SİBER GÜVENLİK
- HACKING - ATÖLYESİ

Onur AKTAŞ

Misafir Yazarlar
Bülent ARSAL
Korhan GÜRLER
Mahmut Esat YILDIRIM

Gazi Kitabevi

İÇİNDEKİLER

Bu Kitabı Nasıl Okumalıyım ?

Elinizde tuttuğunuz bu kitap altı seneden fazla bir süre boyunca gerçekleştirilen onlarca güvenlik testi sonucu en çok karşılaşılan ve teknik olarak farklı senaryolardan esinlenerek, konusunda uzman birçok kişiden görüş ve öneri alınarak, yetkin bir güvenlik uzmanı olmak isteyen kişiler için yazılmıştır. Kitabın ücretli olmasının en büyük sebeplerinin başında atölye masrafları gelmektedir. Kitaptan gelen tüm gelirler, siber güvenlik eğitimleri ve Siberkuvvet'in geliştirilmesi için harcanacaktır.

Teknik bir kitap ne kadar sürükleyici olur bilemiyorum ama elimden geldiğince okuyucuyu sıkmayan, eğlenceli ve temel-orta düzeyde bilgiler içeren bir siber güvenlik kitabı oluşturmaya çalıştım.

Kitabın ilk bölümlerinde en basit yöntemler ile gerçekleştirilebilecek, anlaşılması kolay fakat etkisi oldukça büyük siber saldırılara yer verdim. İlk bölümlerde öğrenilen temel bilgiler ilerleyen bölümlerdeki siber saldırıların yöntemlerini anlamak için oldukça önemli. Bu yüzden ilk bölümlerde anlatılan konulara yeterince hakim değilseniz gerekli ön araştırmalardan sonra kitaba başlamanızı öneririm.

Her bölüm kendi içerisinde 5 ana başlıktan oluşmaktadır. **Genel Bilgi** bölümünde saldırı yöntemini anlamak için bilmeniz gereken teknik bilgileri bulabilirsiniz. Genel olarak saldırı ile alakalı teknolojilerin kullanım yöntemlerini anlatan bu bölümü, saldırı yapacağınız sistemi anlamanız ve saldırının yöntemlerini daha iyi analiz etmeniz amacıyla ekledim.

Saldırı açıklaması bölümünde atölye ortamında bulunmayan bir sisteme yapılan örnek bir saldırı anlatıyorum. Bu bölüm örnek bir saldırının ve hedef sistemin neler olabileceği hakkında detaylı bilgiler içeriyor. Böylece atölye bölümünde kendiniz örnek yapmadan önce örnek bir saldırının nasıl yapıldığı hakkında bilgi edinebilirsiniz.

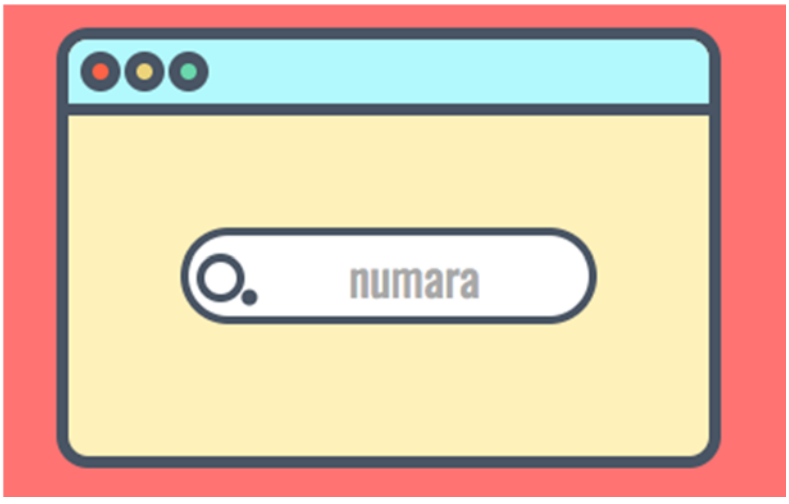
Atölye bölümü, alıştırma amaçlı yapmanız istenen örnekleri anlatan bölüm. Bu bölüm bana göre kitaptaki en önemli bölüm. Çoğu kaynak için internet üzerinde özellikle İngilizce birçok kaynak bulabilirsiniz. Fakat öğrendiklerinizi deneyebileceğiniz, alıştırma yapabileceğiniz, zafiyet

içeren Türkçe sistemlerin olduğu bir web platformu siber güvenlik uzmanı olma yolunda oldukça işinize yarayacaktır. Bu bölümün nasıl kullanılacağını ayrı bir bölümde detaylı olarak bulabilirsiniz.

Gerçek hayat hikayeleri benim ve konusunda uzman birçok profesyonel arkadaşımın konu ile ilgili görüşlerini ve gerçek senaryoları içermektedir. Bu bölümde hem saldırıları gerçekleştirirken dikkat edilmesi gereken noktalar hem de gerçek hayatta karşınıza çıkabilecek saldırı yöntemleri ile ilgili sorunlardan bahsediyorum. Güvenlik testleri gerçekleştirirken başınıza sıkıntılı durumların gelmesini önleyeceğini düşünüyorum.

Okuma Listesi bölümlerinde saldırıda kullanılan altyapıya ve/veya teknolojilere ait daha detaylı bilgiler edinmeniz için sizlere hazırladığım kaynakları ekledim. Bu bölümdekileri okumanız ve anlamanızın yetkin güvenlik uzmanı olmanız açısından çok önemli olduğunu belirtmek isterim. Koca bir bağlantıyı kitaptan okuyup da internet üzerinden erişmenin zor olduğunu tahmin ediyorum. Bu yüzden okuma listesindeki bağlantıların başında ilgili bağlantının 4 rakamlı bir numarasını yazdım. Bu numarayı “ayna.siberkuvvet.com” adresine girip yazarak ilgili bağlantıya kolayca erişebilirsiniz.

ayna.siberkuvvet.com



İleride bağlantının çalışmaması ihtimalini de göz önüne alarak aynı zamanda sitenin bir kopyasını da sisteme kayıt ettim. Eğer bağlantıda bir sorun varsa kopyası alınan siteye yine “ayna.siberkuvvet.com” üzerinden erişebilirsiniz.

Son olarak “**Cevap Verilmesi Gerekenler**” bölümünde ilgili konuyu derinlemesine anladığınızdan emin olmanız adına cevap vermeniz gereken soruları listeledim. Okuma listesini okuduktan sonra kendinizi bu sorular ile kontrol edebilirsiniz. Bu soruların cevabını Siber Kuvvet (“siberkuvvet.com”) web adresindeki soru/cevap bölümünde sorabilirisiniz.

Kitabın tamamında Türkçe kelimeler kullanmaya özen gösterdim. Okuma listesinde de Türkçe kaynaklardan yararlanmaya çalıştım. Fakat İngilizce araştırma yapmak isteyen olursa diye parantez içerisinde tanımların İngilizce karşılığına da yer verdim.

Web Application Firewall (WAF)

Web sayfaları ve uygulamaları için özel tasarlanmış, oldukça gelişmiş işlemler yapabilen cihazlardır. Bir web sayfasından dönen cevaplardaki veya web sayfasına giden isteklerdeki paketlerin, her türlü parametresi ile oynayabilme (veri ekleyen, veri çıkaran, değiştiren) gibi bir yeteneği vardır. Bu da şu demektir: kızdığı bir durum varsa onu silebilir, değiştirebilir.

Web application firewall cihazları uygulaması ve işletmesi en zor sistemlerden biridir. Çünkü bu sistemi yöneten kişilerin korudukları web sayfası ve uygulamasının nasıl çalıştığını, hangi servisleri kullandığını çok iyi bilmesi gerekmektedir. Ayrıca sayfada ya da uygulamada yapılacak güncellemelere göre hemen WAF'ta gerekli düzenlemeyi yapması elzemdir. Aksi takdirde normal kullanıcı isteklerini de engelleme ihtimali çok yüksektir. Bu nedenle WAF sistemleri engelleme modunda doğrudan kullanılmaz, monitör modda yani öğrenme modunda bir süre çalıştırılarak bilgi toplaması sağlanır, sonrasında gerekli kurallar düzenlenerek engelleme yapmaya başlar.

Öğrenme modundan sonra, yazılımın kendisinde bir zafiyet varsa bile düzgün konfigürasyonu yapılmış bir WAF sistemi ile saldırganların bu zafiyeti kullanması çok kolay bir şekilde engellenebilir. Bir kere konfigürasyonu yapıldıktan sonra bu sistemi atlatmak oldukça zordur. Bu sistemlerde OWASP'ın en güncel 10 saldırısı gibi birçok saldırı için varsayılan olarak engelleme gelmektedir. Yani siz bir ' (tırnak işareti) attığınızda WAF bunu farkedecektir.

Bununla birlikte database firewall, DNS firewall'lar da bulunmaktadır. WAF ile benzer mantıklarda çalıştığından dolayı sadece kavram olarak bilinmesi bu aşamada yeterli olacaktır.

Data Leak/Leakage Prevention (DLP)

Bu sistemler daha çok içeriden dışarıya (kendi ağınızdan dışarıdaki bir ağa) veri sızıntısının tespit edilmesi ve engellenmesi için kullanılır. Host ve network DLP olarak iki çeşidi bulunur. Saldırganlar sızdıkları bir ağdan dışarıya bir veri çıkarmak istediğinde (kredi kartı bilgisi vb.)

5.3 Gerçek Hayat Senaryoları

Yetkisiz erişimler doğru yetkilendirme kontrollerinin yapılmadığı her alanda olabilir. Güvenlik testlerinde bu alanları bulmak sizin hayal gücünüze ve testleri yaptığınız ortamdaki teknik ve iş akışına ait bilginize dayalıdır. Yapacağınız en iyi şey, testlerden önce basitçe iş akışını anlamak ve kullanılan teknolojiye hakim olmaktır. Yalnızca web uygulamalarında dahi karşınıza çok farklı yöntem ve teknolojiler çıkabilir. Örneğin yalnızca JavaScript ile yazılmış web uygulamalarında sunucudan kontrol edilmiş değerlerin, istemciye gönderilip sonrasında tekrar sunucuya gönderilerek işleme alındığına birkaç kez şahit oldum. Sayıları artırılabilir bu örneklerin tespiti için bol bol okumanız, sistemlerin çalışma mantığına hakim olmanız ve denemekten çekinmemeniz büyük önem arz ediyor.

Çoğu otomatik zafiyet tarama yazılımları yetkisiz erişimleri doğrulayamaz. Bazı durumlarda doğrulamak için sistemlere müdahale etmeniz gerekebilir. Yönetici kullanıcısının parolasını değiştirdiğimizi hatırlayın. Eğer aktif çalışan bir sistemde güvenlik testi yapıyorsanız, başınız biraz ağrıyabilir. Öncesinde sizinle birlikte çalışan müşterinin size atadığı güvenlik testi sorumlusu ile iletişime geçip, olası bir problemde sistemlerin hızlı bir şekilde eski ayarlarına geri döndürüleceğine dair garanti almanız önemlidir.

Okuma Parçası: Soket Kullanarak Uzaktan Komut Çalıştırma

Bir önceki okuma parçasında uzaktan komut çalıştırmak için iki yöntemden bahsetmiştim. Her iki yöntemde de saldırgan tarafında veya hedef sistem üzerinde sömürülen servisin dışında bir port açılması gereklidir.

Uzaktaki hedef sisteme halihazırda bir port üzerinden erişebiliyorsak aynı portu kullanarak bağlantı açmaya çalışabiliriz. Bunu yaparken iki amacımız var: birincisi tespit edilmesini zorlaştırıyoruz, ikincisi ise yapması zevkli.

Örneğimizde TCP üzerinden işlemler yapacağız.

İşlemleri yapmadan öğrenmemiz gereken şeylerden birisi soketler. Linux'da soketler bir bağlantının en son noktasıdır. Linux'da açılan soketler aynı zamanda açılan soketi adresleyen bir dosya tanımlayıcısı (file descriptor) oluştururlar.

İkincisi ise dosya tanımlayıcıları. Dosya tanımlayıcıları Linux işletim sisteminde bir dosyaya ya da giriş veya çıkış işlemlerinin kaynağına erişmek için kullanılan işlemlere özgü oluşturulan dosyalardır.

Linux'da dosya tanımlayıcılarının oluşması için yapılması gereken işlemlerin içerisinde dosya işlemleri ve soket açma da var. Fakat yalnızca bir işlem başlatmak da bazı dosya tanımlayıcılarını oluşturur.

Örnek C kodu

Fd_test_1.c

```
#include <stdio.h>
int main (int argc, char *argv[])
{
 char in;
 scanf("%c",&in);
 return 0;
}
```

6.4 Atölye

Yarış durumunu oluşturmak için istek yapıldıkça bir dosya oluşturup hemen silen bir web sayfası hazırladık. Görevin - eğer kabul edersen - web sayfasının oluşturduğu dosya adını ve dizin bilgilerini tespit edip silinmeden önce dosyayı okumak. Gerekli bilgiler siberkuvvet.com sayfasında mevcut.

Burada kitaptan duman çıkartıp yazıları silseydik havalı olurdu.

6.5 Okuma Listesi

Bu bölümdeki bağlantılara ayna.siberkuvvet.com üzerinden bağlantı kodlarını (başlarındaki dört haneli rakamlar) kullanarak kolayca ulaşabilirsiniz.

1046 - *https://www.mehmetince.net/web-uygulamalarinda-race-condition-zafiyeti-ve-etkileri/*

Race condition nedir ne değildir öğrenmek ve web uygulamalarında örneklerle açıklamasını okumak için tavsiye ederim.

1047 - *http://members.comu.edu.tr/msahin/courses/isletim_sistemi_giris/ders04.pdf*

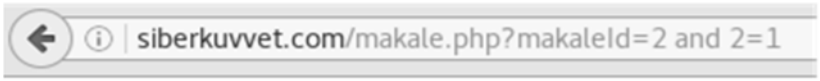
İşletim sistemlerinde yarış durumlarını yazının bir bölümünde bulabilirsiniz.

1048 - *http://oguzkartal.net/blog/index.php/2015/11/18/concurrent-eszamanli-programlama-ve-race-condition-tehlikesi/*

Yazılım geliştirirken ortaya çıkan problemler ile ilgili örneği bu blog yazısından okuyabilirsiniz.

makale.php?makaleId=2 and 2=1

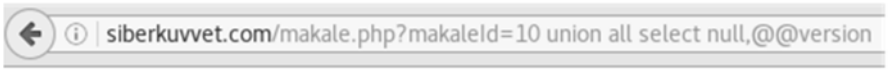
select BASLIK,YAZI from makalelerdiyebirtablo where id=2 and 2=1



Zafiyeti doğruladık. Sistemde SQL Injection zafiyeti var. Bilgi çekmeye başlayalım, öncelikle mysql sürümünü alalım. Bunun için iki sorguyu birleştiren union sorgusunu kullanacağız. İki sorguyu birleştirdiğimizde ilk sorgunun boş olmasını sağlayacağız, böylece ikinci sorgudan gelenleri ekranda gösterebileceğiz.

makale?makaleId=10 union all select null,@@version

select BASLIK,YAZI from makalelerdiyebirtablo where id=10 union all select null,@@version



Bu sorgunu çalışması için ilk sorgudaki sütun isimlerinin sayısını doğru tahmin etmemiz lazım, görüldüğü gibi 2 değil. 3 adet sorguyu deneyelim.

makale.php?makaleId=10 union all select null,null,@@version

select BASLIK,YAZI,ID from makalelerdiyebirtablo where id=10 union all select null,null,@@version

5.6.30-1

Substring fonskiyonu ise bir string değerinin yalnızca belirli bölümlerini almaya yarar. Bu fonksiyon üç parametre alır.

- •Bir string
- •Kaç karakter sonra string değerini bölecek
- •Son olarak kaç karakter ilerleyecek

Örneğin ilk karakterden sonra beş karakter ileri git :

```
select substring('siberkuvvet',1,5)
```

substring('siberkuvvet',1,5)
siber

Ya da 6. karakterden sonra 6 karakter ileri git:

```
select substring('siberkuvvet',6,6)
```

substring('siberkuvvet',6,6)
kuvvet

Şimdi bir SQL komutu üzeriden bildiklerimizi hatırlayalım;

Basit bir SQL sorgusu, ID değeri 1 olan veritabanı kayıtlarını uyeler tablosundan getirdi.

```
select * from uyeler where id='1'
```

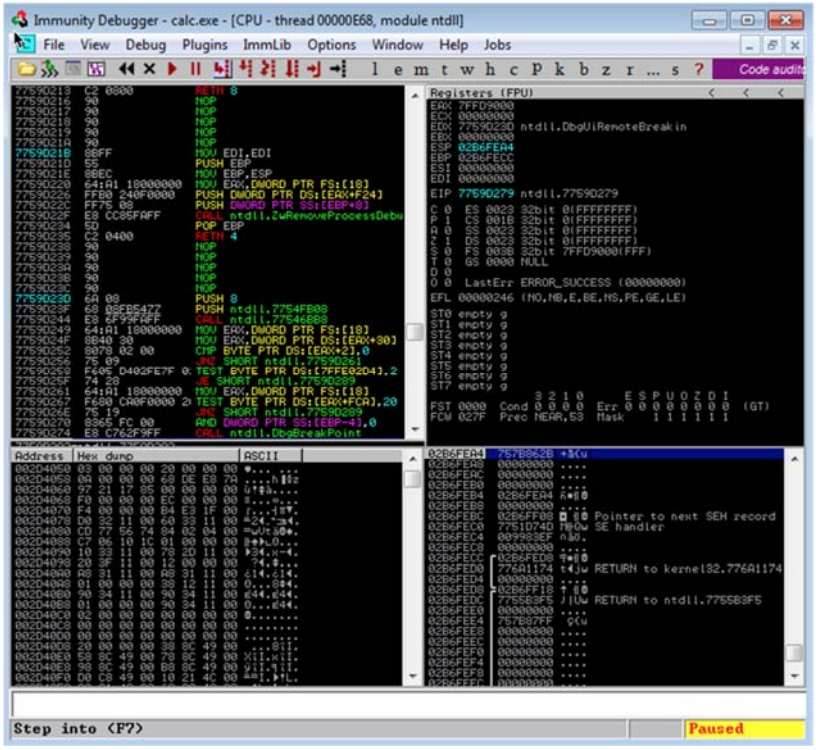
id	username	pass	email
1	admin	0	0

And koşulunda her iki tarafın da doğru olması gerekiyordu. Fakat aşağıdaki örnekte 1 değeri 2 değerine eşit olmadığından id değeri 1 olan kayıt olsa dahi hiçbir veri gelmedi.

10.4 Atölye

Atölyede bir sürpriz sizi bekliyor. Veritabanı MySQL yerine çok daha **küçük** bir **SQL** sunucusu. Hedef ise yine tablo adını çekmek. Tablo adında 2 rakam ve 2 harf var. Araştırma, kod yazma, farklı düşünme için güzel bir örnek olduğunu düşünüyorum.

Genel resim şu şekilde:



Henüz temel sorulara cevap vermedik. Sıradaki sorumuz şu: işlemci bir sonraki işlemin ne olacağını nereden biliyor?

İşlemci içerisinde *EIP* adı verilen register işlemci tarafından çalıştırılacak bir sonraki instruction'ın adresini tutar. Böylece işlemci her seferinde *EIP* register'ı içerisindeki adrese (RAM'e) gidip ilgili veriyi çalıştırır. Yazılımlar çalışırken sürekli olarak başka başka fonksiyonlar çalıştığından EIP değeri gerektiğinde stack'a kayıt edilir (burası biraz karışık detaylı olarak araştırmak isterseniz calling convention, function call, call, ret komutlarını ve kavramlarını araştırabilirsiniz). Resimlere biraz detaylı bakarak bunu görebiliriz. *EIP* içerisindeki adres CPU tarafından çalıştırılacak komutların olduğu adrestir.