ÖN BİLGİ

Bu belge

- https://www.slideshare.net/slideshow/web-uygulama-pentest-egitimi/31682503

resmi adresindeki veya linkin kırık olması ihtimaline karşın alternatif olarak

- https://www.includekarabuk.com/kitaplik/indirmeDeposu/Siber_Guvenlik_Teknik_Makaleler/Teori/BaskalarinaAitMakaleler/Web%20Uygulama%20Pentest%20E%C4%9Fitimi.pdf

adresindeki makaleye çalışılarak elde edilen notlarımı kapsamaktadır. Bu çıkarılan notlar belgemde alıntılar ve/veya kişisel ilavelerim mevcuttur.

1)

Developers should audit all inputs which comes from user and **also from web app itself**, such as malicious parameter values instead of their default values! (Bu çıkarımı Tezde ben yaptım. Bu dökümanla alakası yok. İyi bir çıkarım diye buraya not düşüyorum.)

2)

Birçok saldırı çeşidinin temelinde yetersiz girdi kontrolü yatmaktadır:

- XSS
- SQL Injection
- File Inclusion
- Command Injection
- …

3)

Girdi Denetim Methodları

- Beyaz Liste Girdi Denetimi
    - Beyaz listede yer alan kelimeleri ya da desene uyanları kabul et, diğerlerini kabul etme yaklaşımına dayanır. Bu yöntem güvenli ve tavsiye edilen bir girdi denetim yöntemidir. Garantici bir yapıya sahiptir.
- Kara Liste Girdi Denetimi
    - Kara listede yer alan kelimeleri ya da desene uyanları kabul etme, diğerlerini kabul et yaklaşımına dayanır. Bu yöntem tavsiye edilmez. Çünkü yeterince iyi düşünülmemiş bir kara liste sonucu güvenlik bypass edilebilir.

• Sanitize (Arındırma)

- Beyaz Liste mantığı ile sanitize
  Örn; telefon numarası içinde geçen sadece sayı olanların ayıklanması, gerisinin silinmesi

- Kara liste mantığı ile sanitize
  Örn; adres girdisinin içindeki bütün tek tırnak karakterlerinin ayıklanması

• Encoding

- Girdi içindeki özel anlama sahip karakterlerin başka bir formata değiştirilmesi işlemine denir. Amaç özel anlama sahip karakterleri etkisizleştirmektir.
  Örnek;
  HTML Entities (e.g. < to <)
  URL Encoding

• Escape

- Özel anlama sahip karakterleri özel anlamı dışında kullanabilmek için bir öncesine escape karakteri konulmasına denir. Mesela SQL sorgularında WHERE cümleciğinden sonra gelecek kolon adı değerine tek tırnak koymak için backslash kullanılır.

  Select * from table1 WHERE column1 = ‘Hasan\’ın’;

(page 5-9)

4)

Cross Site Scripting – XSS

• Html/dhtml/css veya javascript kodunun izinsiz olarak kurbanın tarayıcısında çalıştırılmasıdır.

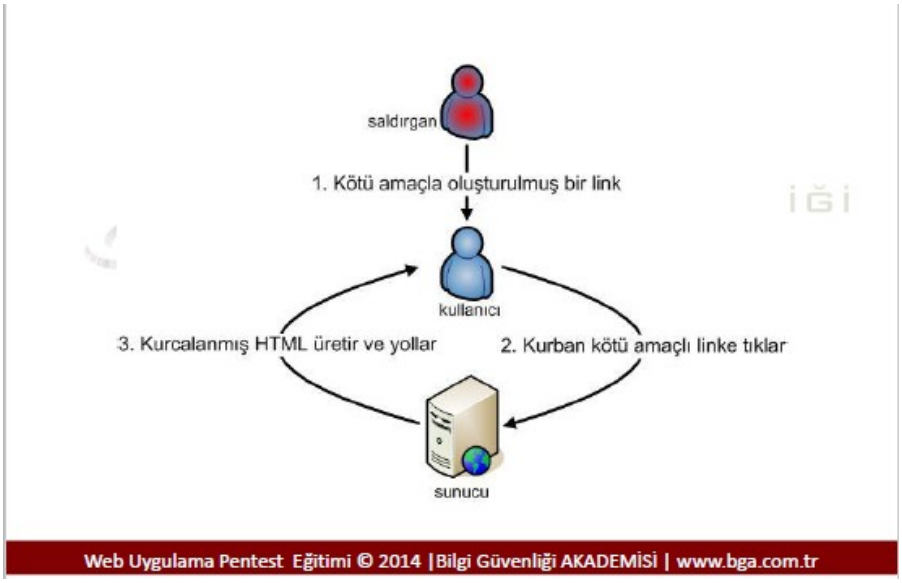
• Üç genel XSS çeşidi mevcuttur:

- Reflected
- Stored
- DOM Based

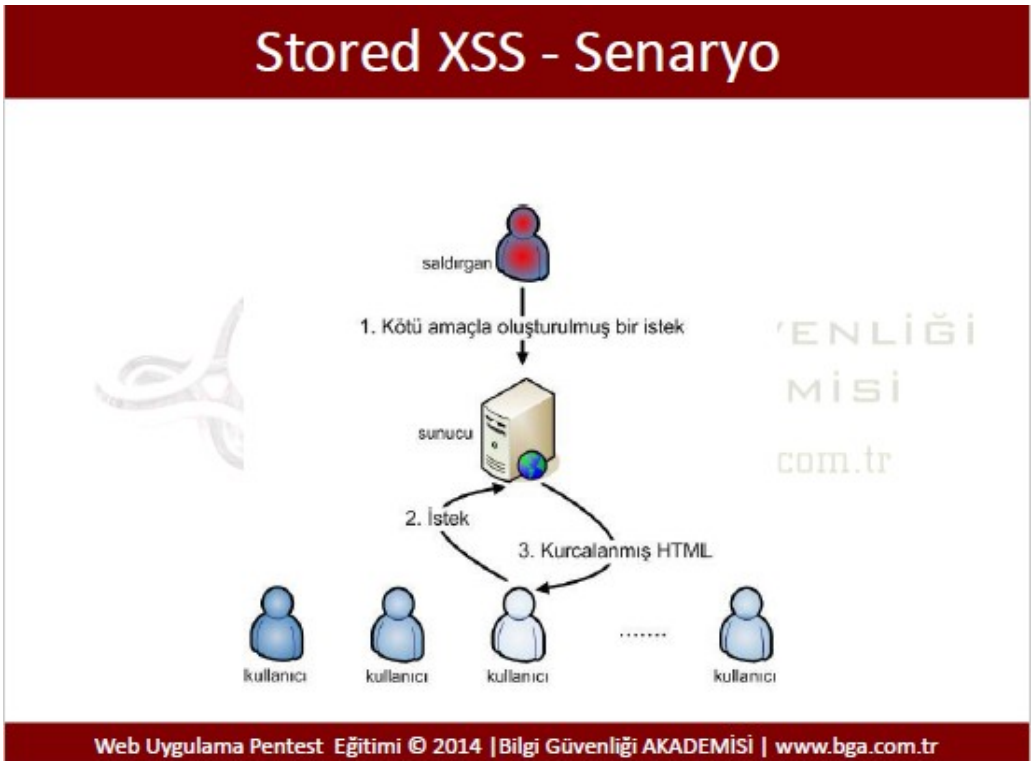
(page 15)

5)

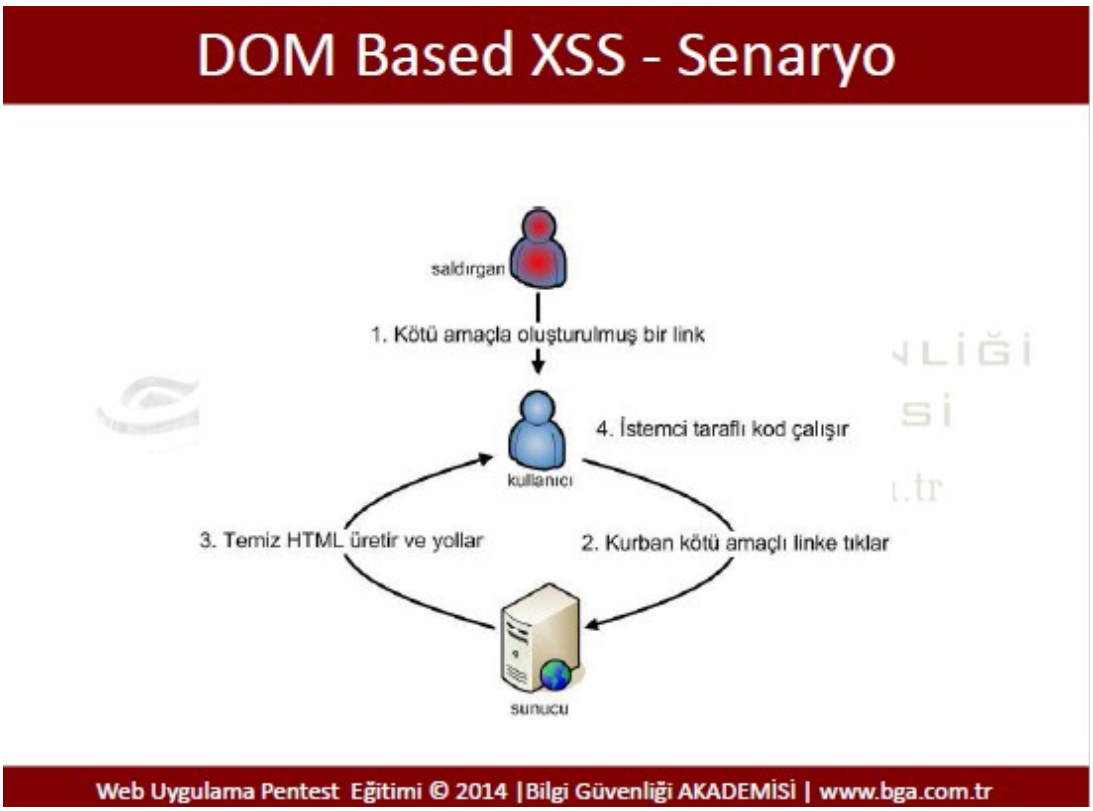
Reflected XSS



Stored XSS



DOM Based XSS



(page 16-18)

6)

XSS’in Neden Olduğu Problemler

- • Bilgi Hırsızlığı
  - Oturum Çalma
  - Clipboard Veri Çalma, Tuş Yakalama, Ekran Görüntüsü alma
- • İçerik Değişikliği (Defacement)
- • Geçmiş Tarama, Port Tarama
- • Dahili IP Çalma, Web Spidering, XSS Botnet
- • Zafiyet Tarama, Worm

(page 24)

7)

SQL Injection Nedir?

SQL Injection,

- Saldırganlar için en popüler,
- Geliştiriciler için en bilindik,
- İş sahipleri için en tehlikeli

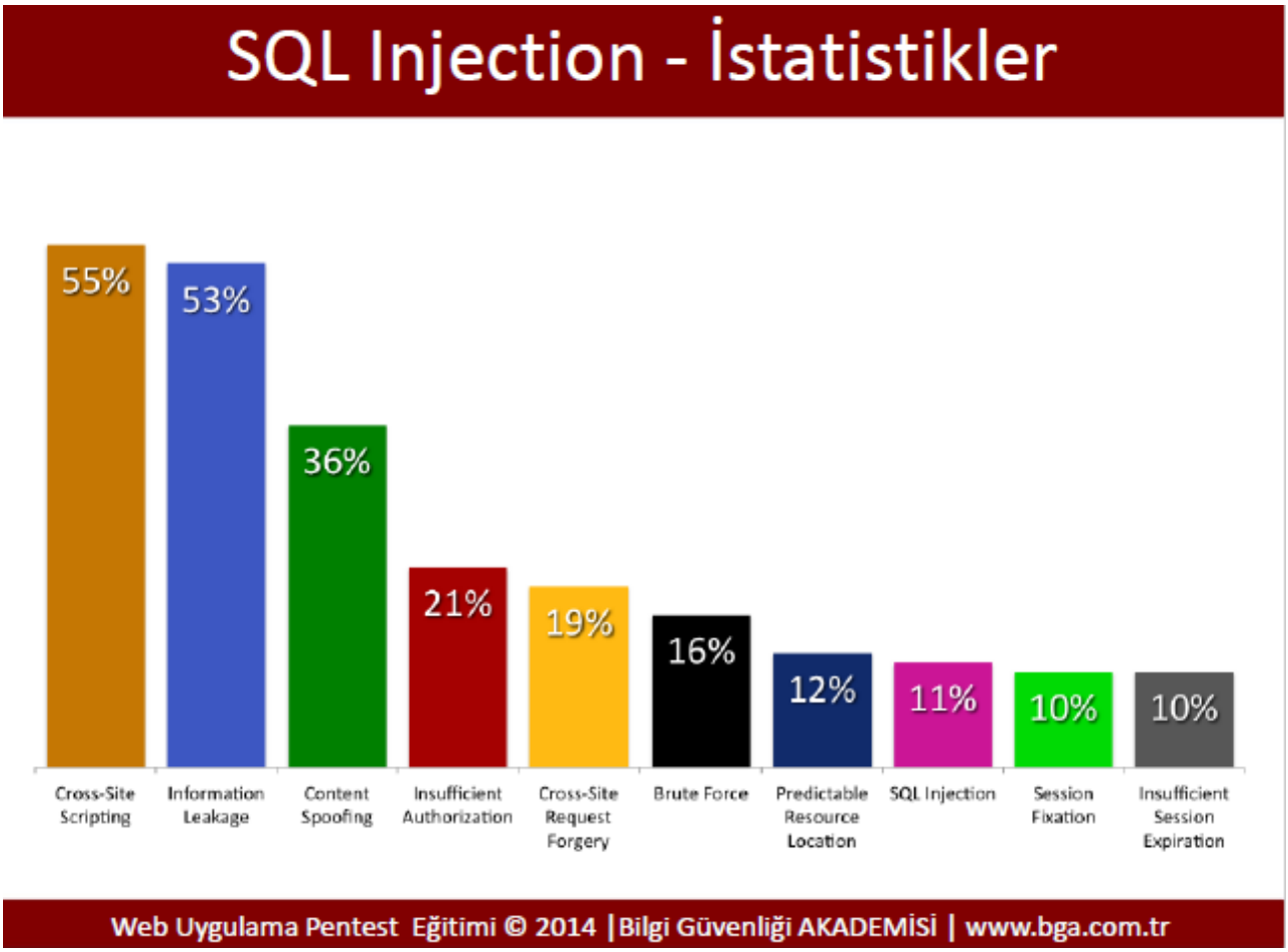
Hedef veritabanında uygulama yoluyla yetkisiz olarak sql sorgularının çalıştırılabilmesidir.

(Page 42)

8)

SQL Injection ile 2012

SQL Injection Vulnerability Used to Deface Israeli Microsoft Sites, Hacker Says



(Page 48)

9)

Kör SQL Injection ile alakalı bazı tuhaf şeyler var. Bir bak: (Page 59-62)

10)

Code Injection

Saldırganın hedef web uygulaması üzerine zararlı shell kodu eklemesidir. Büyük çoğunlukla PHP tabanlı uygulamalarda bulunsa da J2EE/ASP.NET uygulamalarında da görmek mümkündür. (Page 65)

11)

File Inclusion

Karmaşık kodları daha iyi yönetebilmek için modülariteyi arttırmak amacı ile PHP çatısında bir PHP dosyasına bir başka PHP dosyasının dahil edilmesi işlemine denir. Bu işlem için aşağıdaki fonksiyonlar kullanılır:

- include(dosya_ismi)
- require(dosya_ismi)
- include_once(dosya_ismi)
- require_once(dosya_ismi)

PHP include/require fonksiyonlarına giden parametre değerleri saldırganlar tarafından değiştirilebilirse saldırganın içeriği, çalıştırılacak PHP dosyaları içine kopyalanıp saldırgana cevap olarak dönebilir.

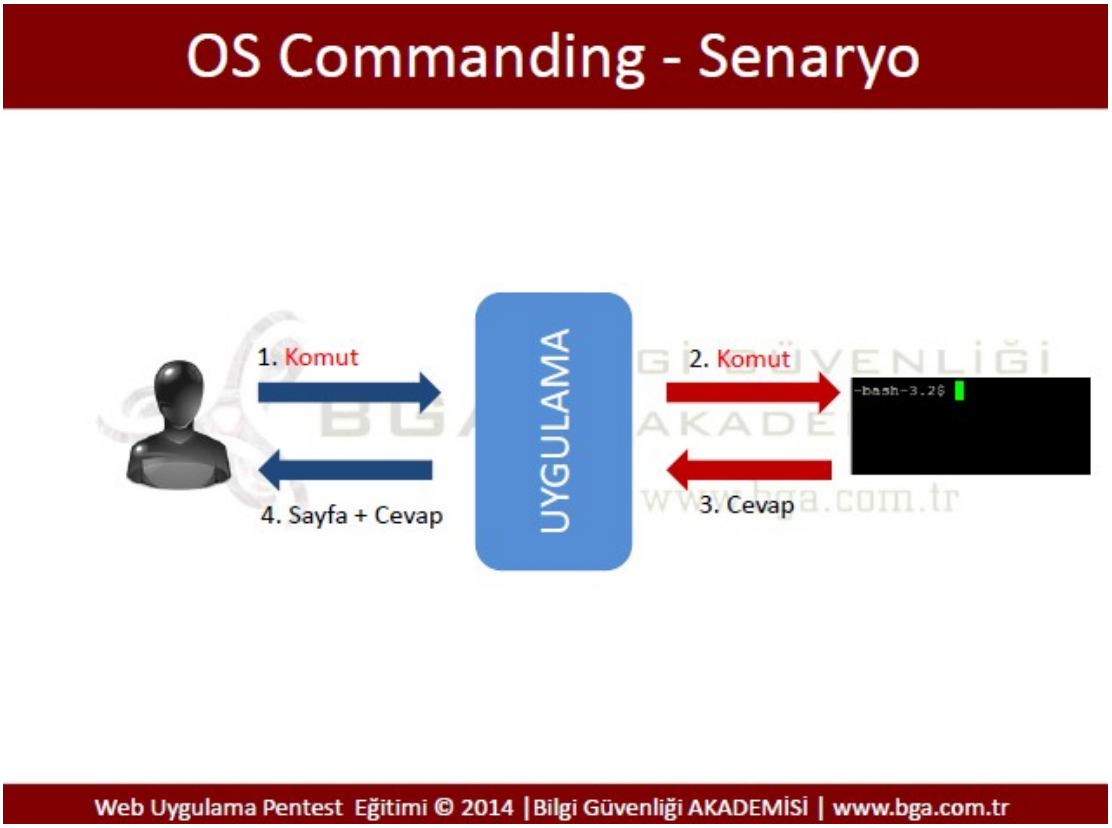
(Page 66-69)

12)

PHP uygulamalarında otomatik RFI ve LFI açığını bulabilen Python dilinde yazılmış bir tool: Fimap. Detaylarına şuradan bakabilirsin: (Page 75-77)

13)

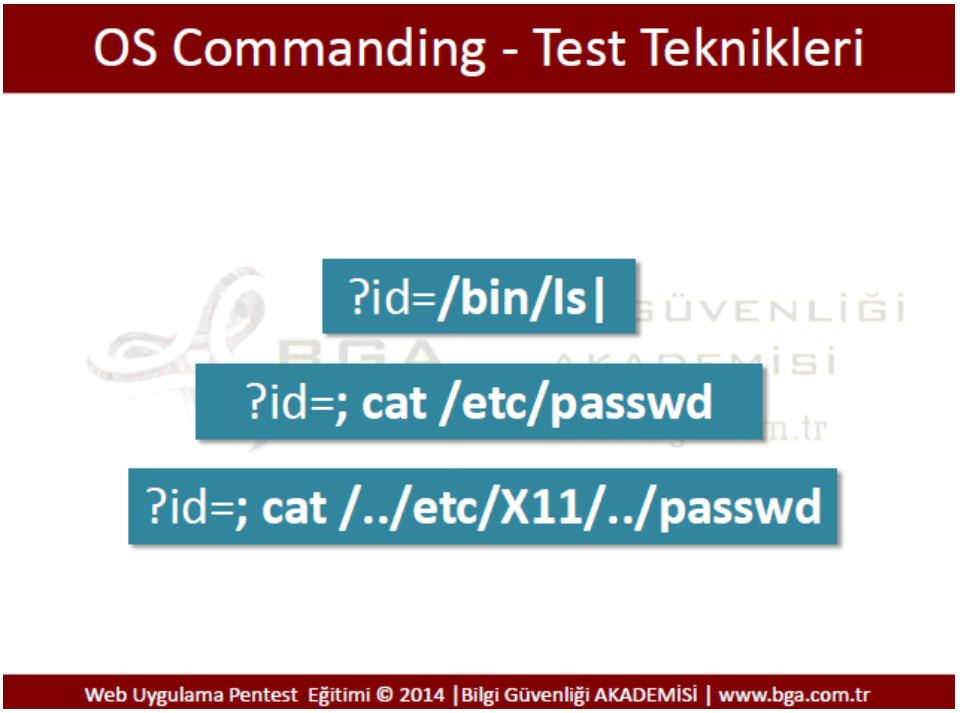
Command Execution

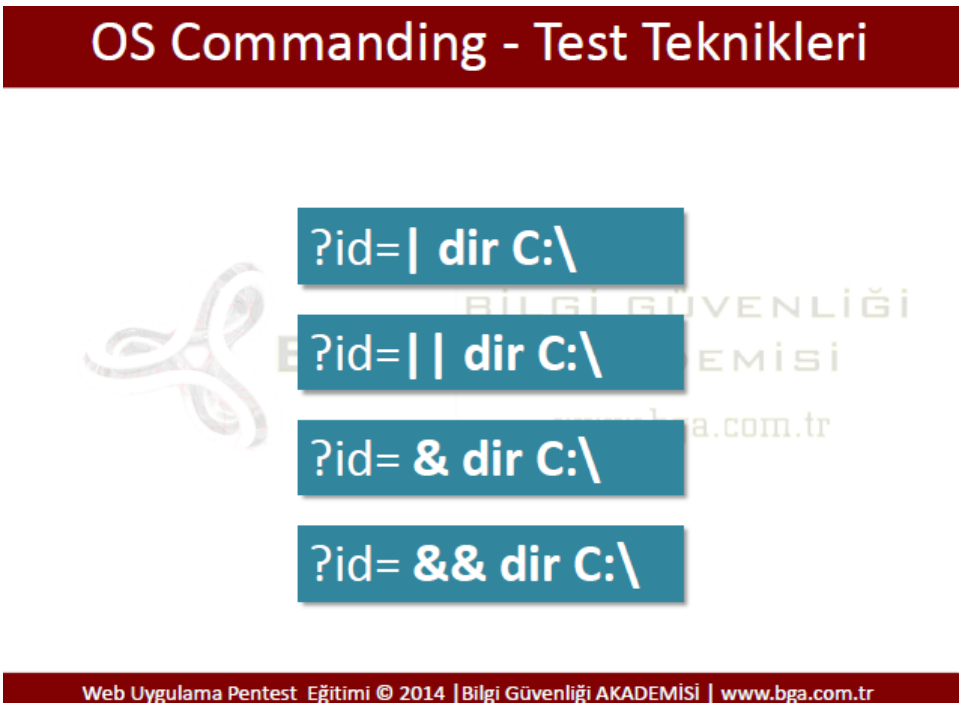


(page 94)

14)

Command Execution Teknikleri





(page 100-101)