Testing for Weak Transport Layer Security (Zayıf Aktarım Katmanı Güvenliği Testi)

Summary (Özet)

Müşteri ile sunucu arasında bilgi gönderildiğinde, bir saldırganın okuyabilmesini veya değiştirebilmesini önlemek için şifrelenmesi ve korunması gerekir. Bu, en çok eski Güvenli Soket Katmanı (SSL) protokolünün yerini alan Ulaştırma Katmanı Güvenliği (TLS) protokolünü kullanan HTTPS kullanılarak yapılır. TLS ayrıca, sunucunun güvenilir bir dijital sertifika sunarak, doğru sunucuya doğru sunucuya bağlı olduklarını göstermesinin bir yolunu sunar.

Yıllar boyunca, SSL ve TLS protokollerinde ve kullandıkları şifrelerde tanımlanan çok sayıda kriptografik zayıflık olmuştur. Ek olarak, bu protokollerin uygulamalarının çoğunun da ciddi güvenlik
açıkları olmuştur. Bu nedenle, sitelerin sadece TLS uygulamakla kalmayıp, bunu güvenli bir şekilde yaptıklarını test etmek önemlidir.

Test Objectives (Test Hedefleri)

- Servis yapılandırmasını doğrulayın.
- Dijital sertifikanın kriptografik gücünü ve geçerliliğini gözden geçirin.
- TLS güvenliğinin baypas edilemez olduğundan ve uygulama genelinde uygun şekilde uygulandığından emin olun.

How To Test (Nasıl Test Edilir)

Ulaştırma katmanı güvenlikle ilgili konular geniş çapta aşağıdaki alanlara ayrılabilir:

Server Configuration (Sunucu Yapılandırması)

TLS tarafından desteklenen çok sayıda protokol versiyonu, şifre ve uzantı bulunmaktadır. Bunların çoğu miras olarak kabul edilir ve aşağıda listelenenler gibi kriptografik zayıflıklara sahiptir. Yeni

zayıflıkların zaman içinde tanımlanmasının muhtemel olduğunu unutmayın, bu nedenle bu liste eksik olabilir.

- SSLV2 (BUGÜN)
- SSLV3 (NOODLE)
- TLSv1.0 (BEAST)
- İHRACA şifreleri süitler (FREAK)
- NULL şifreleri (sadece kimlik doğrulama sağlarlar).
- Anonim şifreler (bunlar RFC 7672'de tartışıldığı gibi SMTP sunucularında desteklenebilir)
- RC4 şifreleri (NOMORE)
- CBC mod şifreleri (BEAST, Lucky 13)
- TLS sıkıştırma (SUÇ)
- Zayıf DHE tuşları (LOGJAM)

Mozilla Server Side TLS Kılavuzu, şu anda önerilen protokolleri ve şifreleri detaylandırır.

Exploitability (İstismar edilebilirlik)

Bu saldırıların çoğunun bir laboratuvar ortamında gösterilmiş olmasına rağmen, (genellikle aktif) bir MitM saldırısı ve önemli kaynaklara ihtiyaç duydukları için gerçek dünyada istismar etmek için
genellikle pratik olarak kabul edilmedikleri vurgulanmalıdır. Bu nedenle, ulus devletlerden başka kimse tarafından sömürülmeleri pek olası değildir.

Digital Certificates (Dijital Sertifikalar)

Cryptographic Weaknesses (Kriptografik Zayıflıklar)

Kriptografik bir perspektiften bakıldığında, dijital bir sertifikada gözden geçirilmesi gereken iki ana alan vardır:

- Anahtarlık en *az* 2048 bit olmalıdır.

- İmza algoritması *en az* SHA-256 olmalıdır. MD5 ve SHA-1 gibi yasal algoritmalar kullanılmamalıdır.

Validity (Geçerlilik)

Kriptografik olarak güvenli olmasının yanı sıra, sertifika da geçerli (veya güvenilir) olarak kabul edilmelidir. Bu, gerektiği anlamına gelir:

- Tanımlanmış geçerlilik süresi içerisinde olun.
  - 1 Eylül 2020'den sonra verilen herhangi bir sertifika, 398 günden fazla maksimum ömre sahip olmamalıdır.
- Güvenilir bir sertifika yetkilisi (CA) tarafından imzalayın.
  - Bu, dışa dönük uygulamalar için güvenilir bir kamu CA veya iç uygulamalar için dahili bir CA olmalıdır.
  - Sadece *your*sisteminiz CA'ya güvenmediği için iç başvuruları güvensiz sertifikalara sahip olarak işaretlemeyin.
- Sistemin ev sahibi ile eşleşen bir Konu Alternatif Adı (SAN) var.
  - Common Name (CN) alanı, sadece SAN'a bakan modern tarayıcılar tarafından göz ardı edilir.
  - Sisteme doğru isimle eriştiğinizden emin olun (örneğin, ana bilgisayara IP ile erişirseniz, herhangi bir sertifika güvenilmez görünür).

Wildcard alan adları için bazı sertifikalar verilebilir (örneğin `*.example.org`), birden fazla alt takım için geçerli olabilecekleri anlamına gelir. Uygun olmasına rağmen, bunun etrafında dikkate alınması gereken bir dizi güvenlik endişesi vardır. Bunlar OWASP Ulaştırma Katmanı Güvenlik Hile Sayfasında tartışılıyor.

Sertifikalar ayrıca, iç ağın bir resmini oluşturmaya veya sosyal mühendislik faaliyetlerini yürütmeye çalışırken yararlı olabilecek İhraç ve SAN alanlarındaki dahili sistemler veya alan adları hakkında bilgi de sızdırabilir.

Application Vulnerabilities (Uygulama Zafiyetleri)

Yıllar boyunca çeşitli TLS uygulamalarında güvenlik açıkları olmuştur. Burada listelenecek çok fazla var, ancak bazı önemli örnekler şunlardır:

- Debian OpenSSL Öngörülebilir Rastgele Sayı Jeneratörü (CVE-2008-0166)
- OpenSSL Güvensiz Yeniden Müjdeleme (CVE-2009-3555)
- OpenSSL Heartbleed (CVE-2014-0160)[değiştir | kaynağı değiştir]
- F5 TLS HAVASI (CVE-2014-8730)
- Microsoft Kanal Hizmet İnkarı (MS14-066 / CVE CVE-2014-6321)

Application Vulnerabilities (Uygulama Güvenlik Açıkları)

Altta yatan TLS yapılandırmasının güvenli bir şekilde yapılandırılmasının yanı sıra, uygulamanın da güvenli bir şekilde kullanması gerekir. Bu noktalardan bazıları bu kılavuzun başka
yerlerinde ele alınmıştır:

- Şifrelenmemiş kanallar üzerinden hassas veri göndermemek (WSTG-CRYP-03)
- HTTP Strict-Transport-Security başlığını ayarlamak (WSTG-CONF-07)
- Güvenli bayrağı çerezlerin üzerine ayarlanması (WSTG-SESS-02)

Mixed Active Content (Karışık Aktif İçerik)

Karma aktif içerik, aktif kaynakların (CSS'ye komut dosyaları gibi) şifrelenmemiş HTTP üzerinden yüklendiği ve güvenli (HTTPS) bir sayfaya dahil edilmesidir. Bu tehlikelidir, çünkü bir saldırganın bu dosyaları değiştirmesine izin verir (şifresiz gönderildikleri gibi), bu da sayfada rastgele kod (JavaScript veya CSS) yürütmelerine izin verebilir. Güvensiz bir bağlantının üzerine yüklenen pasif içerik (görüntüler gibi) de bilgiyi sızdırabilir veya bir saldırganın sayfayı tahrip etmesine izin verebilir, ancak tam bir uzlaşmaya yol açma olasılığı daha düşüktür.

> Not: Modern tarayıcılar, güvenli olmayan kaynaklardan güvenli sayfalara yüklenen aktif içeriği engelleyecektir.

Redirecting from HTTP to HTTPS (HTTP'den HTTPS'ye yönlendirmek)

Birçok site, şifrelenmemiş HTTP üzerinden bağlantıları kabul eder ve daha sonra kullanıcıyı derhal sitenin güvenli (HTTPS) sürümüne yönlendirir. 301 Moved

`Permanently` Yönlendirmeye yönlendirin. Sitenin HTTPS sürümü daha sonra ayarlanır `Strict-Transport-Security` Tarayıcıya gelecekte her zaman HTTPS kullanma talimatı vermek için başlık.

Bununla birlikte, bir saldırgan bu ilk talebi engelleyebilirse, kullanıcıyı kötü amaçlı bir siteye yönlendirebilir veya sonraki istekleri engellemek için slstrip gibi bir araç kullanabilirler.

Bu tür bir saldırıya karşı savunmak için sitenin ön yükleme listesine eklenmesi gerekir.

Automated Testing (Otomatik Test)

Bir hizmetin SSL / LS yapılandırmasındaki zayıflıkları tanımlamak için kullanılabilecek çok sayıda tarama aracı vardır, hem özel araçlar hem de genel amaçlı güvenlik açığı tarayıcıları da dahil olmak üzere. Daha popüler olanlardan bazıları şunlardır:

- Nmap (çeşitli senaryolar)
- OWASP O-Saft'ın
- Slscan
- sslyze
- SSL Laboratuvarları
- testlsl.sh

Manual Testing (Manuel Test)

Çoğu kontrolü manuel olarak yapmak, komut satırı görünümlerini kullanarak gerçekleştirilmesi de mümkündür. `openssl s_client` ya da `gnutls-cli` Belirli protokoller, şifreler veya seçeneklerle bağlantı kurmak.

Bunun gibi test yaparken, çoğu modern sistemle birlikte gönderilen OpenSSL veya GnutLS sürümünün SSLV2 veya EXPRop şifreleri gibi bazı modası geçmiş ve güvensiz protokolleri desteklemeyebileceğini unutmayın. Versiyonunuzun test için kullanmadan önce modası geçmiş sürümleri desteklediğinden veya yanlış negatiflerle sonuçlanacağınızdan emin olun.

Bir web tarayıcısı kullanarak sınırlı test yapmak da mümkün olabilir, çünkü modern tarayıcılar geliştirici araçlarında kullanılan protokollerin ve şifrelerin ayrıntılarını sağlayacaktır. Ayrıca, bir
sertifikanın güvenilir kabul edilip edilmediğini test etmenin, hizmete göz atarak ve size bir sertifika uyarısı sunulup sunulmadığını test etmenin kolay bir yolunu sağlarlar.

Referances (Referanslar)

- OWASP Transport Layer Protection Cheat Sheet
- Mozilla Server Side TLS Guide