Fingerprint Web Server (Parmak İzi Web Sunucusu)

Summary (Özet)

Web sunucusu parmak izi, bir hedefin üzerinde çalıştığı web sunucusunun türünü ve sürümünü belirleme görevidir. Web sunucusu parmak izi genellikle otomatik test araçlarında kapsüllenirken, araştırmacıların bu araçların yazılımı nasıl tanımlamaya çalıştığının temellerini ve bunun neden yararlı olduğunu anlamaları önemlidir.

Bir uygulamanın çalıştığı web sunucusu türünü doğru bir şekilde keşfetmek, güvenlik testçilerinin uygulamanın saldırıya karşı savunmasız olup olmadığını belirlemesini sağlayabilir. Özellikle, güncel güvenlik yamaları olmadan yazılımın eski sürümlerini çalıştıran sunucular, bilinen sürüme özgü istismarlara duyarlı olabilir.

Test Objectives (Test Hedefleri)

- Bilinen herhangi bir güvenlik açığının daha fazla keşfedilmesini sağlamak için çalışan bir web sunucusunun sürümünü ve türünü belirleyin.

How to Test (Nasıl Test Edilir)

Web sunucusu parmak izi için kullanılan teknikler arasında afiş kapma, kötü biçimlendirilmiş isteklere yanıtlar çıkarmak ve taktik kombinasyonunu kullanan daha sağlam taramalar yapmak için otomatik araçlar kullanmak yer alıyor. Tüm bu tekniklerin işlettiği temel önerme aynıdır. Hepsi, daha sonra bilinen yanıtların ve davranışların bir veritabanıyla karşılaştırılabilen ve böylece bilinen bir sunucu türüyle eşleştirilebilen web sunucusundan bazı yanıtlar elde etmeye çalışırlar.

Banner Grabbing (Banner Kapma)

Web sunucusuna HTTP isteği göndererek ve yanıt başlığını inceleyerek bir banner kapma gerçekleştirilir. Bu, dahil olmak üzere çeşitli araçlar kullanılarak gerçekleştirilebilir. `telnet` HTTP istekleri için veya `openssl` SSL üzerinden istekler için.

Örneğin, bir Apache sunucusundan gelen bir isteğin cevabı burada.

```
HTTP/1.1 200 OK
Date: Thu, 05 Sep 2019 17:42:39 GMT
Server: Apache/2.4.41 (Unix)
Last-Modified: Thu, 05 Sep 2019 17:40:42 GMT
ETag: "75-591d1d21b6167"
Accept-Ranges: bytes
Content-Length: 117
Connection: close
Content-Type: text/html
...
```

İşte başka bir cevap, bu kez nginx'ten.

```
HTTP/1.1 200 OK
Server: nginx/1.17.3
Date: Thu, 05 Sep 2019 17:50:24 GMT
Content-Type: text/html
Content-Length: 117
Last-Modified: Thu, 05 Sep 2019 17:40:42 GMT
Connection: close
ETag: "5d71489a-75"
Accept-Ranges: bytes
...
```

İşte Lightpd'den bir tepkinin neye benzediği.

```
HTTP/1.0 200 OK
Content-Type: text/html
Accept-Ranges: bytes
ETag: "4192788355"
Last-Modified: Thu, 05 Sep 2019 17:40:42 GMT
Content-Length: 117
Connection: close
```

```
Date: Thu, 05 Sep 2019 17:57:57 GMT
Server: lighttpd/1.4.54
```

Bu örneklerde sunucu tipi ve sürümü açıkça açığa çıkar. Bununla birlikte, güvenlik bilincine sahip uygulamalar, başlığı değiştirerek sunucu bilgilerini gizleyebilir. Örneğin, değiştirilmiş bir başlık içeren bir site talebine verilen yanıttan bir alıntı:

```
HTTP/1.1 200 OK
Server: Website.com
Date: Thu, 05 Sep 2019 17:57:06 GMT
Content-Type: text/html; charset=utf-8
Status: 200 OK
...
```

Sunucu bilgilerinin gizlendiği durumlarda, testçiler başlık alanlarının siparişine göre sunucu türünü tahmin edebilir. Yukarıdaki Apache örneğinde, alanların bu sırayı takip ettiğini unutmayın:

- Tarih
- Sunucu
- Son Modifiye
- ETag
- Kabul-Rhangişler
- İçerik Boynu
- Bağlantı
- İçerik-Tip

Bununla birlikte, hem ninx hem de belirsiz sunucu örneklerinde, ortak alanlar bu sırayı takip eder:

- Sunucu
- Tarih
- İçerik-Tip

Testçiler bu bilgileri, belirsiz sunucunun nginx olduğunu tahmin etmek için kullanabilirler. Bununla birlikte, bir dizi farklı web sunucusunun aynı alan siparişini paylaşabileceğini ve alanların değiştirilebileceğini veya kaldırılabileceğini göz önünde bulundurarak, bu yöntem kesin değildir.

Sending Malformed Requests (Malforme Talepleri Göndermek)

Web sunucuları hata yanıtlarını inceleyerek ve özelleştirilmemiş durumda varsayılan hata sayfalarını inceleyerek tanımlanabilir. Bir sunucuyu bunları sunmaya zorlamanın bir yolu, kasıtlı olarak yanlış veya kötü biçimlendirilmiş istekler göndermektir.

Örneğin, mevcut olmayan yöntem talebine verilen yanıt burada `SANTA CLAUS` Apache sunucusundan.

```
GET / SANTA CLAUS/1.1


HTTP/1.1 400 Bad Request
Date: Fri, 06 Sep 2019 19:21:01 GMT
Server: Apache/2.4.41 (Unix)
Content-Length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
</body></html>
```

İşte nginx'ten gelen aynı talebe cevap.

```
GET / SANTA CLAUS/1.1



<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.17.3</center>
</body>
</html>
```

İşte aynı talebe ışıktan gelen cevap.

```
GET / SANTA CLAUS/1.1



HTTP/1.0 400 Bad Request
Content-Type: text/html
Content-Length: 345
Connection: close
Date: Sun, 08 Sep 2019 21:56:17 GMT
Server: lighttpd/1.4.54

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title>400 Bad Request</title>
 </head>
 <body>
  <h1>400 Bad Request</h1>
 </body>
</html>
```

Varsayılan hata sayfaları, web sunucuları türleri arasında birçok farklılaşma faktörü sunduğundan, incelemeleri sunucu kafalı alanları gizlendiğinde bile parmak izi için etkili bir yöntem olabilir.

Using Automated Scanning Tools (Otomatik Tarama Araçları Kullanma)

Daha önce de belirtildiği gibi, web sunucusu parmak izi genellikle otomatik tarama araçlarının işlevselliği olarak dahil edilir. Bu araçlar, yukarıda gösterilenlere benzer isteklerde bulunabilir ve diğer daha sunucuya özgü sondalar gönderebilir. Otomatik araçlar, web sunucularından gelen yanıtları manuel testten çok daha hızlı bir şekilde karşılaştırabilir ve sunucu tanımlamasını denemek için bilinen yanıtların büyük veritabanlarını kullanabilir. Bu nedenlerden dolayı, otomatik araçların doğru sonuçlar üretme olasılığı daha yüksektir.

İşte web sunucusu parmak izi işlevselliğini içeren bazı yaygın olarak kullanılan tarama araçları.

- Netcraft, web sitesi web sitesini bilgi için tarayan çevrimiçi bir araç.
- Nikto, Açık Kaynak komut satırı tarama aracı.
- Nmap, aynı zamanda bir GUI'ye sahip olan bir Açık Kaynak komut satırı aracı, Zenmap.

Remediation (Düzeltme)

Açıkta kalan sunucu bilgileri mutlaka kendi başına bir güvenlik açığı olmasa da, saldırganlara varabilecek diğer güvenlik açıklarından yararlanmada yardımcı olabilecek bilgilerdir. Açık sunucu bilgileri, saldırganların, el değmemiş sunuculardan yararlanmak için kullanılabilecek sürüme özel sunucu güvenlik açıklarını bulmasına da yönlendirebilir. Bu nedenle bazı tedbirlerin alınması tavsiye edilir. Bu eylemler şunları içerir:

- Apache'nin mod_headers modülü gibi başlıklarda web sunucusu bilgilerini gizlemek.
- Web sunucusu ile İnternet arasında ek bir güvenlik katmanı oluşturmak için sertleştirilmiş bir ters proxy sunucusu kullanın.

- Web sunucularının en son yazılım ve güvenlik yamalarıyla güncel tutulmasını sağlamak.