Testing for HTTP Incoming Requests (HTTP Gelen İstekleri Testi)

Summary (Özet)

Bu bölüm, hem istemci tarafındaki hem de sunucu tarafındaki tüm / giden HTTP isteklerini nasıl izleyeceğinizi açıklar. Bu testin amacı, arka planda gönderilen gereksiz veya şüpheli HTTP talebi olup olmadığını doğrulamaktır.

Web güvenlik test araçlarının çoğu (yani. AppScan, BurpSuite, ZAP) HTTP Proxy olarak hareket eder. Bu, istemci tarafı uygulaması veya tarayıcıda vekalet değişiklikleri gerektirecektir. Aşağıda listelenen test teknikleri, üretim kullanım senaryosuna daha yakın olacak olan istemci tarafı değişiklikleri olmadan HTTP isteklerini nasıl izleyebileceğimize odaklanmıştır.

Test Objectives (Test Hedefleri)

- Şüpheli istekleri incelemek için Web Sunucusuna gelen ve giden tüm HTTP isteklerini izleyin.
- Son kullanıcı Tarayıcı proxy veya istemci tarafı uygulaması değişiklikleri olmadan HTTP trafiğini izleyin.

How To Test (Nasıl Test Edilir)

Reverse Proxy (Ters Vekil)

Web sunucusundaki tüm HTTP gelen isteklerini izlemek istediğimiz bir durum var, ancak tarayıcı veya uygulama istemci tarafında yapılandırmayı değiştiremeyiz. Bu senaryoda, web sunucusundaki tüm gelen / giden istekleri izlemek için web sunucusunun ucunda ters bir proxy kurabiliriz.

Windows platformu için Fiddler tavsiye edilir. Sadece izleme değil, aynı zamanda HTTP isteklerini de düzenleyebilir / yanıtlayabilir. Fiddler'ın ters Proxy olarak nasıl yapılandırılacağına dair bu referansa bakın

Linux platformu için Charles Web Debugging Proxy kullanılabilir.

Test adımları:

1. Web Server'a Fiddler veya Charles'ı yükleyin
2. Fiddler veya Charles'ı Ters Vekil olarak yapılandırmak
3. HTTP trafiğini yakalayın
4. HTTP trafiğini kontrol edin
5. HTTP isteklerini değiştirin ve test için değiştirilmiş talepleri tekrar oynatın

Port Forwarding (Liman İlerleme)

Liman iletim, istemci tarafı değişiklikleri olmadan HTTP isteklerini engellememize izin vermenin başka bir yoludur. Ayrıca Charles'ı bir ÇOCUK proxy olarak kullanarak liman iletmek veya Port Forwarding araçlarının kullanımları olarak kullanabilirsiniz. Tüm istemci tarafı yakalanan tüm trafiği web sunucusu bağlantı noktasına iletmemizi sağlayacaktır.

Test akışı şunlar olacaktır:

1. Charles veya port ileri taşımacılığını başka bir makineye veya web Server'a yükleyin
2. Charles'ı Çorap vekili olarak port ileri olarak yapılayın.

TCP-level Network Traffic Capture (TCP düzeyinde ağ trafik yakalama)

Bu teknik, TCP düzeyindeki tüm ağ trafiğini izler. TCPDump veya WireShark araçları kullanılabilir. Ancak, bu araçlar yakalanan trafiği düzenlememize ve test için değiştirilmiş HTTP taleplerini göndermemize izin vermez. Yakalanan trafik (PCAP) paketlerini tekrar oynatmak için Ostinato kullanılabilir.

Test adımları şunlar olacaktır:

1. Ağ trafiğini yakalamak için Web Sunucusunda TCPDump veya WireShark'ı etkinleştirin
2. Yakalanan dosyaları izleyin (PCAP)
3. PCAP dosyalarını ihtiyaç üzerine göre Ostinato aracına göre düzenleyin

4. HTTP isteklerini yanıtlayın

Fiddler veya Charles, bu araçlar HTTP trafiğini yakalayabilir ve ayrıca değiştirilmiş HTTP isteklerini kolayca düzenleyebilir/telefon edebilir. Buna ek olarak, web trafiği HTTPS ise, HTTPS mesaj gövdesini incelemek için kablosuz tuşu web sunucusu özel anahtarını içe aktarması gerekecektir. Aksi takdirde, yakalanan trafiğin HTTPS mesaj gövdesinin hepsi şifrelenecektir.

Tools (Araçlar)

- Fiddler
- TCPProxy
- Charles Web Debugging Proxy
- WireShark
- PowerEdit-Pcap
- pcapteller
- replayproxy
- Ostinato

References (Referanslar)

- Charles Web Debugging Proxy
- Fiddler
- TCPDUMP
- Ostinato