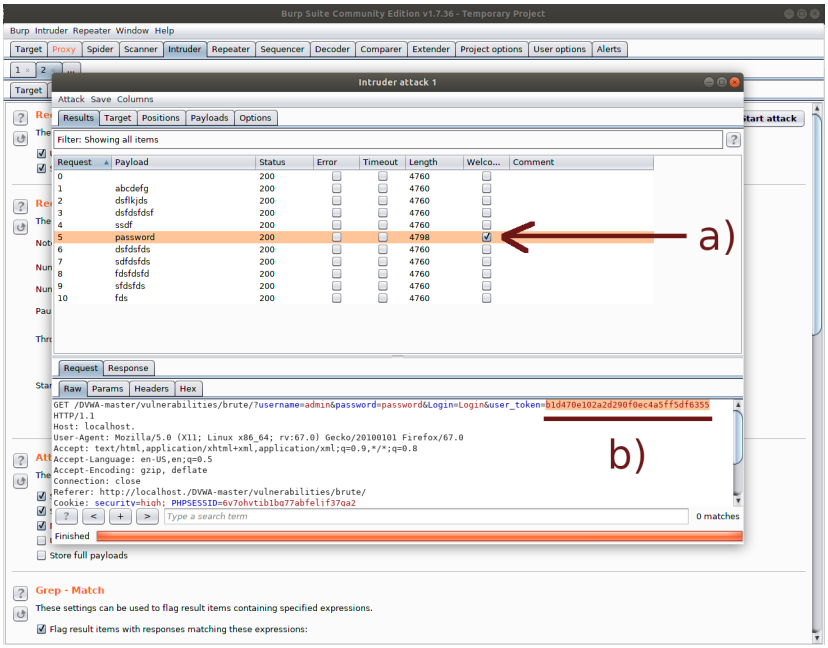
Böylece dvwa high seviyesindeyken brute force ekranında şifre kırma saldırısı yapabildik ve doğru şifreyi elde edebildik. Saldırı sırasında csrf token'ın güncellemelerinin yapılıp yapılmadığını gönderilen http talep paketlerindeki token parametresinin değerinin her gönderimde farklı oluşundan anlayabiliriz.

Aşağıda token parametresinin gönderilen her deneme paketinde farklı token değeri aldığını görüntülemektesin.




Burp Suite Community Edition v1.7.36 - Temporary Project
Intruder attack 1
a)
b)
GET /DVWA-master/vulnerabilities/brute/?username=admin&password=abcdefg&Login=Login&user_token=cdd7573600a1d9ad2490f394bbc446d2
Grep - Match


Burp Suite Community Edition v1.7.36 - Temporary Project
Intruder attack 1
a)
b)
GET /DVWA-master/vulnerabilities/brute/?username=admin&password=dsflkjds&Login=Login&user_token=6605f87581c5b9a3474d704cba5061a5
Grep - Match



Sonuç

High seviyesindeki Dvwa Brute Force ekranına burp ile makro tanımlaması yokken saldırı yapıldığında ekrana düşen denemelerde sürekli aynı csrf token değerinin karşıya gönderildiği görülmüştür. Yani burp ile araya girip yakalanan http talep paketinin sahip olduğu token değeri her defasında tekrarlanmıştır.

High seviyesindeki Dvwa Brute Force ekranına burp ile makro tanımlaması varken saldırı yapıldığında ise ekrana düşen denemelerde her defasında farklı csrf token değerinin karşıya gönderildiği görülmüştür. Yani burp ile araya girip yakalanan http talep paketinin sahip olduğu token değeri her denemede tekrarlanmamıştır.

Çünkü yakalanan http talep paketi üzerinde makro ile ilave bir işaretlemede bulunduk. Bu ilave işaretleme ile burp, sunucuya göndereceği her şifre denemesi paketinde paketi hazır hale getirirken (yani pakette şifre parametresine sözlük dosyasındaki bir satırı koyarken) ilave olarak pakette csrf token parametresine de bir değer koymaktadır. Bu değer en son gelen http yanıtındaki token değeri olmaktadır.

Yapılan ilave işaretlemenin direk Intruder sekmesi altında şifre parametresine yapıldığı gibi değil de ayrı bir sekmeden yapılmasının nedeni Intruder sekmesinin iş kapsamı dışında olmasındandır. Çünkü Intruder sekmesi elde edilen http talep paketi üzerinde kurallar koyup o paketi karşıya gönderme üzerine bir faaliyet gösterir. Burada ise http talep paketindeki bir değerin gönderildikten sonra gelen yanıttaki değere göre şekilleniyor olması gibi biraz daha dinamik bir kural söz konusudur. Bu nedenle Burp, modülerlik açısından bu kuralı bizlere Makro sekmesi altında yapma imkanı tanımaktadır.

Sonuç olarak tanımlanan kurallar (intruder ve makro kuralları) neticesinde paket tam hazır olduktan sonra karşıya yollanarak sözlük saldırısı sürmüştür ve doğru şifre elde edilebilmiştir.

Kaynaklar

https://support.portswigger.net/customer/portal/questions/17460071-csrf-token-extraction-in-forms-responding-with-3-2-redirect-headers

https://support.portswigger.net/customer/portal/questions/17408949-burp-suite-anti-csrf-post-

https://www.youtube.com/watch?v=U43o5cCVfXo