PROJE FİNAL RAPORU

Adı Soyadı: İrem Onaran

Tarih: 15/12/2025

1. PROJE İSMİ

Web Güvenlik Başlıkları, Çerez ve TLS Konfigürasyon Analiz Aracı

2. ÖZET

Bu dönem projesi kapsamında, web sunucularının güvenlik yapılandırmalarını otomatik olarak analiz eden Python tabanlı bir araç geliştirilmiştir. Geliştirilen uygulama; HTTP güvenlik başlıklarını (Security Headers), çerez yapılandırmalarını, SSL/TLS sürümlerini ve sunucu üzerindeki kritik portları taramaktadır. Elde edilen bulgular, risk seviyelerine göre renklendirilmiş bir formatta terminal ekranına yansıtılmakta ve aynı zamanda detaylı bir .txt raporu olarak kaydedilmektedir. Proje sonucunda, manuel olarak yapılması zaman alan kontrollerin saniyeler içinde tamamlanması sağlanmıştır.

Bu araç, port tarama ve zafiyet analizi özellikleri içerdiğinden, yalnızca yasal izinlerin alındığı hedeflerde veya kişisel test ortamlarında (localhost) eğitim amacıyla kullanılmak üzere geliştirilmiştir.

3. GİRİŞ VE PROJENİN AMACI

Siber güvenlik derslerinde de sıkça vurgulandığı üzere, web uygulamalarına yönelik saldırıların önemli bir kısmı sunucu taraflı yapılandırma hatalarından (misconfiguration) kaynaklanmaktadır. Özellikle XSS ve MITM (Man-in-the-Middle) gibi saldırılar, HSTS veya CSP gibi güvenlik önlemlerinin eksikliğinde başarıya ulaşmaktadır.

Bu projenin temel amacı, karmaşık araçlar kullanmadan, bir web sitesinin temel güvenlik duruşunu (security posture) analiz edebilen hafif ve taşınabilir bir script yazmaktır. Proje sürecinde sadece web katmanıyla sınırlı kalınmamış, ağ katmanına (Network Layer) inilerek açık port taraması özelliği de eklenmiştir.

4. KULLANILAN YÖNTEM VE TEKNOLOJİLER

Proje Python dili kullanılarak geliştirilmiştir. Hazır güvenlik kütüphaneleri kullanmak yerine, protokolün çalışma mantığını daha iyi kavrayabilmek adına temel kütüphaneler tercih edilmiştir:

- **requests:** HTTP/HTTPS isteklerini yönetmek ve sunucudan dönen ham başlıkları (Raw Headers) yakalamak için kullanılmıştır.
- **socket:** Ağ katmanında port taraması yapmak ve sunucuyla doğrudan bağlantı kurmak için kullanılmıştır.
- **ssl:** Sunucunun kullandığı şifreleme protokolünü (TLS 1.2/1.3 vb.) ve sertifika detaylarını sorgulamak için kullanılmıştır.
- **argparse:** Aracın komut satırından (CLI) profesyonel bir şekilde parametre alabilmesi için eklenmiştir.

Kullanım: Araç terminal üzerinden python analyzer.py -u google.com komutuyla veya parametresiz çalıştırılarak interaktif modda kullanılabilir.

5. PROJE MİMARİSİ VE ANALİZ ADIMLARI

Bu bölümde, geliştirilen analiz aracının genel mimarisi ile hedef web sitesi üzerinde gerçekleştirilen güvenlik kontrollerinin hangi adımlar izlenerek yapıldığı ayrıntılı olarak açıklanmaktadır. Geliştirilen araç (analyzer.py), hedef URL'yi aldıktan sonra sırasıyla 5 aşamalı bir analiz gerçekleştirmektedir:

- **HTTP Güvenlik Başlıkları Kontrolü**
  Sunucudan dönen yanıttaki başlıklar taranarak Strict-Transport-Security, X-Frame-Options gibi korumaların varlığı denetlenmektedir.
  - *Önemli Detay:* Content-Security-Policy başlığının sadece "Report-Only" modunda olması durumu kod içerisinde ayrı bir if-else bloğu ile yakalanmış ve bu durum "Tam Koruma Sağlamıyor" şeklinde raporlanmıştır.

  HTTP güvenlik başlıkları, web sitelerinde sık karşılaşılan XSS ve clickjacking gibi saldırılara karşı temel koruma sağlar. Bu başlıkların eksik olması durumunda tarayıcı taraflı saldırılar daha kolay gerçekleştirilebilir. Bu araç, sunucudan dönen başlıklarda bu korumaların eksik olup olmadığını kontrol etmektedir.

- **Çerez (Cookie) Güvenliği**
  Sunucu tarafından istemciye gönderilen çerezler analiz edilerek Secure ve HttpOnly bayraklarının eksik olup olmadığı kontrol edilmektedir. Bu bayrakların eksik olması, kullanıcı oturumlarının güvenliğini riske atabileceğinden araç tarafından uyarı olarak raporlanmaktadır.

- **TLS/SSL Analizi**
  İstemci ile sunucu arasındaki iletişimin güvenliğini değerlendirmek amacıyla, sunucunun kullandığı TLS sürümü ve SSL sertifikasının geçerlilik süresi kontrol edilmektedir. Güncel olmayan TLS sürümleri güvenlik riski olarak değerlendirilmektedir.

- **Sunucu Bilgi İfşası (Reconnaissance)**
  Sunucunun Server: Apache/2.4 veya X-Powered-By: PHP/7.4 gibi versiyon bilgisi verip vermediği kontrol edilmektedir. Bu tür bilgilerin açık şekilde paylaşılması, saldırıların daha kolay planlanmasına neden olabilir. Araç, sunucunun bu bilgileri yanıt başlıklarında paylaşıp paylaşmadığını kontrol etmektedir.

- **Port Taraması (Network Scan)**
  Aracın web güvenliğinin ötesine geçerek sunucu güvenliğini de denetlemesi hedeflenmiştir. Bu kapsamda FTP (21), SSH (22), SQL (3306) gibi sık kullanılan ve kritik kabul edilen portlar üzerinde bir tarama gerçekleştirilmektedir. Açık durumda tespit edilen portlar, saldırı yüzeyini artırabileceği için raporlanmaktadır.

CIA ve OWASP Değerlendirmesi

Geliştirilen analiz aracı tarafından tespit edilen zafiyetler, temel olarak CIA güvenlik üçgeni (Confidentiality, Integrity, Availability) çerçevesinde değerlendirilmektedir. Örneğin, HSTS eksikliği ve zayıf TLS sürümleri gizlilik (Confidentiality) açısından risk oluştururken; Content-Security-Policy ve X-Frame-Options başlıklarının eksikliği bütünlük (Integrity) ihlallerine yol açabilmektedir. Açık portlar ise saldırı yüzeyini genişleterek erişilebilirlik (Availability) ve gizlilik üzerinde olumsuz etki yaratmaktadır. Bu bulgular, OWASP Top 10 kapsamında özellikle A05: Security Misconfiguration ve A02: Cryptographic Failures kategorileriyle örtüşmektedir. Proje kapsamında yapılan analizler, bu teorik kavramların pratikteki karşılıklarını somut örneklerle göstermektedir.

6. TEST SONUÇLARI VE DOĞRULAMA

Kodun doğruluğunu test etmek için farklı güvenlik seviyelerindeki siteler üzerinde denemeler yapılmıştır.

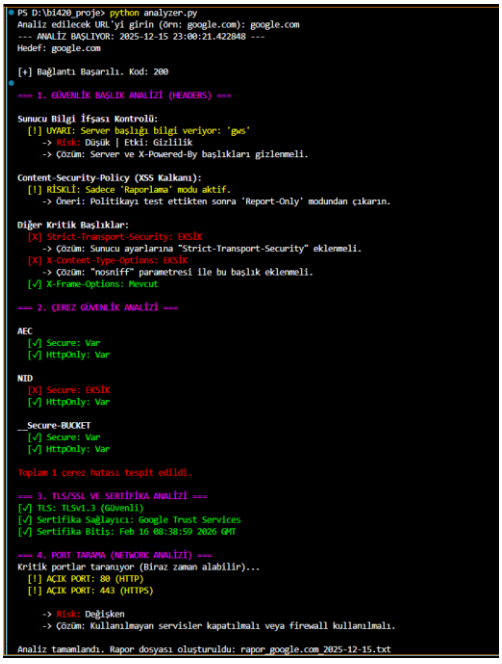
6.1. Zafiyetli Site Testi

Güvenlik önlemleri eksik olan test sitelerinde (örn: example.com), aracın eksik başlıkları "KIRMIZI" renk koduyla işaretlediği ve her eksiklik için bir çözüm önerisi sunduğu doğrulanmıştır

Şekil 1: example.com analizinde tespit edilen zafiyetler ve çözüm önerileri.

6.2. Güvenli Site Testi

Google üzerinde yapılan taramada, aracın güvenlik başlıklarını ve TLS 1.3 sürümünü doğru tespit ettiği görülmüştür. Ayrıca Google'ın sunucu bilgisini (Server: gws) ifşa ettiği araç tarafından yakalanmıştır.

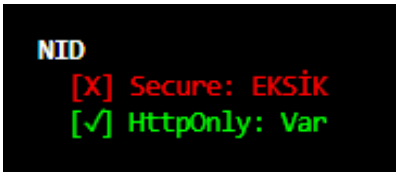


Şekil 2: google.com hedefli başarılı analiz ve tespit edilen güvenlik parametreleri.

6.3. Doğrulama (Cross-Check)

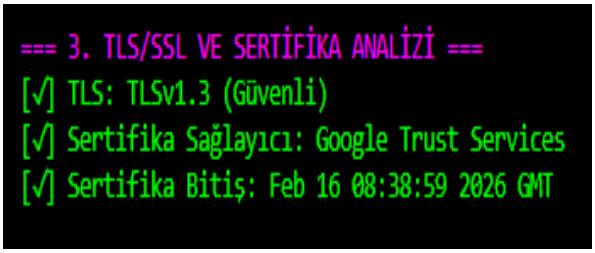
Aracın bulduğu sonuçlar, Google Chrome Geliştirici Araçları ve cURL komutları ile karşılaştırılmıştır.

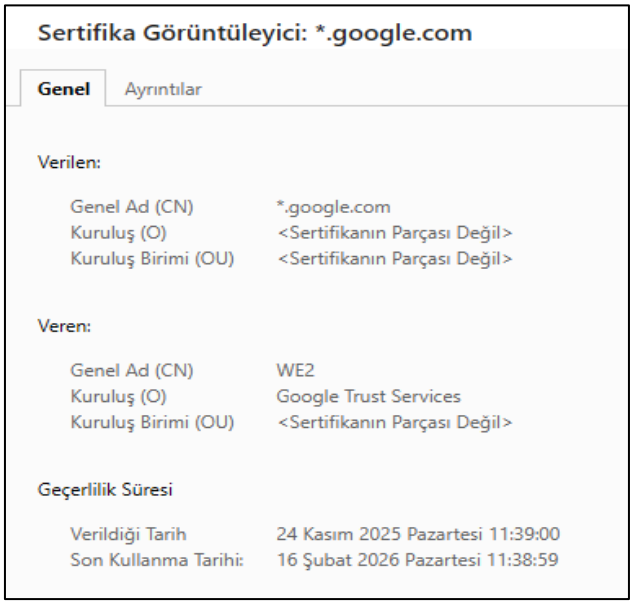
- Chrome tarayıcısında 'Secure' olarak işaretlenen NID çerezinin, araç tarafından yapılan ham veri analizinde 'Eksik' olarak tespit edilmesi; raporun 7. bölümünde detaylandırılan sunucu/tarayıcı davranış farkını doğrulamaktadır.



Name	V...	Domain	P...	E.▲	Size	HttpOnly	Secure
NID	5...	.google.com	/	2...	419	✓	✓

- SSL sertifika sağlayıcısı (Google Trust Services / WE2) ve sertifika bitiş tarihi, tarayıcı detaylarıyla birebir uyuşmaktadır.





7. KARŞILAŞILAN ZORLUKLAR VE ÇÖZÜMLER

Proje geliştirme sürecinde karşılaşılan en büyük teknik zorluk, tarayıcıların ve sunucuların çerez yönetimindeki davranış farkı olmuştur.

- **Sorun:** Tarayıcıda "Secure" görünen bir çerez, Python ile atılan ilk istekte bazen "Secure" etiketi olmadan gelmekteydi.

- **Analiz:** Bunun sebebinin sunucuların yük dengeleme politikaları ve oturum durumu olduğu anlaşılmıştır.
- **Çıkarım:** Bu durum, tarayıcı tabanlı analizlerin bazen yanıltıcı olabileceğini ve sunucuya atılan ham isteklerin analiz için daha saf veri sağladığını göstermiştir.
- TLS Versiyon Tespiti: Araç, sunucu ile bağlantı kurarken 'uzlaşılan' (negotiated) şifreleme protokolünü raporlamaktadır. Sunucunun desteklediği tüm protokollerin dökümü (Enumeration) projenin kapsamı dışında bırakılmıştır.

8. SONUÇ

Bu dönem projesi ile birlikte, derste teorik olarak gördüğümüz HTTP protokolü, SSL el sıkışması ve soket programlama konuları pratiğe dökülmüştür. Ortaya çıkan araç; modüler yapısı, hata yakalama mekanizmaları ve raporlama özelliği ile temel düzeyde bir güvenlik denetimini başarıyla gerçekleştirmektedir. İlerleyen çalışmalarda araca SQL Injection taraması gibi daha ofansif özelliklerin de eklenmesi mümkündür.

9. KAYNAKÇA

- OWASP Foundation, "OWASP Top 10:2021", A02: Cryptographic Failures & A05: Security Misconfiguration.
- MDN Web Docs, "HTTP Security Headers", Mozilla Developer Network.
- Python Software Foundation, "ssl — TLS/SSL wrapper for socket objects", Python 3.10 Documentation.