|  |
| --- |
| ***XSS Zafiyeti*** |
| Kaan Efe Öğüt  *ADLİ BİLİŞİM MÜHENDİSLİĞİ* |
| XSS nedir ve Web for Pentester laboratuvarı üzerinde XSS zafiyeti nasıl uygulanır bunun hakkında bilgi vereceğim. |

**01.08.2021**

***XSS***

-Genelde web uygulamalarında bulunan bir güvenlik açık türüdür.

-Web sayfalarına dışardan müdahale ile HTML ve JS kodları çalıştırılabilir.Saldırgan kişiler yazdıkları zararlı JS kodlarıyla kurban kişilerin cookie,tarayıcı,IP bilgileri ve daha fazlasını bu güvenlik açığını kullanarak çalabilirler.

-Kullanıcıların veri girişi yapabileceği arama,yorum vb. text bölümlerinde bulunur.

3 Adet türü vardır.

-**Reflected XSS :** Bu saldırı türü güvenlik açığı bulunan sitede uygun alanlara yazılan JS kodlarının çalıştırılması ile olur.İlgili JS kodu o an çalıştırılabilir veri tabanına kayıt edilemez.

-**Stored XSS :** Bu saldırı türü Reflected XSS’e göre daha tehlikelidir.Saldırgan kişinin yazmış olduğu zararlı kodlar veritabanına kaydedilir.Sayfayı ziyaret eden herkes bu zararlı JS kodlarının istismarına uğrar.

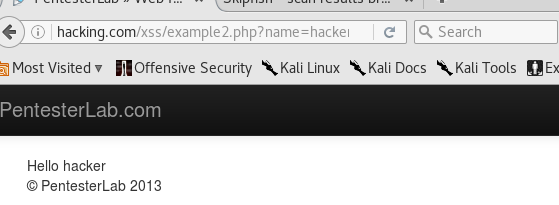
-**DOM XSS :** Bu saldırı türü en tehlikeli olandır.Virüs,trojan gibi zararlı kodlar sayfaya entegre edilebilir.Sitede gezinirken başka sayfaya yönlendirebilir.

-BeEF-Xenotix-XSS Me(Firefox Eklentisi),Open NTX,Net Sparker gibi araçlar kullanılarak bu zafiyetten yararlanılabilir.

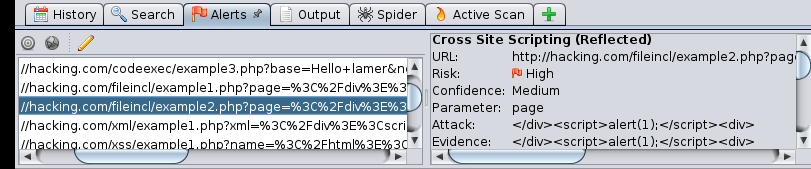
-Bu zafiyet türü için Web for Pentester üzerinden 6 adet Uygulama yapılacaktır.

**Uygulama 1 :** Alternatif XSS Payload tespiti.

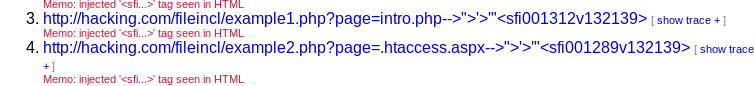
-Bu örnekler üzerinde; dışarıdan kullanıcı girilmiş bir sayfaymış gibi düşüneceğiz.



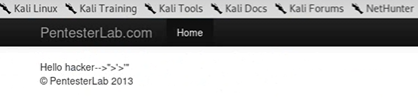
- Öncelikle WFP zafiyeti olan web sitesinin görünümünü bakmak istiyorum.Basitdüze bir web sitesi olduğunu görüntülüyorum.



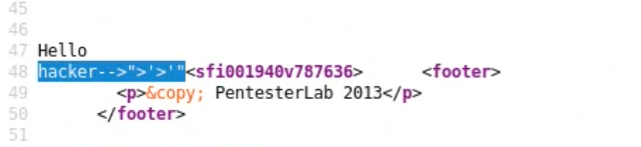
-Zaproxy aracımın bu zaafiyeti tespit edip edemediğine bakıyorum ve kullanmış olduğu payloadı görüntüleyebiliyorum.



- Skipfish aracımın bu zafiyeti tespit edip edemediğine bakıyorum.Tag ekleme işlemi ile zafiyeti kullandığını görüyorum.



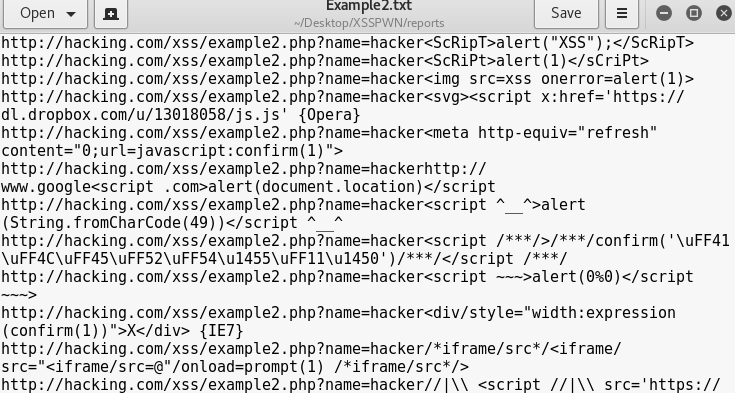
-Skipfish üzerinde ki zafiyeti açtığımda ekrana herhangi bir uyarı vermiyor.



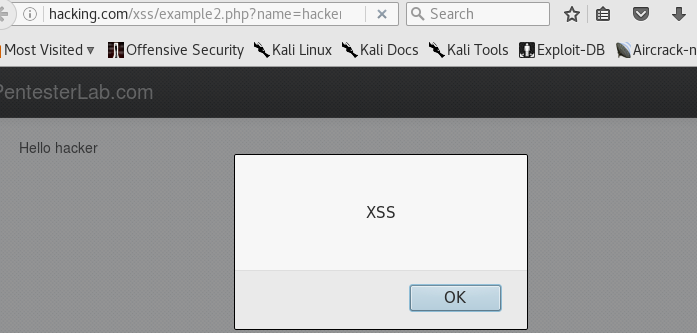
-Ardından kaynak koduna indiğimde çalıştırdığı kodun kaynak koduna eklendiğini görebiliyorum.



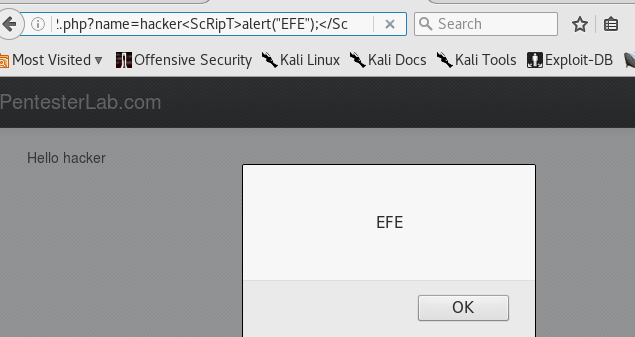
- XSS zafiyetin de kullanmak için “Xsspwn” aracı üzerinde payload denemesi yapacağım.Öncelikle WFP üzerinden linki kopyalıyorum.İşlemi “Xsspwn” üzerinden başlatıyorum.



- XSSPWN üzerinde başlatmış olduğum işlem bitti ve Bana kullanabileceğim Payloadları listeledi.İçerisinden bir tanesini deniyorum.



- Çalıştırdığım payloadı sitenin URL’sine eklediğimde karşıma bir alert geliyor.



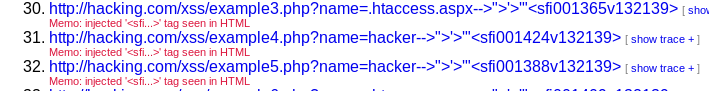
- Kullanmış olduğum payload üzerinden link üzerinde bir oynama yaparsam EFE yazdırabiliyorum.

-Ayrıca linke baktığımızda büyüklü küçüklü şekilde yazdığını görüntülüyorum. Bunun sebebi ise küçük harflerle yazıldığında script engelliyor.

**Uygulama 2 :** Uzak sunucuda ki PDF’i çalıştırma.



-Hem URL hem de site yapısı hakkında bilgimiz olsun diye öncelikle zafiyetin bulunduğu siteyi açıyorum.



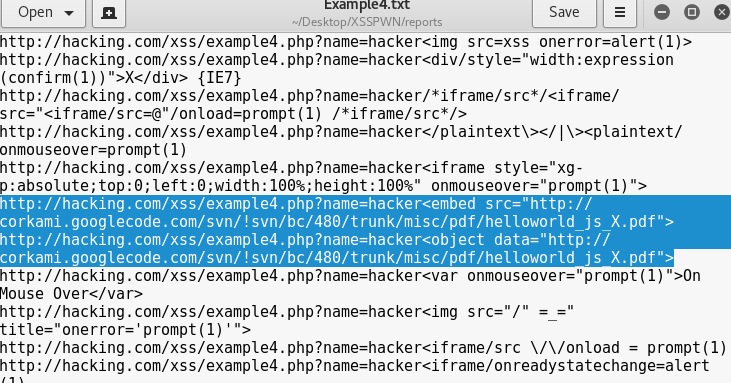
- Ardından skipfish aracı üstünden yapmış olduğum tarama sonucuna bakıyorum.



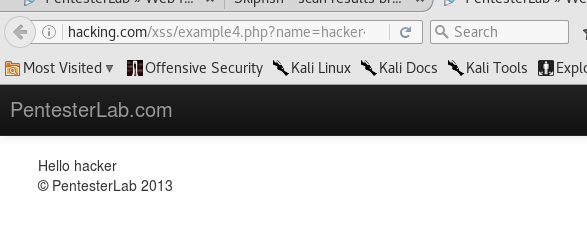
- Zaproxy üzerinden de baktığımda zafiyeti başarılı bir şekilde bulduğunu söyleyebilirim.



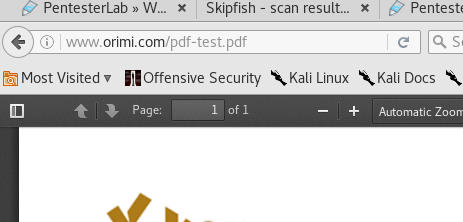
- XSS açıklarında kullandığım XSSPWN aracıyla devam edeceğim.Bunun için zaafiyet bulunan sitenin linkini kopyalıyorum ve Xsspwn'i çalıştırıyorum.



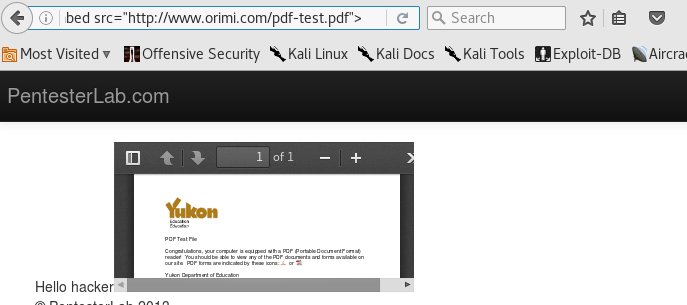
- Payload listesini açtığımda içinde linke eklenen bir uzantı ile pdf eklentisini çalıştırdığını görüntülüyorum.Bu linki kopyalayıp çalıştırmayı deniyorum.



- Linki açtığımda bana herhangi bir dönüt olmadığını görüyorum.Bunun sebebini karşı tarafta bir pdf dosyası olmamasından dolayı olduğunu düşünüyorum.



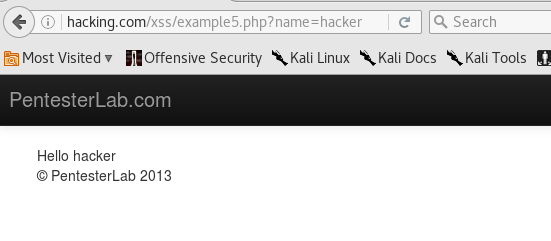
- Bu sebeple internet üzerinden bir pdf dosyası açıyorum ve açtığım pdf dosyasının linkini kullanarak sunucu üzerinde işlem yapmaya çalışacağım.



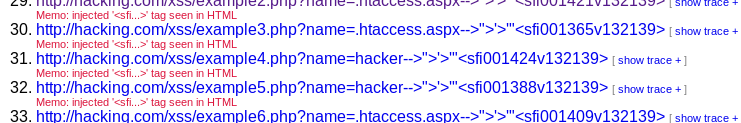
- Linke ekleme yaptığımda zafiyetten yararlandım ve uzaktaki bir pdf dosyasını sanki oradaymış gibi görüntüleyebildim.

-Sunucuda veya sunucu tarafına başka bir pdf eklenebilir ve çalıştırılabilir.

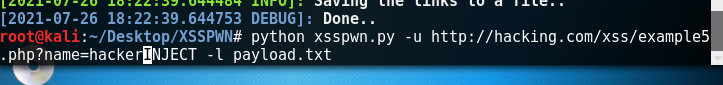
**Uygulama 3 :** XSS ile hedefe Trojan indirtme.



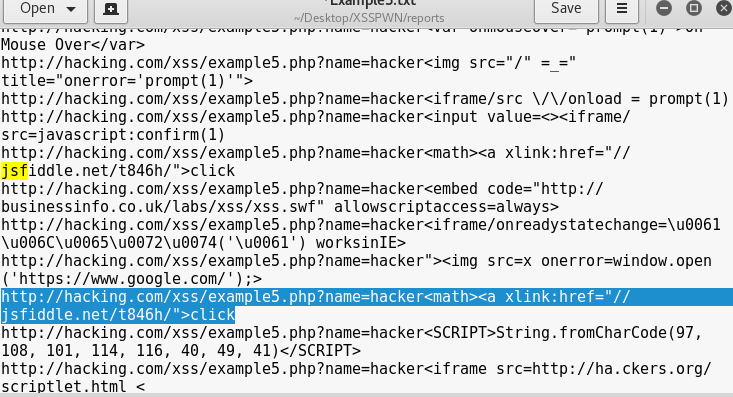
- Öncelikle zafiyet bulunan siteye göz gezdirelim.



- Zaproxy aracı zafiyeti tespit edemedi.Skipfish aracı zafiyeti tespit etti.

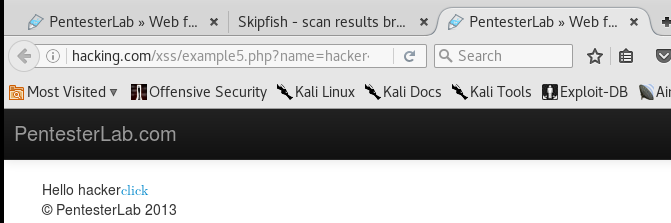


-Ardından Xsspwn üzerinden bir zafiyet işlemi gerçekleştireceğim.Bunun için URL'yi kopyalıyorum.



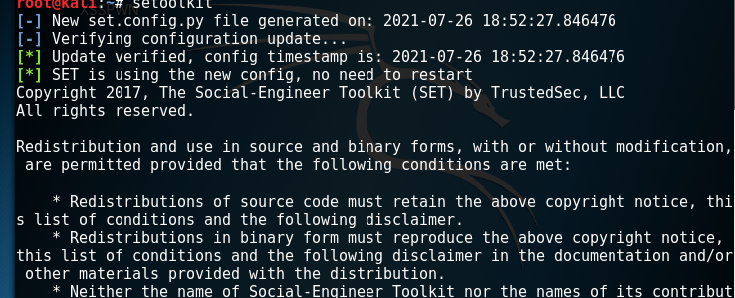
- XSS işleminden sonra döndürdüğü payloadlara bakıyorum.Burada amacım hedef sunucuya bir şey indirtmek olduğu için payloadlar içerisinde o işlemi yaptırabileceğim bir şey arıyorum.

-Bulduğum payload sayesinde sunucuya istediğim dosyayı indirtme işlemini gerçekleştireceğim.

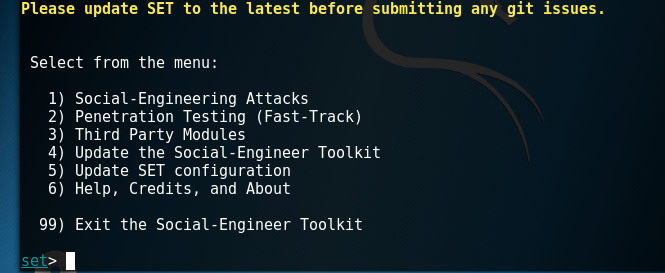


- Bu payloadı tarayıcı üzerinden çalıştırıp görsel olarak nasıl gözüktüğüne bakalım.

-Açılan site üzerinde "click" butonuna bir virus enjekte ederek karşı bilgisayarda çalıştırabilirim.



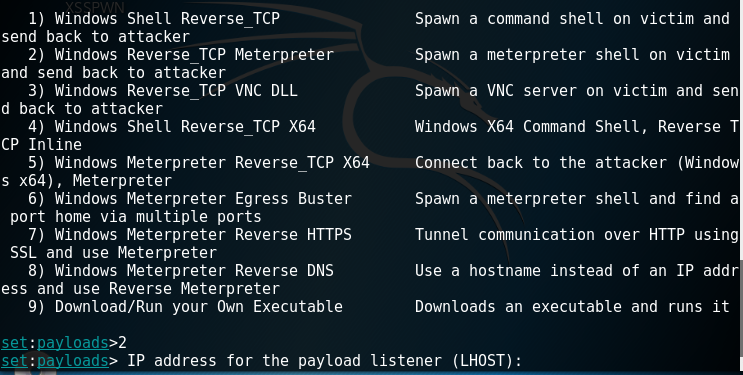
- Öncelikle bir Trojan oluşturalım bunun için “Setoolkit” kullanacağım.”Setoolkit” komutu ile Trojan oluşturma işlemini başlatıyorum.



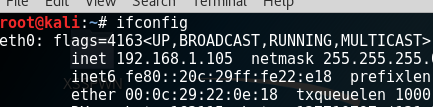
-Açılan sekmede “Social-Engineering Attack” ile devam ediyorum.



- Ardından gelen sayfa üzerinde “Create a Payload And Listener” seçeneği ile devam ediyorum.



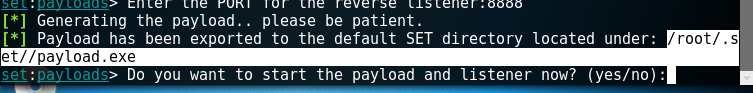
- Hedefe uygun şekilde istediğimizi seçebiliriz.”Reverse\_TCP\_Meterpreter” ile devam ediyorum.Ardından gelen sekmede bana LHost soruyor.



- LocalHostu öğrenmek için İfconfig komutunu çalıştırıyorum.Ip adresimi öğreniyorum.



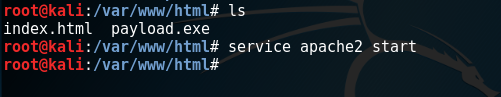
-Ardından “Setoolkit” e dönüyorum ve LHOST kısmına Ip adresimi yazıyorum ve dinlenecek port olarak 8888 veriyorum.



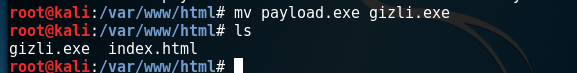
-Payload'ım oluştu ve yolunu kopyalıyorum ve ardından “Yes” yazıp programı dinlemeye başlatıyorum.



-Ardından oluşturduğum Trojanı Linuxun Apache server dosyalarına ekliyorum.



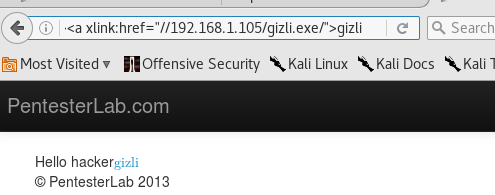
-Dosya kopyalanmış mı diye önce teyit ediyorum ardından start komutu ile Apache serverı başlatıyorum ki dışarıdan veri akışı olsun.



-Virüsün ismini gizli olarak değiştiriyorum ve teyit ediyorum.

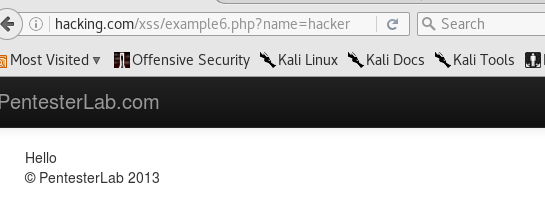


-Gizli dosyasının dışarıdan ulaşılacak şekilde bir URL yapıyorum.

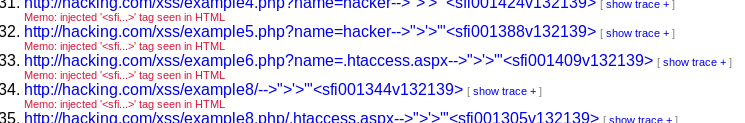


-Ardından zafiyetin bulunduğu payload içerisine bu exe dosyasını gömüyorum.Başarılı bir şekilde gömdüm.

**Uygulama 4 :** Base64 ile XSS Tespiti.



-Öncelikle zafiyetin bulunduğu siteyi göstermek istiyorum.



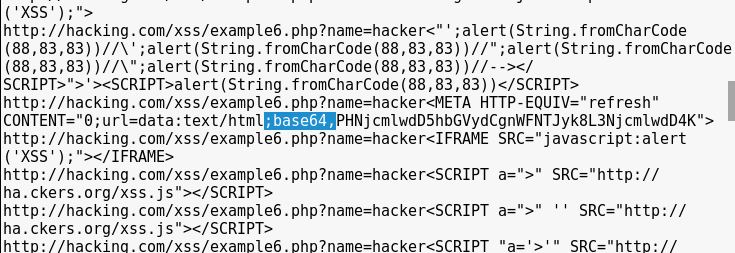
-Skipfish aracıyla zafiyet taraması yaptığımızda zafiyeti başarılı şekilde buluyor.



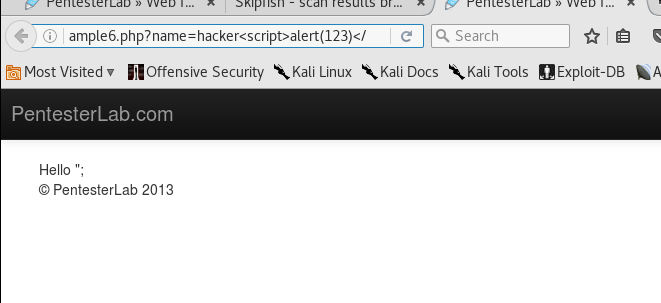
-Owasp ZAP aracıda zafiyeti buluyor.



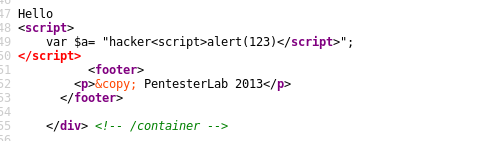
-Ardından XSSPWN aracı ile tarama işlemi yapıyoruz.



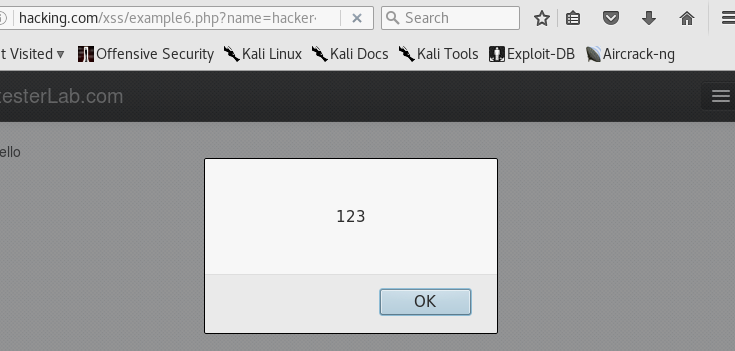
-Sonuçlara baktığımda base64 ile script kodlarını gizlediklerini görüntüleyebiliyorum.



-İçerisinde script kodu bulunan paylaodlardan deneme yapıyorum.Fakat script kodunu ekrana bastırmıyor.



-Bunun sebebini kaynak kodu üzerinden görüntülediğimde script kodunu bir değişkene atadığını görebiliyorum.

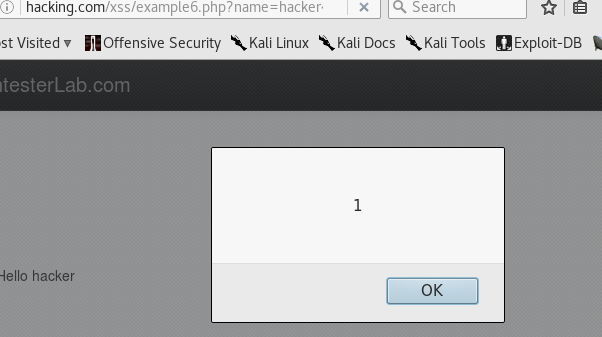


-Bunu çözmek için sonuna aynı script kodunu bir kere daha ekleyip deniyorum ve karşıma script olarak geliyor.

-İkinci olarak linke eklediğim payloada eklenen scriptler ile farklı işlemler yapılabilir. Bunun için önce Base64 ile yazılan payloadı kopyalıyorum ve ikinci olarak bu sefer base64 paylaodını ekliyorum.



-Paylaod üzerinden name='den sonraki kısmı kopyaladım.



-Şimdi ise 2.payload olarak ekliyorum ve karşıma yine alert bölümü geldi.

>>>>>



-Base64 ile gizlenmiş linki base64decoder ile dönüştürüyorum.

-Dönüştürme işleminden sonra bir script kodu olduğunu görüyorum.

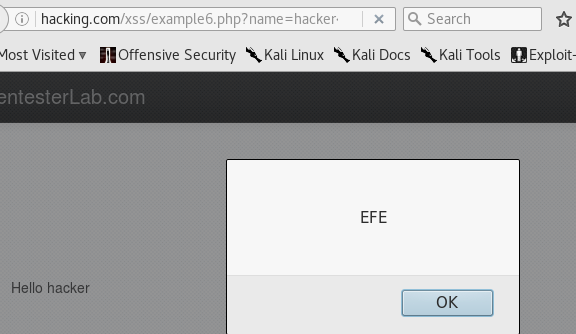
-Script tagleri engellendiğinde base64 ile atlama işlemi yapılabilir.

>>>>>>>>>



-Karşıma çıkan script kodunu kendi isteğim doğrultusunda değiştiriyorum ve içine "EFE" ismini ekliyorum ardından decode işlemini gerçekleştiriyorum.

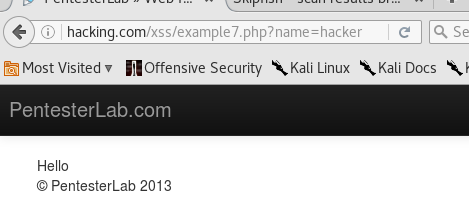
-Encode işleminden sonra gelen linki bu sefer 2.payload olarak ekliyorum.



-Kodu çalıştırdığımda karşıma "EFE" şeklinde bir uyarı geliyor.

-Zafiyetten başarılı bir şekilde yararlanabildim.

**Uygulama 5 :** Alternatif Payload Listeleriyle Çalışmak.



-Öncelikle zafiyeti olan siteyi göstermek istiyorum.



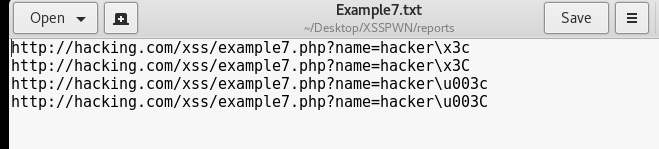
-Skipfish aracı site üzerinde ki zafiyeti tespit edemedi.



-Zaproxy aracı güvenlik açığı başarılı şekilde buldu.



-Ardından XSS üzerinde her zaman olduğu gibi XSSPWN aracı ile bir payload denemesi gerçekleştirdik.Çalıştırma komutum yine önceki örneklerde yaptığım gibidir.



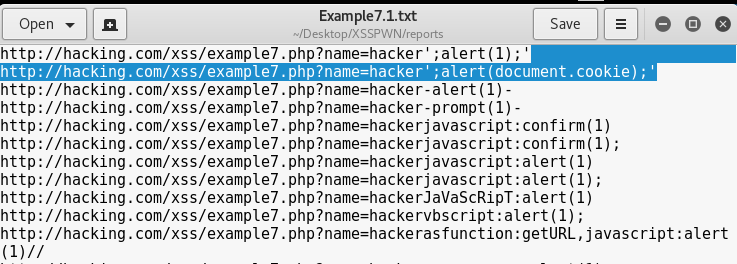
-Ardından bulmuş olduğu payloadlara baktığımda 4 tane payload bulduğunu görüyorum ve çıkan payloadlar çok fazla ufuk açıçı payloadlar değil.

-XSSPWN aracnın payloadının bu zaafiyette yetersiz olduğunu görüyorum bu yüzden farklı paylaod denemesi yapıyorum.

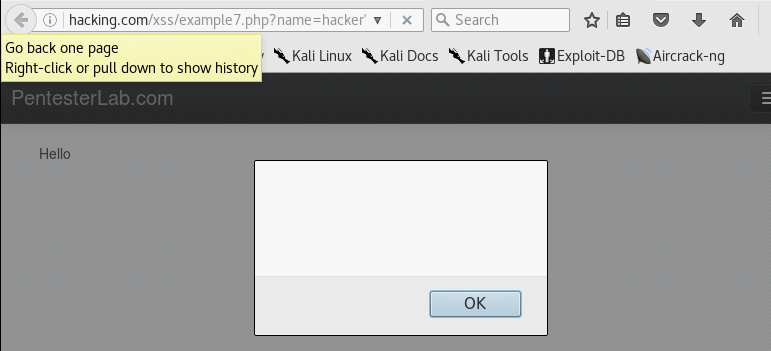


-1632 tane payloadın bulunduğu .list dosyası üzerinden işlem yapıyorum.Eğer burada hata alırsanız payload'ın uzantısını .txt olarak değiştirebilirsiniz.

-Yapmış olduğum işlem yaklaşık 1 saat sürdü.



-Sonucunda elde ettiğim payloadlar şu şekildedir.Dikkatimi içerisinde cookie kelimesi bulunduran payload çekiyor.

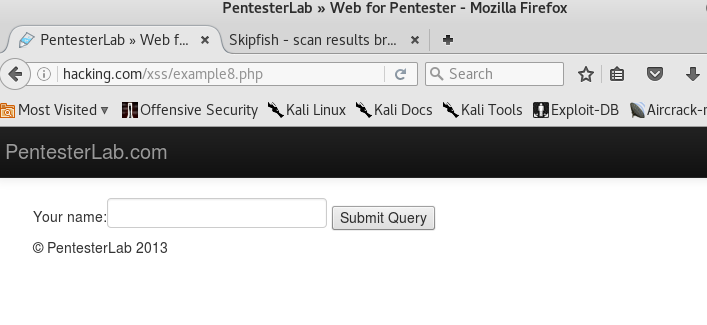


-Bu paylaod’ı çalıştırdığımda boş bir alert kutusu ile karşılaşıyorum.Zafiyet bulunan site üzerinden bir giriş çıkış işlemi olmadığı için cookie’yi görüntüleyemedik fakat

-Giriş/çıkış sistemli çalışan bir yer olsaydı bütün bilgiler karşımıza gelecekti.

-Cookie ise sisteme giriş yaptığımızda arka planda dönen değerdir.

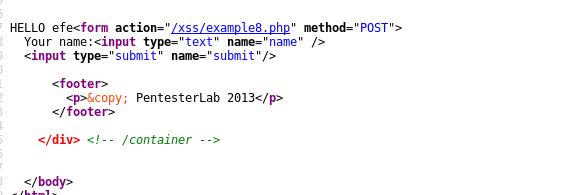
**Uygulama 5 :** Yarı otomatik araç ile payload tespiti.



-Öncelikle zaafiyetin bulunduğu sayfayı göstererek başlamak istiyorum.Diğer sitelerden tek farkı görsel olması.Ama dikkat etmemiz gereken içerisine bir değer girdiğimizde biz bir dönüt oluyor mu?



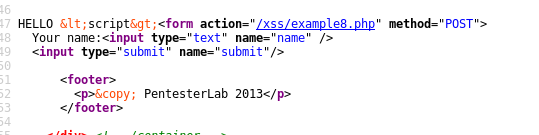
-İsim yazıp döndürüyorum ve bana bir geri dönüş oluyor.Kaynak koduna baktığımda daha detaylı bilgi sahibi olabilirim.



-Kaynak koduna geçiş yaptığımda Post metodu ile bir form alındığını görüntülüyorum.

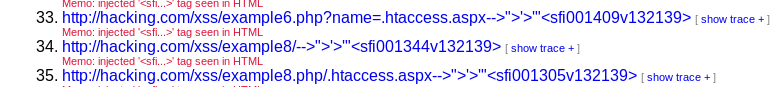


-Öncelikle <script> kodunu çalıştırabiliyor muyum diye deniyorum.Script kelimesini çalışıyor şimdi ise kaynak koduna bakıyorum.



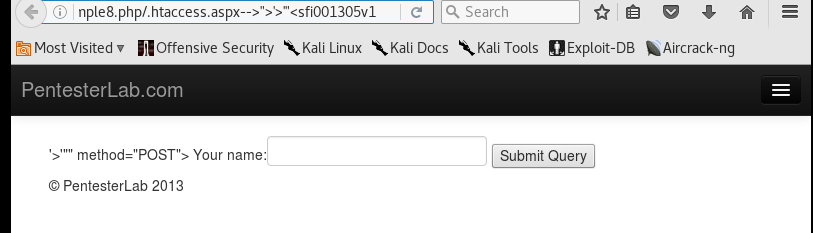
-"<>" karakterleri form içerisinde değişime uğramış.Bu formdan çıkabilirsem bir zaafiyetten yararlanabilirim diye düşünüyorum.

-Ama bunu manuel olarak yapsam işlemler hem çok uzun sürecek belki binlerce deneme yapmam gerekecek hem de gözden kaçırma gibi bir ihtimal de bizi karşılıyor.



-İşlemlere başlamadan önce birde Zaproxy ve skipfish araçlarına bakıyorum.Zaproxy aracı zafiyeti bulamadı.

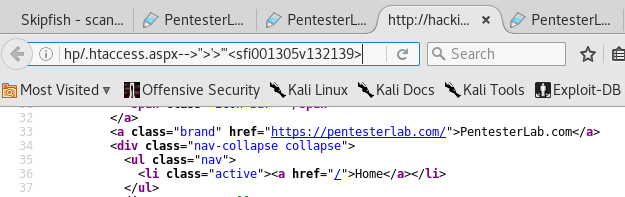
-Skipfish aracına baktığımda zafiyeti bulmuş.Bulmuş olduğu zafiyetleri görüntülüyorum ve sayfalarını incelemek istiyorum.



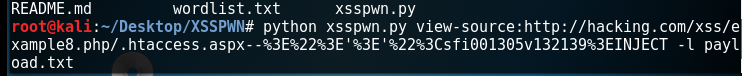
-Web üzerinde görüntülendiğinde bu şekilde fakat bana asıl bilgiyi verecek olan yer kaynak kodu bu sebeple kaynak kodunu görüntülüyorum.



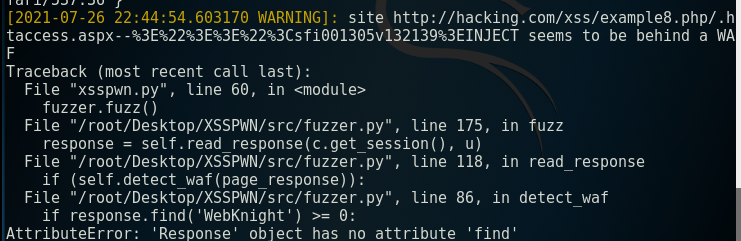
-Kaynak kodunu görüntülediğimde form ile kaynak koduna giriş yapılmış fakat “<>” içerisinde kalan alan ile form nesnesinin dışarısına çıkılmış bu da tam benim yapmak istediğim işlem.



-Link üzerinden baktığımda bu formatta <> olan yere bir “XSS payload” denemesi yaparak işlem yapabilirim.



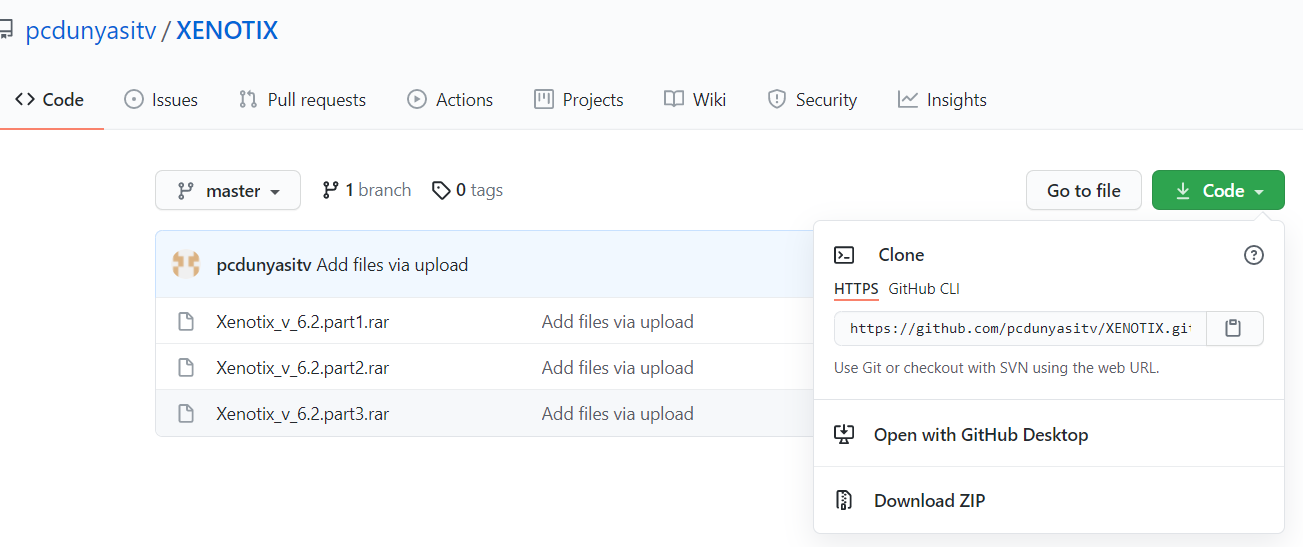
-Bu işlem için direk aklıma Xsspwn geliyor.Xsspwn'i açıyorum ve gerekli komutları yazıyorum ve işlemi başlatıyorum.



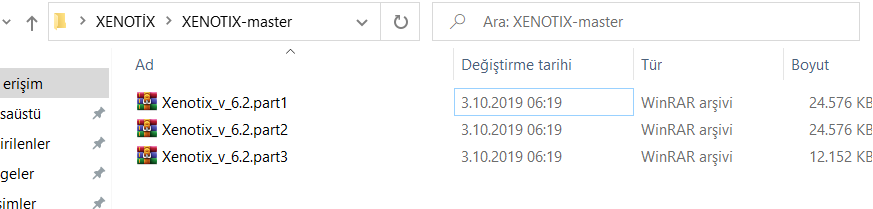
-XSSPwn bir güvenlik duvarıyla karşılaştığını ve işleme devam edemeyeceği ile ilgili bana bir bilgi veriyor.

-Bu işleme ya manuel olarak devam edicem ya da başka bir uygulama üzerinden yaptığım işleme devam edicem.

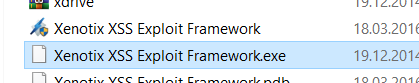
-Bu işlemler için kullanabileceğim yarı otomatik bir araç var(XETONIX) ve onu kullanmak istiyorum.Kullanacağım araç bu sefer linux üzerinde değil windows üzerinde kullanılabilir bir araç.



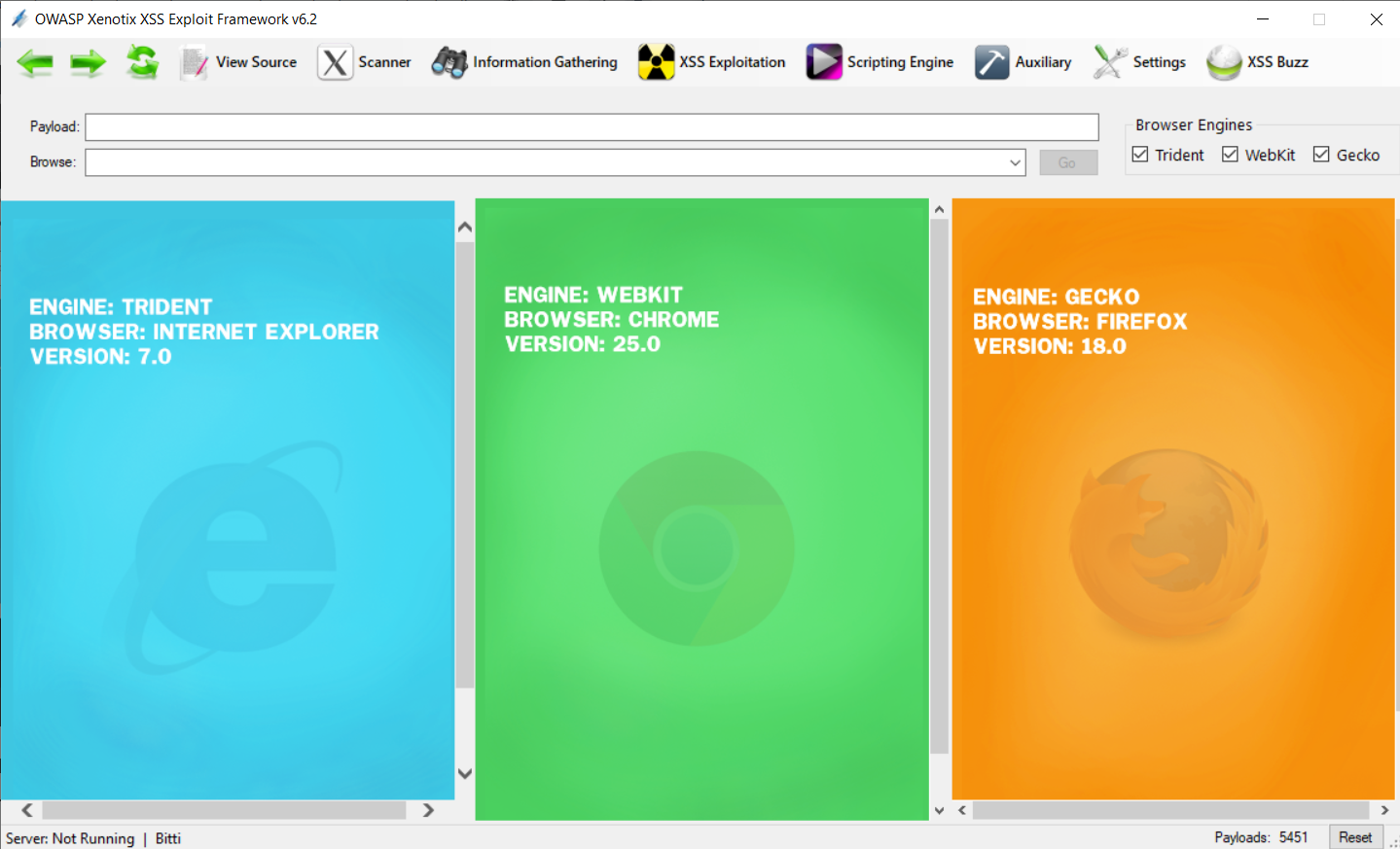
-Windows üzerine geçiş yapıyorum ve github üzerinden XETONIX aracını indiriyorum.



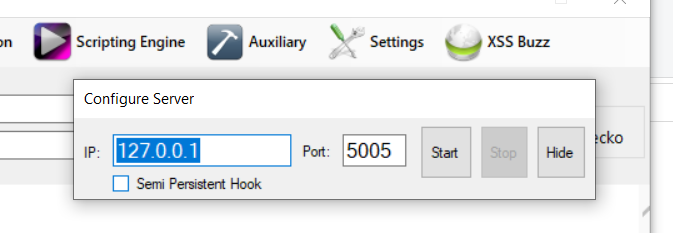
-İndirme işlemi sonrasında rardan çıkardığımda beni 3 adet daha rar dosyası karşılıyor.Tüm rarları bulunduğum dizine çıkarıyorum.



-Framework ü çalıştırıp uygulamayı açıyorum.



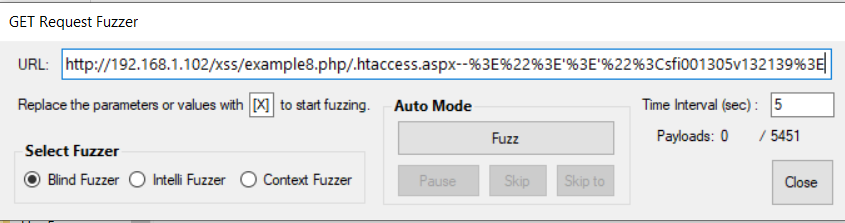
-Framework’ü çalıştırdığımda karşıma böyle bir ekran geliyor ve Browser Engines kısmından Trident Webkit ve Gecko’yu kapatıyorum.



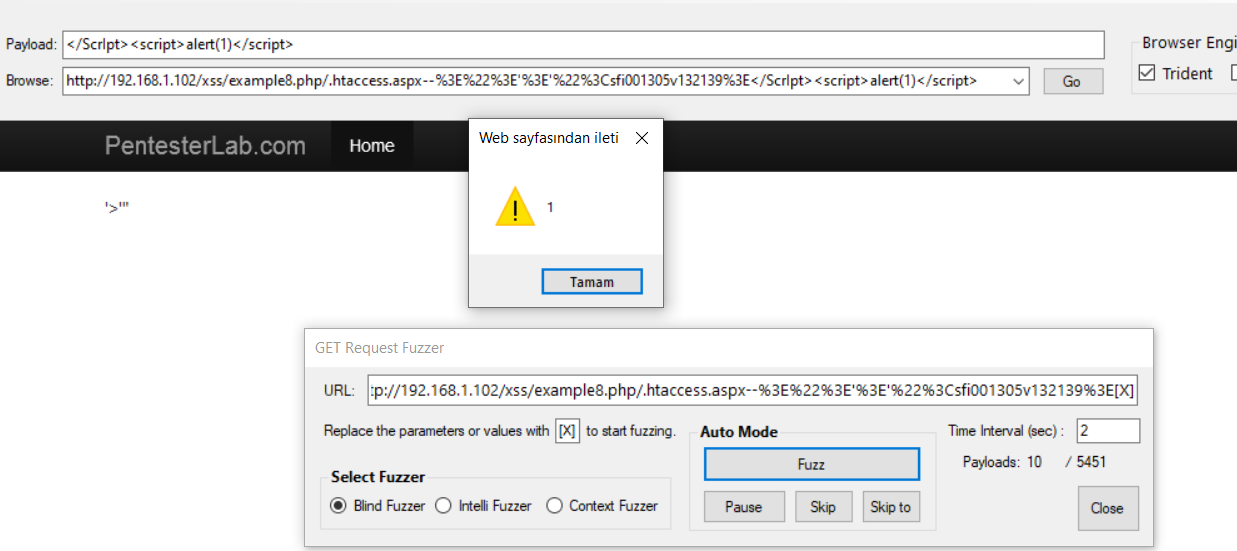
-Settings bölümünden “configure server” ile devam edip server'ı başlatıyorum.



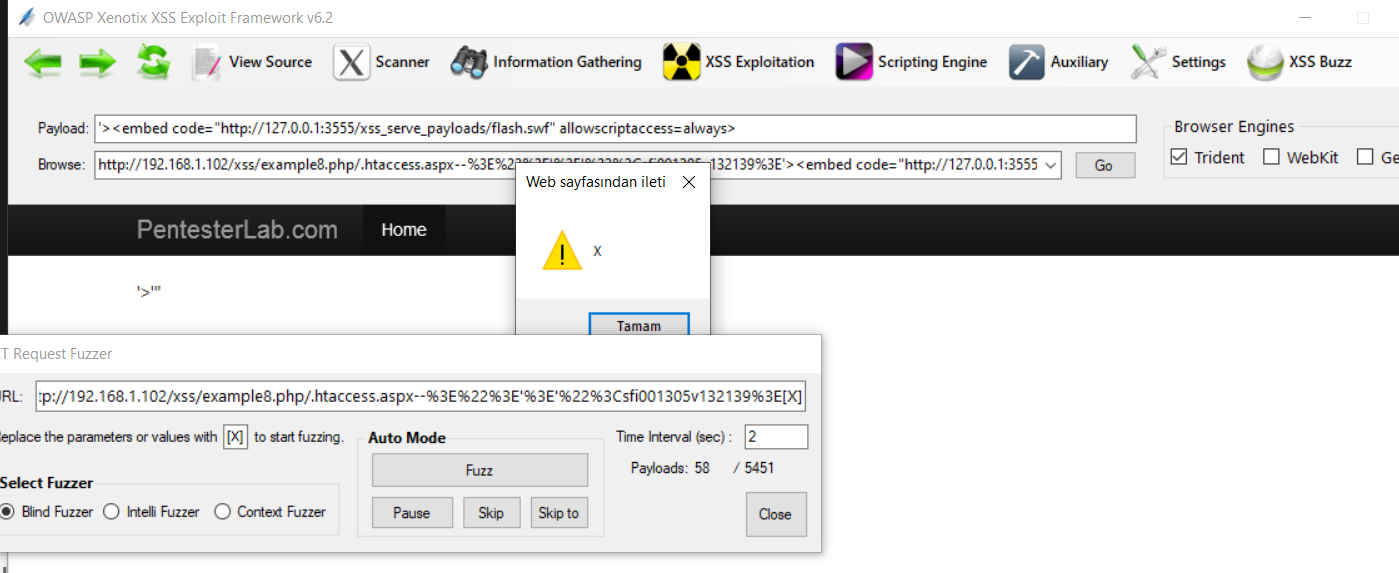
-Server'ı başlattıktan sonra Scanner bölümünden “Get Request Fuzzer” ile devam ediyorum.



-Ardından ise XSSPWN üzerinde deneme yaptığım URL'yi buraya yapıştırıyorum.Sonuna ise XSSPWN üzerinde INJECT yazıyordum.Burda ise [X] koyuyorum. Ardından “fuzz” diyerek başlatıyorum.

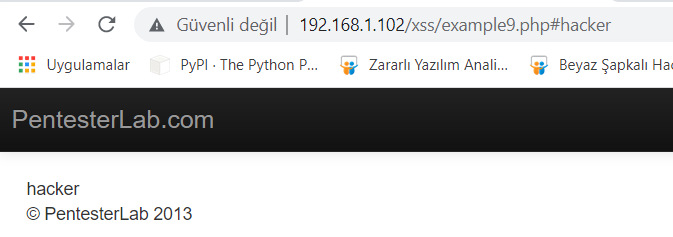


- İlk hatayı aldık ve pause diyerek programı durduruyorum.PAyload bölümünde kullanılabilir payloadı tarafımıza aktardı.Script Alerti kapatıp continue ile devam ediyorum.



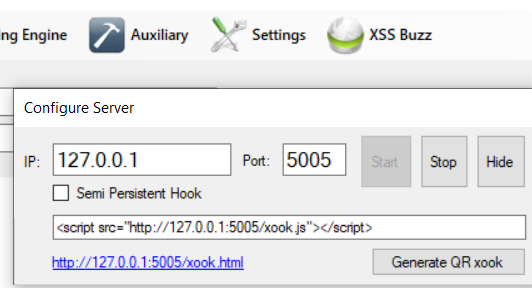
- Ardından aynı işlemi sürekli tekrarlıyorum burada kullanabileceğim paylaodları sıralıyor.Programın tek eksisi bu payloadları kaydetmemesi bu sebeple bunları bir txt dosyası açıp içine kaydediyoruz.

**Uygulama 6 :** DOM XSS

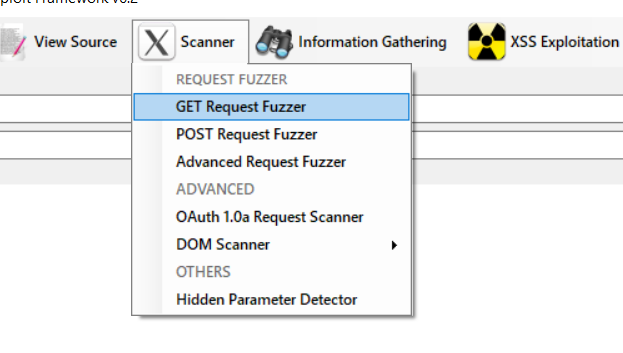


-Öncelikle siteyi görüntülüyorum ve bu açığı zafiyet araçları bulamıyor.Sadece link üzerinden bir işlem yapılıyor.Kaynak koduna işlem eklenmiyor.

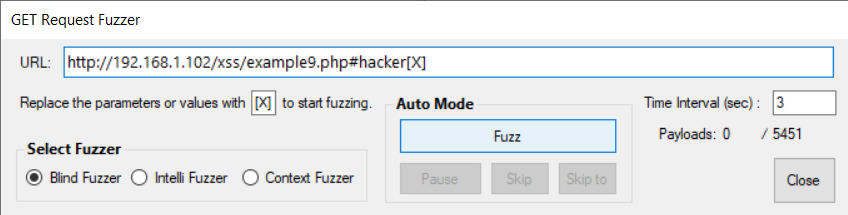
-İşlem kaynak koduna eklenmediği için zafiyet açığı olarak gözükmüyor.Bu sebeple ya manuel olarak tek tek işlem yapılacak ya da yarı otomatik olan XENOTIX ile devam edilecek.



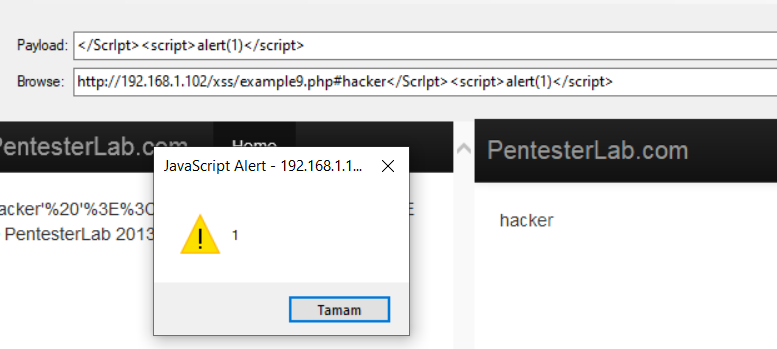
-XENOTIX aracını çalıştırıyorum ve server'ı başlatıyorum.



-Ardından scanner bölümüne geçiş yapıyorum ve “get request fuzzer” ile devam ediyorum.



-Ardından zafiyetin bulunduğu sitenin URL'ini kopyalıyorum ve sonuna (X) koyup işlemi başlatıyorum.



-DOM XSS için XETONIX 1 kere işlem yapıyor ardından 1 kerede ben REFRESH butonu ile yeniden başlatıyorum anca o zaman payloadlar çalışıyor.

-Bu zaafiyet diğerlerinden çok farklı bir zaafiyettir.Bu zaafiyet nesneye yönelik bir zaafiyettir ve site sürekli yenilendiği için kaynak kodu ile oynama yapılabilir.

-Çok fazla yaygın bir zaafiyet türü değildir.

**Korunma Yöntemleri**

-Belirli bir bölge hariç HTML kodunun içerisine konulmamalı.

-Kullanıcıdan alınan girdiler ve ekrana bastırılacak çıktılar kontrol edilmeli.

-Kullanıcıdan gelen girdiler aynı haliyle ekrana bastırılmamalıdır.

-XSS için kullanılan özel karakterler ve kelimeler filtrelenmelidir.

-Kara listede ki elemanlar encoding yöntemleri gibi yöntemler kullanılarak atlatılabileceği için,filtreleme yaparken kara iste yerine beyaz liste kullanılmalıdır.

-Hazır taslaklar kullanılması yarar sağlayacaktır.

-Olası saldırı simülasyonları ile testler yapılmalıdır.

-Cookie değerlerinin http-only olarak set edilmesi gerekmektedir.

-IDS/IPS veya WAF kullanılmalıdır.