

## A7 – Missing Function Level Access Control

### Nedir?

Ağa erişimi olan herhangi biri uygulamanıza istek gönderebilir. Saldırgan sadece tarayıcıdaki URL adresinde değişiklikler yaparak yetkisi olmayan sayfalara erişmeye çalışır. Örneğin URL:

<http://example.com/app/getappInfo> şeklinde olsun. Saldırgan bu adreste,

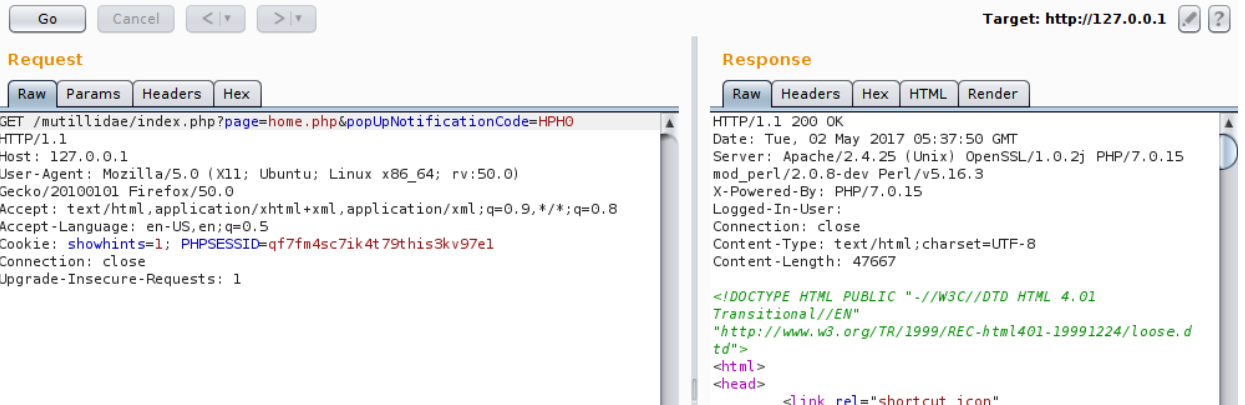
[http://example.com/app/admin\\_getappInfo](http://example.com/app/admin_getappInfo) şeklinde bir değişiklik yaparak admin panelindeki bilgilere erişebiliyorsa bu, uygulamada açıklık bulunduğu anlamın gelir.

### Uygulama

Owasp 2013 -> A7 – Missing Function Level Access Control -> "Secret" Administrative Pages sayfasında yazılanlardan yola çıkarak Mutillidae'nin gizli sayfalarının isimleri öğrenmek ve içeriklerini görüntülemek adım adım anlatılmaktadır.

URL	<a href="http://127.0.0.1/mutillidae/index.php?page=home.php&amp;popUpNotificationCode=HPHO">http://127.0.0.1/mutillidae/index.php?page=home.php&amp;popUpNotificationCode=HPHO</a>
HTTP Talep Türü	GET

1. Burp Suite kullanabilmek için tarayıcıdaki gerekli proxy ayarlamaları yapılır. Belirtilen URL'e istek gönderilir. Giden istek Burp Suite'de incelenir.
2. HTTP isteğinin görüntülediği kısma sağ tıklayarak "Send to Repeater" seçilir. Burp Suite uygulamasının Repeater tabı siteye yapacağımız istekleri ve dönecek sonuçları görmemize olanak sağlar. Sağ tıklayarak gönderdiğimiz isteği "Request" başlığı içerisinde görebiliriz. "Go" butonuna tıklayarak bu istek sonucunda dönecek cevabı "Response" altında inceleyebiliriz. Raws, Headers, Hex, HTML, tablaları istek ve cevap üzerinde farklı incelemeler yapmamıza olanak sağlar. Render sekmesi cevap olarak dönecek sayfanın arayüzünü görüntüler.



3. İstek içerisinde değişiklik yaparak olmayan bir sayfa talep edilir. "page=home.php" değeri "homeasdfg.php" gibi hata vermesi beklenen bir sayfa ismi ile değiştirilir. Go butonuna tıklanarak cevap görüntülenir. İncelendiğinde, Response – Raw altında dönen cevabın içerisinde "Validation Error: 404 - Page Not Found" şeklinde bir mesaj gözükmetedir.

```
GET /mutillidae/index.php?page=homeasdfg.php&popupNotificationCode=HPH0
HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0)
Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Cookie: showhints=1; PHPSESSID=qf7fm4sc7ik4t79this3kv97e1
Connection: close
Upgrade-Insecure-Requests: 1
```


```
<table style="margin-left:auto; margin-right:auto;">
  <tr id="id-bad-page-tr">
    <td colspan="2" class="error-message">
      Validation Error: 404 - Page Not
Found
    </td>
  </tr>
</table>
```

4. Bir önceki home.php sayfasına yaptığımız istek ile geçersiz isteği karşılaştırmak amacıyla Response alanına sağ tıklanarak "Send to Comparer" seçilir. Burada daha önce yaptığımız istekler sonucunda gönen cevapları ve cevap uzunluğunu görebiliriz. "Words" butonuna tıklayarak iki sonucu karşılaştırabiliriz.

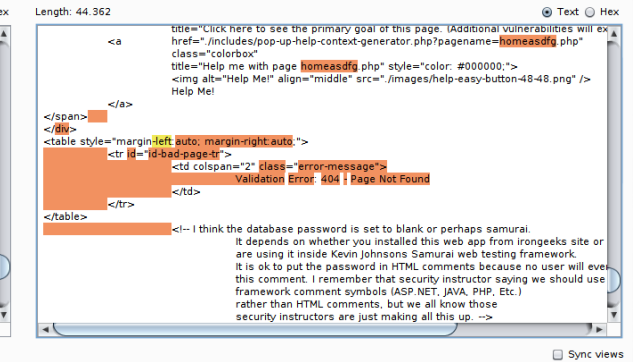
#	Length	Data
1	47936	HTTP/1.1 200 OKDate: Tue, 02 May 2017 05:37:50 G...
2	44362	HTTP/1.1 200 OKDate: Tue, 02 May 2017 05:38:22 G...

5. Words butonuna tıklanarak sonuçlar karşılaştırılır. Biri sayfa, diğeri hata döndüren iki sonuç incelenir. Turuncu ile işaretlenen kısımlar değişiklik oluşan kısımlardır. Mavi eklenen ve sarı ise silinen kısımları temsil etmektedir. Turuncu alanları detaylı incelersek daha önce gördüğümüz hata mesajına ulaşabiliriz.

Length: 47.936



Length: 44.362



6. Repeater kısmına geri dönülür. İlk yaptığımız istek açılır. Yapılan istekler arasında geçiş yapmak için < ve > butonları kullanılabilir. Home.php sayfasına yapılan isteğin üzerine sağ tıklanarak "Send to Intruder" seçilir. Intruder kısmında server detayları, HTTP isteği, payloadlar vb. alanlar bulunmaktadır. Positions tabına tıklanarak istek görüntülenir. Seçili alanlar parametrelerdir ve bu kısımlar istismar edilebilmektedir. Bu eğitimde isteğimiz gizli sayfaları bulmak olduğu için sadece sayfa ismi istismar edilecektir. Dolayısıyla tüm seçili alanları temizlemek için Clear butonuna tıklanır. Ardından *page=* devamında yer alan "home" seçilir (uzantısı dahil edilmez) ve Add butonuna tıklanır. Atak tipi olarak Sniper seçilir. (Atak

tiplerinden yazının sonunda bahsedilmektedir.)

```
Attack type: Sniper

GET /mutillidae/index.php?page=shome$$.php&popupNotificationCode=HPH0 HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Cookie: showhints=1; PHPSESSID=qf7fm4sc7ik4t79this3kv97e1
Connection: close
Upgrade-Insecure-Requests: 1
```

7. Payloads sekmesi açılır. Payload set "1", Payload type "Simple List" varsayılan olarak gelmektedir. Payload set Positions sekmesinde işaretlediğimiz alan sayısını belirtir. Bu örnekte sadece sayfa ismini işaretlediğimiz için 1 seçilidir fakat daha fazla alanı seçmiş olsaydık Payload set açılır menüsünde daha çok seçenek görebilirdik. Simple List ise örneğimizde kullanacağımız liste için yeterli olacaktır. Farklı bir örnek için diğer seçenekler incelenerek farklı liste türleri kullanılabilir. Payload Options altında Load butonuna tıklanarak istismar edilecek alan için kullanılacak liste yüklenir. Bu örnekte FuzzDB'den "WordlistSkipFish.txt" dosyası kullanılacaktır. (FuzzDB'den yazının sonunda bahsedilmektedir.)

Payload set: 1 Payload count: 1.918  
Payload type: Simple list Request count: 1.918

#### Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste .bash\_history  
Load ... .bashrc  
Remove .cvsignore  
Clear .history  
.htaccess  
.htpasswd  
.passwd  
.perf  
.ssh  
.svn  
Add Enter a new item  
Add from list ... [Pro version only]

8. Options sekmesi açılır. Grep – Match altında varsayılan olarak gelen bazı kelimeler bulunmaktadır. Bu kelimeler dönen sonuç içerisinde gelebilecek hata mesajlarıyla eşleşmektedir. Yapmakta olduğumuz örnek için halihazırda aldığımız bir hata mesajı bulunmaktadır. Dolayısıyla Clear butonuna tıklayarak pencere temizlenir. Add butonu yanındaki alana daha önce tespit ettiğimiz hata mesajı: *Validation Error: 404 - Page Not Found* yazılıp butona tıklanır. Örneğimizde aldığımız hata isminden yola çıkarak brute force

ile sayfa isimlerini bulabilmekteyiz. Ancak farklı bir örnek için listedeki varsayılan hatalar

☒ Flag result items with responses matching these expressions:

Paste Load ... Remove Clear Add

Validation Error: 404 - Page Not Found

Validation Error: 404 - Page Not Found

kullanılabilir.

9. Atağı başlatmadan önce hata mesajının başı ve sonunu Grep – Extract altında belirtmemiz gerekmektedir. Şuanda değişiklikleri yaptığımız istek doğru sonuç veren bir istek olduğu için hata mesajını bu istek içerisinde arayamayız. Hata mesajını bulabilmek ve Grep – Extract içerisine gerekli parametreleri yazabilmek için Repeater sekmesine geri dönülür. Hatalı yapılan istek bulunarak Intruder'a gönderilir. Options sekmesi açılır ve Grep – Extract altındaki Add butonuna tıklanır. Cevap içeriği görüntülenmiyorsa Refetch Response butonuna tıklanır ve arama çubuğuna 404 yazılarak cevap içerisinde hata mesajının yer aldığı kısım bulunur. Mesajın tamamı (taglar içerisindeki alan) seçilir. "Define start and end" alanında başlangıç ve bitiş kısımları otomatik oluşturulmuştur. Bu kısımlarda boşlukları temsil eden x09\x09 şeklindeki ifadeler silinebilir. Sonuçta elde edilen başlangıç ve bitiş ifadeleri kopyalanır, Intruder'da sayfa ismini işaretlediğimiz ve hata içermeyen isteğe geri dönülür ve kopyalanan alanlar ile Grep – Extract alanında Add butonu tıklanarak başlangıç/bitiş parametreleri doldurulur. Bu işlem tamamlandığında brute force atağı başlatılabilir.

☒ Define start and end

☒ Start after expression:

☐ Start at offset:

☒ End at delimiter:

☐ End at fixed length:

☐ Extract from regex group

☒ Case sensitive

☐ Exclude HTTP headers ☒ Update config based on selection below

Refetch response

```
</div>
<table style="margin-left:auto; margin-right:auto;">
  <tr id="id-bad-page-tr">
    <td colspan="2" class="error-message">
      Validation Error: 404 - Page Not Found
    </td>
  </tr>
</table>
<!-- I think the database password is set to blank or perhaps samurai.
It depends on whether you installed this web app from irongeeks site
or
are using it inside Kevin Johnsons Samurai web testing framework.
It is ok to put the password in HTML comments because no user will
```

404 1 match

10. Intruder sekmesi altında Start Attack butonuna tıklanır. İstenen sonuçlar görüntüldüğünde Attack -> Pause tıklanarak saldırı durdurulabilir. .htaccess ve .htpasswd için diğer denemelerden daha uzun sonuçlar döndüğünü görebiliriz. (normalde yaklaşık 44300 iken bu

Request	Payload	Status	Error	Timeout	Length	Valida...	-message">	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	47936	<input type="checkbox"/>		
1	.bash_history	200	<input type="checkbox"/>	<input type="checkbox"/>	44386	<input checked="" type="checkbox"/>	Validation Error: 404...	
2	.bashrc	200	<input type="checkbox"/>	<input type="checkbox"/>	44350	<input checked="" type="checkbox"/>	Validation Error: 404...	
3	.cvsignore	200	<input type="checkbox"/>	<input type="checkbox"/>	44368	<input checked="" type="checkbox"/>	Validation Error: 404...	
4	.history	200	<input type="checkbox"/>	<input type="checkbox"/>	44356	<input checked="" type="checkbox"/>	Validation Error: 404...	
5	.htaccess	200	<input type="checkbox"/>	<input type="checkbox"/>	142630	<input type="checkbox"/>		
6	.htpasswd	200	<input type="checkbox"/>	<input type="checkbox"/>	142641	<input type="checkbox"/>		
7	.passwd	200	<input type="checkbox"/>	<input type="checkbox"/>	44350	<input checked="" type="checkbox"/>	Validation Error: 404...	
8	.perf	200	<input type="checkbox"/>	<input type="checkbox"/>	44338	<input checked="" type="checkbox"/>	Validation Error: 404...	
9	.ssh	200	<input type="checkbox"/>	<input type="checkbox"/>	44332	<input checked="" type="checkbox"/>	Validation Error: 404...	
10	.svn	200	<input type="checkbox"/>	<input type="checkbox"/>	44332	<input checked="" type="checkbox"/>	Validation Error: 404...	
11	.web	200	<input type="checkbox"/>	<input type="checkbox"/>	44332	<input checked="" type="checkbox"/>	Validation Error: 404...	
12	0	200	<input type="checkbox"/>	<input type="checkbox"/>	44314	<input checked="" type="checkbox"/>	Validation Error: 404...	
13	00	200	<input type="checkbox"/>	<input type="checkbox"/>	44320	<input checked="" type="checkbox"/>	Validation Error: 404...	

- |   |           |     |                          |                          |        |                                     |
|---|-----------|-----|--------------------------|--------------------------|--------|-------------------------------------|
| 4 | .history  | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 44356  | <input checked="" type="checkbox"/> |
| 5 | .htaccess | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 142630 | <input type="checkbox"/>            |
| 6 | .htpasswd | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 142641 | <input type="checkbox"/>            |
| 7 | .passwd   | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 44350  | <input checked="" type="checkbox"/> |
| 8 | .perf     | 200 | <input type="checkbox"/> | <input type="checkbox"/> | 44338  | <input checked="" type="checkbox"/> |
- Request

Response
- Raw

Params

Headers

Hex
- ```

GET /mutillidae/index.php?page=%2ehtaccess.php&popUpNotificationCode=HPH0 HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Cookie: showhints=1; PHPSESSID=qf7fm4sc7ik4t79this3kv97e1
Connection: close
Upgrade-Insecure-Requests: 1

```

- [illegible]

Tek parametre hedef alındığında kullanılır.

### **Bettering ram**

Birden fazla parametre için kullanılır. Tüm parametreler için aynı payload kullanılır. Her parametrenin değeri aynı olacaktır.

### **Pitchfork**

Tüm parametreler için ayrı payloadlar yüklenir. Burada şöyle bir durum geçerlidir; ilk istekte birinci parametre için birinci payload listesinden birinci eleman ikinci parametre için ikinci parametre listesinden birinci eleman seçilir ve istek yapılır. İkinci istek için birinci listeden ikinci eleman, ikinci listeden ikinci eleman seçilir ve istek yapılır. Bu durum bu şekilde devam eder.

### **Cluster bomb**

Bu atak tipinde de tüm parametreler için ayrı payloadlar yüklenir. Fakat burada bir çaprazlama söz konusudur. Yani ilk istekte birinci parametre için birinci listeden birinci eleman, ikinci parametre için ikinci listeden birinci eleman seçilir ve istek yapılır. İkinci istek için ise birinci listeden birinci eleman, ikinci listeden ikinci eleman seçilir ve istek yapılır. Bu çaprazlama tüm liste boyunca devam eder.

### **FuzzDB Nedir?**

FuzzDB saldırı şablonları, tahmin edilebilir kaynak isimleri, sunucu yanıtlarını belirlemede kullanılan regex kalıpları vb. hazır liste ve dökümanları içeren açık kaynaklı bir veritabanıdır. Websheller, sık kullanılan kullanıcı adı ve şifrelerini içeren listeler ve işe yarar pek çok wordlist FuzzDB'de bulunabilir.

<https://github.com/fuzzdb-project/fuzzdb> linkinden ulaşarak zip formatında indirilebilir.

Bu örnekte kullanılan WordlistSkipFish.txt dosyası */fuzzdb-master/discovery/predictable-filepaths/filename-dirname-bruteforce/* dizini altında bulunmaktadır.