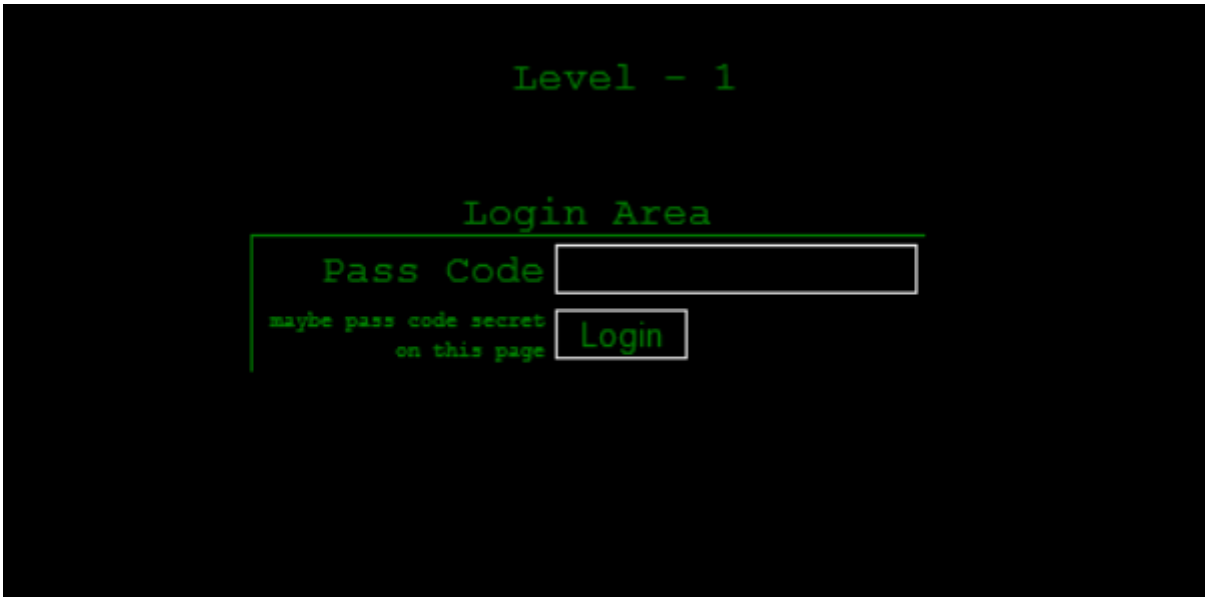
WGT Capture the Flag

Onur Yılmaz, Aralık 2009, WGT E-Dergi 3. Sayı

OWASP-Türkiye e-posta grubuna geçtiğimiz günlerde gönderilen CTF yarışmasının çözümünü aşağıda bulabilirsiniz. İki aşamadan oluşan yarışmada amaç her iki bölümdeki kimlik kontrollerini aşarak şifreye ulaşmaktı.

İlk sayfamız, sadece giriş kodu isteyen bir sayfadan oluşuyor. Aslında ipucu olarak giriş kodunun sayfanın içinde olabileceği de belirtilmiş :)



Tabi hemen sayfanın kaynak kodunu inceliyoruz. İncelememiz sonucunda aslında sayfada görünmeyen bir "password.jpg" resim dosyasının varlığını öğreniyoruz.

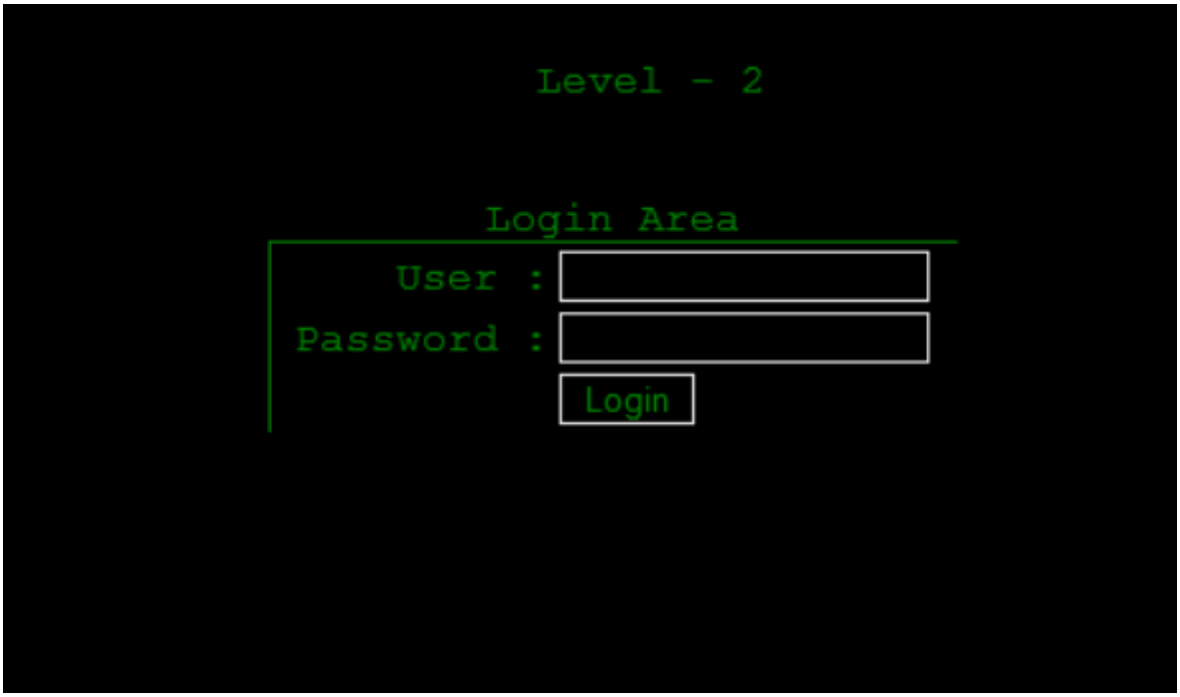
```
    <td align="right">
        <div style="font-size: 9px;">
            <div style="display: none;">
                <span id="lblPass"></span></div>
            <span id="lblInfo">
            maybe pass code secret on this page</span></div>
        <div style="display: none;">
            <img id="imgPass" src="password.jpg" style="border-width:0px;" /></div>
    </td>
    <td align="left">
```

İpuçlarını sırayla değerlendirip password.jpg dosyasını açıyoruz ama karşımıza bir bilgi içermeyen bir resim çıkıyor. Arkadaş da sanırım parolasını unutmuş :)

Resim dosyası yüklendikten sonra giriş sayfasına tekrar gidip kaynak kodunu incelediğimizde az önce görmediğimiz ve Base 64 ile kodlanmış bir değer görüyoruz.

```
<td align="right">
    <div style="font-size: 9px;">
        <div style="display: none;">
            <span id="lblPass">NTh3Z3QqKg==</span></div>
        <span id="lblInfo">
        maybe pass code secret on this page</span></div>
    <div style="display: none;">
        <img id="imgPass" src="password.jpg" style="border-width:0px;" /></div>
</td>
```

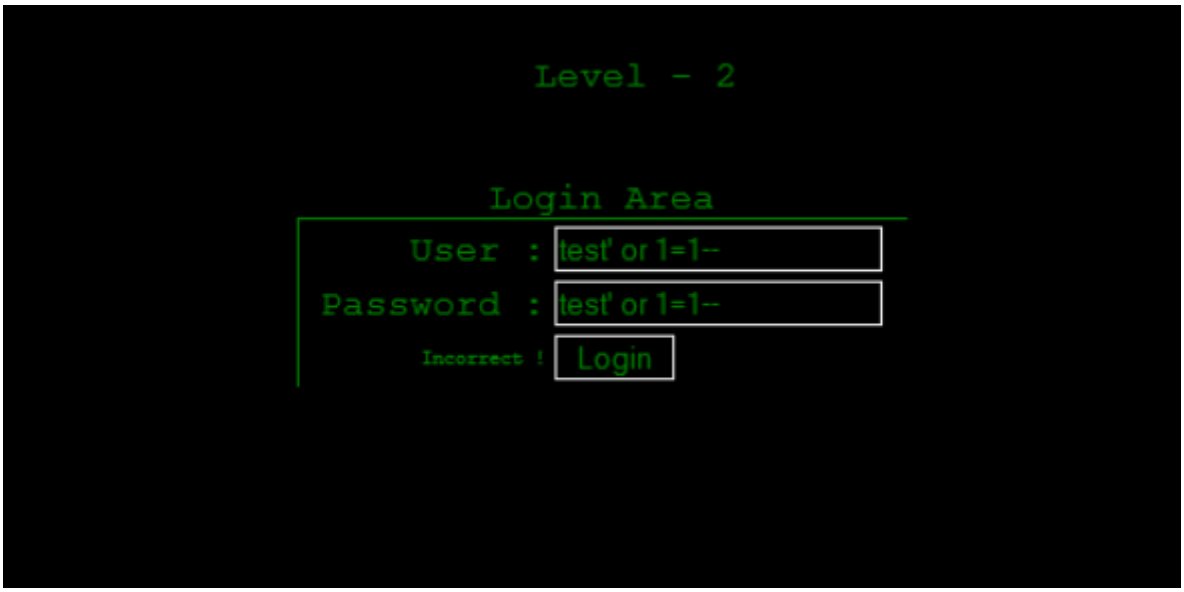
Bu değeri Base 64 kod çözücü yardımı ile çözdüğümüzde "58wgt**" değerine ulaşıyoruz ve bulmuş olduğumuz bu değeri birinci aşamada kullanarak 2. aşamaya geçiyoruz.



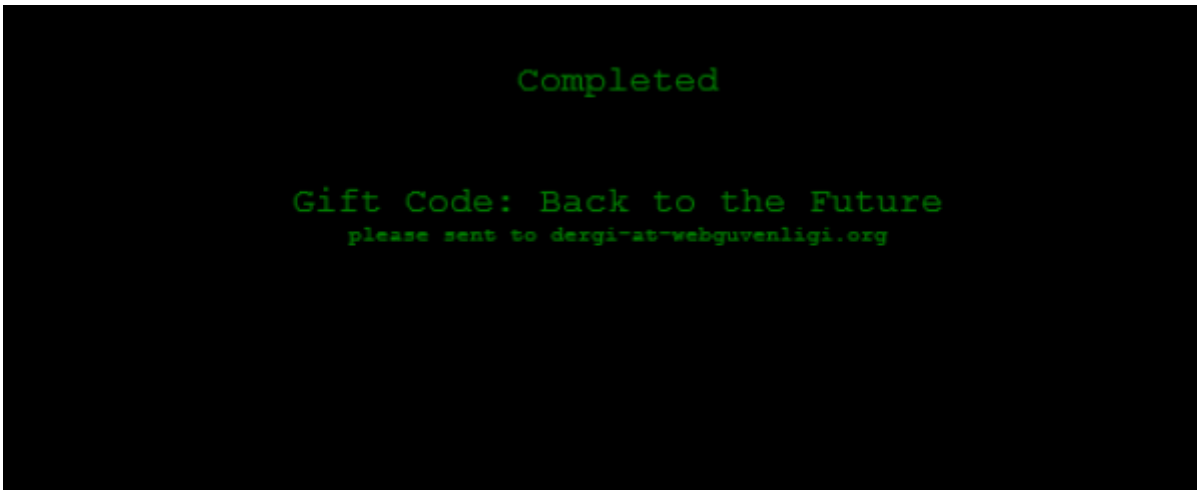
İkinci aşamada bu defa hem kullanıcı adı hem de parola soran bir ekran ile karşı karşıyayız. Sayfanın kaynak kodunu incelediğimizde istemci tarafında javascript ile SQL Injection'a karşı alınmış bir güvenlik önlemi olduğunu görüyoruz.

```
<script language="javascript" type="text/javascript">
function quoteCheck()
{
    var text = document.getElementById("txtUser").value;
    var pattern = /'/;
    if(text.match(pattern))
    {
        document.getElementById("lblState").innerHTML = "Incorrect !";
        return false;
    }
    else
    {
        document.getElementById("form1").submit()
        return true;
    }
}
</script>
```

Bu kontrol istemci tarafında yapıldığı için bunu aşmamız aslında hiç de zor değil. Bunu aşmak için en basit olarak Firefox eklentisi olan "Noscript" kullanabiliriz. Böylece bu javascript çalışmayacak ve güvenlik önlemini aşmış olacağız. Ancak Noscript kullanarak yapmış olduğumuz SQL Injection denemesinde aşağıdaki gibi "Incorrect" hatası alıyoruz.



Biraz daha detaylı inceleme için Proxy yardımı ile gelen ve geden verileri incelediğimizde kullanılan Çerez içerisindeki "login" değerinin "false" olarak gittiğini görüyoruz. Yine Firefox eklentisi olan "Tamper Data" yardımı ile SQL Injection komutlarımıza ek olarak Çerez içerisindeki login değerini "true" olarak değiştirdiğimizde hedefimiz olan şifreye ulaşıyoruz:



İstemci tarafında alınan güvenlik önlemleri oldukça kolay bir şekilde aşılabileceği için yeterli değildir. Sunucu performansını artırmak amacıyla istemci tarafında birtakım kontrollerin yapılabilir ancak asıl önlemlerin sunucu tarafında da alınması gereklidir.

Onur Yılmaz arkadaşımıza bu eğlenceli yarışma için teşekkür ederiz.