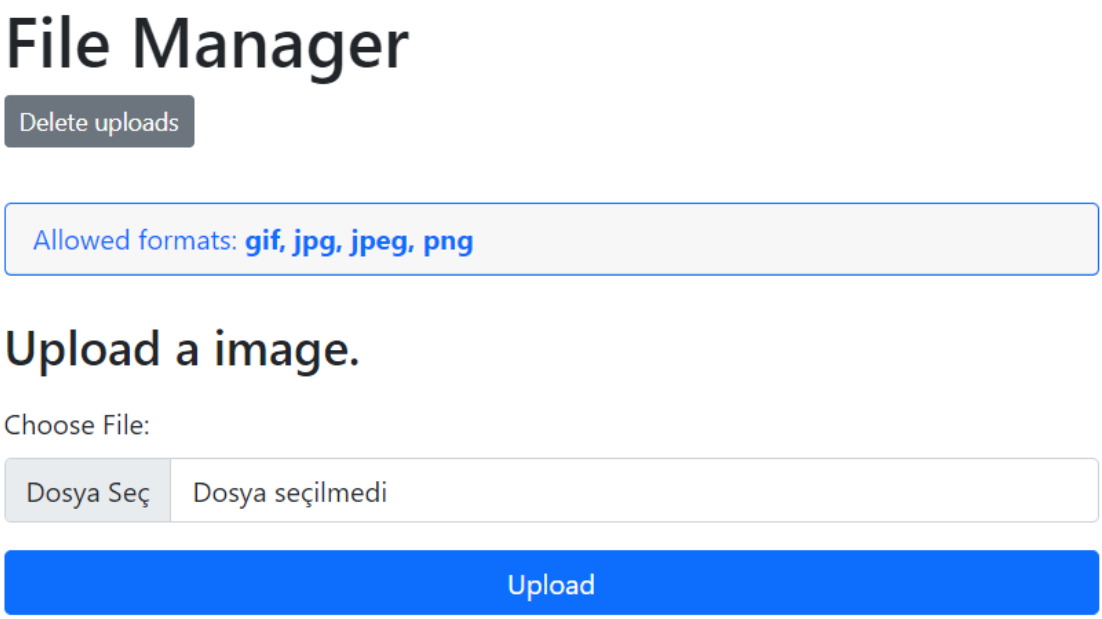
Unrestricted File Upload/ Basic Unrestricted File Upload

Herkese selamlar bu yazımda Hackviser üzerinde bulunan Web security lablarından biri olan **Unrestricted File Upload/ Basic Unrestricted File Upload** çözümünü anlatacağım.

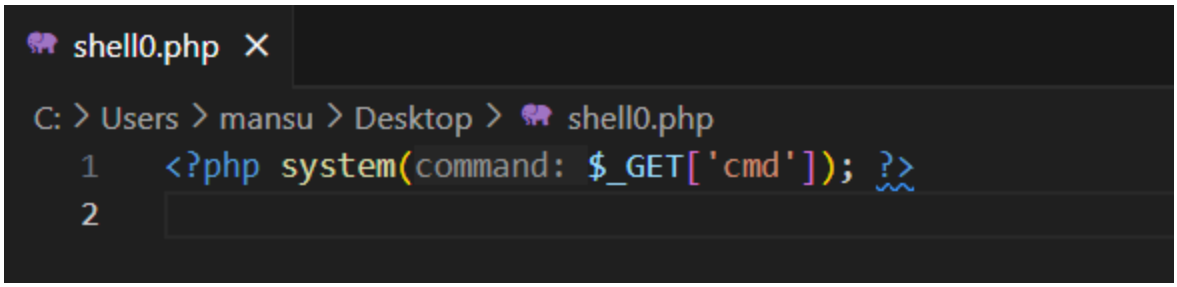
File Upload Açığı: File Upload açığı, bir web uygulamasının kullanıcıların dosya yüklemelerine izin verdiği durumlarda, saldırganların kötü niyetli dosyalar yükleyerek sunucuda zararlı eylemler gerçekleştirebileceği bir güvenlik açığıdır. Bu açık, genellikle dosya türü kontrolü, boyut sınırlamaları veya içerik taraması gibi yeterli güvenlik önlemleri alınmaması sonucunda ortaya çıkar.

Sırada için makinenin çözümü var

Öncelikle bizi aşağıdaki gibi bir sayfa karşılıyor.

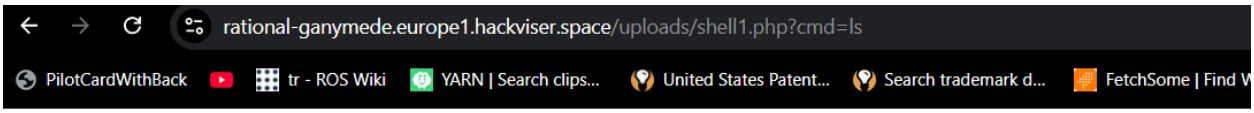


Bu kısıma istediğimiz türde dosya yükleyebiliyoruz. Benim de aklıma basit bir php shell yüklemek geliyor.



Yukarıdaki gibi basit bir shell dosyası oluşturuyorum.

ardından bu shelli sisteme yüklüyorum.

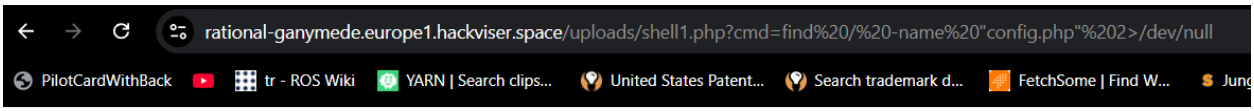


shell1.php

Başarılı bir şekilde yüklendi

Bu komut ile dosyamın bulunduğu dizini de buldum.

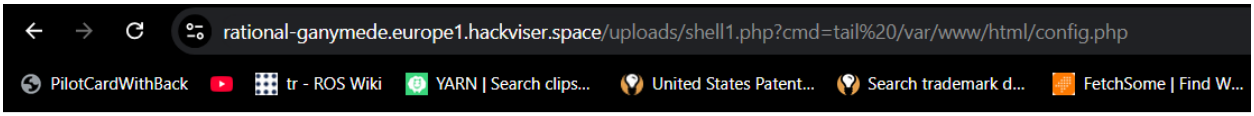
find%20/%20-name%20"config.php"%202>/dev/null



/var/www/html/config.php

Bu komut ile de dosyamızın son kısmını görüntüledik.

tail%20/var/www/html/config.php



$db_name = 'hv_database'; $charset = 'utf8'; $username = 'root'; $password = '8jv77mvXwR7LVU5v'; $db = new PDO("mysql:host=$hos

Son kısmını yazdırmamızın sebebi ise cat head more less gibi yazdırma komutları çalışmadı ama tail komutu çalıştı. yazdırabildiğimz kısımda da başarılı bir şekilde istediğimiz şifre kısmına ulaşabildik. Ayrıca url de kodlama yaptık çünkü url de boşluk vb. karakterler şifreleniyor.

File Upload Açığına Karşı Alınabilecek Önlemler:

1. **Dosya Türü Kontrolü:** Yalnızca belirli ve güvenilir dosya türlerinin (örneğin, .jpg, .png, .pdf) yüklenmesine izin verilmelidir. Sunucu, dosya uzantısını ve içeriğini kontrol ederek yüklenen dosyanın beklenen türde olduğunu doğrulamalıdır. Ayrıca, kullanıcı tarafından sağlanan dosya adlarından gelen güvenlik tehditlerine karşı önlem almak önemlidir.
2. **MIME Türü Kontrolü:** Yüklenen dosyanın MIME türü kontrol edilmelidir. Bu, dosyanın içeriğinin gerçek bir görüntü dosyası veya belge olup olmadığını belirlemeye yardımcı olur. Ancak, yalnızca MIME türüne dayanarak karar vermek yeterli değildir; dosya içeriği de kontrol edilmelidir.
3. **Dosya Boyutu Sınırlaması:** Yüklenebilecek dosyaların boyutları sınırlandırılmalı ve çok büyük dosyaların yüklenmesine izin verilmemelidir. Böylece, sunucu kaynaklarının aşırı kullanımını ve olası DoS (Hizmet Dışı Bırakma) saldırılarını önleyebilirsiniz.
4. **Dosya Yükleme Dizinine Güvenlik Önlemleri:** Yüklenen dosyaların çalıştırılabilir dosyalar olmadığına emin olmak için, yükleme dizininde çalıştırılabilir dosyaların (örneğin, .php, .exe) çalıştırılmasına izin verilmemelidir. Bu, kötü niyetli dosyaların sunucuda çalıştırılmasını önler.
5. **Geçici Dosya Kullanımı:** Kullanıcıların yüklediği dosyaları, son onaydan önce geçici bir dizine kaydetmek ve burada güvenlik kontrolleri gerçekleştirmek faydalı olabilir. Sadece kontrollerden geçen dosyalar ana dizine taşınmalıdır.
6. **İçerik Taraması:** Yüklenen dosyaların içeriği zararlı yazılımlar açısından taranmalı ve şüpheli veya zararlı içerikler tespit edilmelidir. Bu, kötü niyetli dosyaların sunucuya yüklenmesini engelleyebilir.
7. **Yükleme İşlemi için Yetkilendirme:** Yalnızca oturum açmış ve yetkili kullanıcıların dosya yüklemesine izin verilmelidir. Kullanıcıların dosya yükleme yetkileri dikkatlice kontrol edilmelidir.
8. **Loglama ve İzleme:** Dosya yükleme işlemleri loglanmalı ve izlenmelidir. Bu, olası kötü niyetli faaliyetlerin tespit edilmesine yardımcı olur ve gerektiğinde hızlı müdahale imkanı sağlar.

Örnek Senaryo

Bir web uygulamasında kullanıcılar dosya yükleyebiliyorsa, yukarıda belirtilen güvenlik önlemleri alınmazsa bir saldırgan kötü niyetli bir dosya yükleyebilir. Örneğin, bir PHP shell dosyası yükleyerek sunucuda uzaktan kontrol elde edebilir. Ancak, yukarıdaki önlemler alındığında, yalnızca belirli türde dosyaların yüklenmesine izin verildiği için saldırganın eylemleri engellenmiş olur.

Bu önlemlerle birlikte, güvenlik bilincine sahip bir geliştirme süreci, dosya yükleme açığının risklerini büyük ölçüde azaltır ve uygulamanın güvenliğini artırır.

Umarım bu yazıyı sıkılmadan okumuşsunuzdur. Keyifli çalışmalar.