Yıldız CTI Teknik Raporu

WEB SCRAPER PROJESİ

Furkan YOLCU

İçindekiler

1. GİRİŞ VE AMAÇ

Bu proje, Siber Tehdit İstihbaratı (CTI) eğitimi kapsamında Go (Golang) programlama dili kullanılarak geliştirilmiş bir Web Scraper uygulamasıdır. Program, verilen bir web sitesinin HTML içeriğini çekmekte, ekran görüntüsünü almakta ve sayfa içindeki linkleri listelemektedir.

Siber Tehdit İstihbaratı toplamanın en temel adımı, web sayfalarından veri okumak ve çekmektir. Bu araç, potansiyel tehdit kaynaklarından veri toplanması, zararlı web sitelerinin analizi ve güvenlik araştırmaları için temel bir yapı taşı oluşturmaktadır.

2. KULLANILAN TEKNOLOJİLER

2.1 Programlama Dili

- Go (Golang) 1.25.5

2.2 Kütüphaneler

- chromedp/chromedp v0.14.2
- chromedp/cdproto
- gobwas/ws v1.4.0

2.3 Go Standart Kütüphaneleri

- context: İptal ve timeout yönetimi
- flag: Komut satırı argüman işleme
- fmt: Formatlı giriş/çıkış
- html/template: HTML template işleme
- net/http: HTTP sunucu ve istemci
- os: İşletim sistemi etkileşimi
- path/filepath: Dosya yolu işleme
- strings: String manipülasyonu
- time: Zaman işlemleri

3. PROGRAM MİMARİSİ VE ÇALIŞMA MANTIĞI

3.1 Temel Fonksiyonlar

main(): Program giriş noktası. Komut satırı argümanlarını işler ve uygun modu başlatır.

scrapeOnce(): Ana scraping fonksiyonu. HTTP status kodu kontrol eder, headless Chrome başlatır, sayfaya gider, HTML içeriği çeker, ekran görüntüsü alır ve linkleri toplar.

getSiteName(): URL'den dosya adı için uygun site adı çıkarır. http/https ön eklerini ve www alt alan adını kaldırır, özel karakterleri alt çizgi ile değiştirir. getStatusCode(): Hedef URL'nin HTTP durum kodunu alır. startServer(): Web arayüzünü başlatan HTTP sunucusu.

ensureDirs(): Gerekli çıktı klasörlerini oluşturur (data/, screenshot/, urls/).

3.2 Program Akışı

1. Komut satırı argümanları parse edilir
2. --serve modu ise web sunucusu başlatılır, değilse CLI modu çalışır
3. scrapeOnce() fonksiyonu çağrılır
4. HTTP status kodu alınır
5. Headless Chrome başlatılır
6. Hedef URL'ye gidilir
7. HTML içeriği çekilir
8. Tam sayfa ekran görüntüsü alınır
9. Sayfadaki tüm linkler toplanır
10. Çıktılar dosyalara kaydedilir

4. KULLANIM KILAVUZU

4.1 Komut Satırı Kullanımı

Tek bir site için scraping:

- go run main.go https://google.com

Timeout ayarı ile (varsayılan 30 saniye):

- go run main.go --timeout 60 https://google.com

4.2 Web Arayüzü Kullanımı

Web sunucusunu başlat:

Tarayıcıda http://127.0.0.1:8080 adresine giderek kullanılır.

4.3 Çıktı Dosyaları

Klasör	Dosya Formatı	Açıklama
data/	<site>_data.html	Sayfa HTML içeriği
screenshot/	<site>_screenshot.png	Sayfa ekran görüntüsü
urls/	<site>_urls.txt	Sayfadaki linkler listesi

5. HATA YÖNETİMİ

Program aşağıdaki hata durumlarını ele almaktadır:

Hata Türü	Açıklama
Bağlantı Hatası	Sunucuya ulaşılamama durumunda "status kodu alinamadi" mesajı verilir
HTTP 404	Sayfa bulunamadığında HTTP durum kodu gösterilir
Timeout	İstek zaman aşımında hata mesajı ile bildirilir
Dosya Yazma	Dosya oluşturma hatası spesifik mesaj ile bildirilir

6. TEST EDİLEN WEB SİTELERİ

Program, yaygın olarak kullanılan çeşitli web siteleri üzerinde başarıyla test edilmiştir. Bazı siteler bot koruma mekanizmaları ile otomatik erişimi engellemektedir; ancak bu tür korumaları bulunmayan sitelerde program sorunsuz şekilde çalışmaktadır.

7. ÖRNEK ÇALIŞTIRMA ÇIKTISI

PS C:\Users\Furkan\Desktop\Scraper> go run main.go https://google.com

Hedef URL: https://google.com http status 200 html icerigi kaydedildi

data\google.com_data.html ekran goruntusu kaydedildi

screenshot\google.com_screenshot.png

16 adet url kaydedildi urls\google.com_urls.txt islem

basariyla tamamlandi HTTP 200

8. OLUŞTURULAN DOSYA YAPISI

Scraper/

```
├── main.go
├── go.mod
├── go.sum
├── README.md
├── data/
│   └── ... (HTML dosyaları)
├── screenshot/
│   └── ... (PNG dosyaları)
└── urls/
    └── ... (TXT dosyaları)
```

9. EK ÖZELLİKLER (BONUS)

9.1 URL Listeleme (Ek Puan Görevi)

Program, çekilen sayfadaki tüm <a> etiketlerinin href özelliklerini toplayarak ayrı bir dosyaya kaydeder.

9.2 Web Arayüzü

Kullanım kolaylığı için HTML tabanlı bir web arayüzü geliştirilmiştir.

9.3 Yapılandırılabilir Timeout

--timeout parametresi ile istek zaman aşımı süresi ayarlanabilir.

9.4 HTTP Durum Kodu Kontrolü

Her istekte HTTP durum kodu alınarak kullanıcıya bildirilir.

10. SONUÇ

Bu proje, Go dilinde güvenilir HTTP iletişimi kurabilme yeteneğini göstermekte ve CTI çalışmaları için temel web scraping işlevselliğini sağlamaktadır.

Karşılanan Gereksinimler:

URL'ye Bağlanma: Belirtilen URL'ye HTTP isteği gönderilmektedir

Hata Kontrolü: HTTP hata kodları (404, bağlantı hatası vb.) yakalanıp kullanıcıya bildirilmektedir

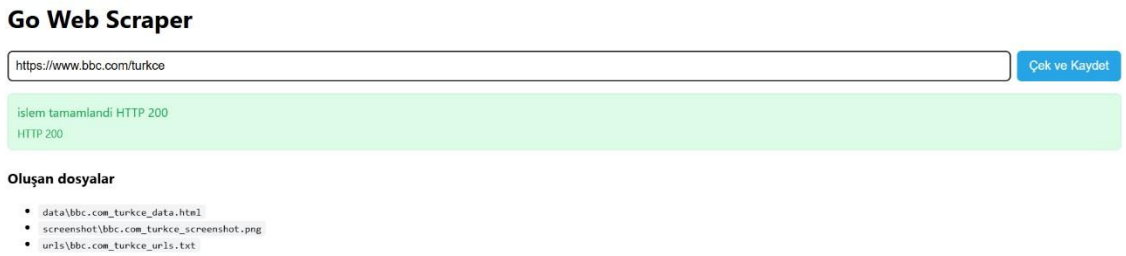
Veriyi Kaydetme: HTML içeriği dosyaya kaydedilmektedir

Ekran Görüntüsü: Sayfa ekran görüntüsü PNG formatında kaydedilmektedir

URL Listeleme (Bonus): Sayfa içindeki linkler listelenmektedir

15+ Site Testi

11. EKRAN GÖRÜNTÜLERİ



Örnek URL Listesi

https://reqres.in/ https://reqres.in/#how

https://reqres.in/#use-cases

https://reqres.in/#pricing

https://app.reqres.in/documentation

https://reqres.in/signup

https://reqres.in/signup

https://app.reqres.in/getting-started

https://app.reqres.in/documentation

https://reqres.in/signup

https://reqres.in/signup

https://reqres.in/signup

https://reqres.in/signup

mailto:hello@reqres.in

https://reqres.in/signup

https://app.reqres.in/getting-started

https://app.reqres.in/documentation

mailto:hello@reqres.in

Github reposu;

https://github.com/Furkanyolcu/web-scraper

İlgili Projeler

Daha önce yapay zeka alanında geliştirdiğim Chrome tarayıcı eklentisi olan projeme aşağıdaki bağlantıdan ulaşılabilir:
https://github.com/Furkanyolcu/SahibindenCarAnalyzer