# YILDIZ CTI_GOREV_1

## PROJE RAPORU: GO DILI ILE CTI ODAKLI WEB SCRAPER GELIŞTIRIMI

Muhammed Seyrek

ARTIFICIAL INTELLIGENCE DEPARTMENT | CYBERSECURITY

Proje Linki : https://github.com/muhammedSeyrek/Scrapper

1. Giriş

Bu projenin amacı, Siber Tehdit İstihbaratı (CTI) süreçlerinde "Keşif" (Reconnaissance) aşamasını otomatize etmektir. Hedef web sitelerinden HTML kaynak kodunu, ekran görüntülerini ve dış bağlantıları (linkleri) toplayan, Go dili tabanlı bir araç geliştirilmiştir. Araç, anti-bot önlemlerine sahip modern güvenlik firmalarının sitelerinde bile çalışacak şekilde optimize edilmiştir.

## 2. Kurulum ve Ortam Hazırlığı

Proje geliştirme sürecine Go modül yapısının oluşturulmasıyla başlanmıştır. Gerekli olan chromedp (Headless Browser motoru) ve ağ dinleme kütüphaneleri projeye dahil edilmiştir.

- go mod init ile proje başlatılmış,
- go get komutları ile bağımlılıklar indirilmiş,
- go mod tidy ile proje temizlenmiş ve paketlenmiştir.

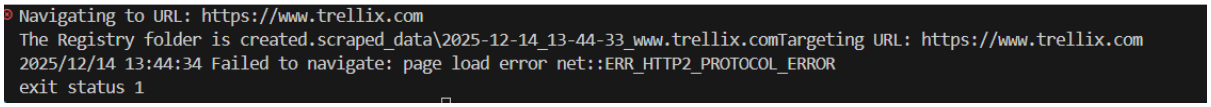
```
(base) PS C:\Users\Seyrek\Desktop\Scrapper> go version
go version go1.25.5 windows/amd64
(base) PS C:\Users\Seyrek\Desktop\Scrapper> go mod init scrapper-assignment
go: creating new go.mod: module scrapper-assignment
go: to add module requirements and sums:
        go mod tidy
(base) PS C:\Users\Seyrek\Desktop\Scrapper> go get -u github.com/chromedp/chromedp
go: downloading github.com/chromedp/chromedp v0.14.2
go: downloading github.com/chromedp/cdproto v0.0.0-20250724212937-08a3db8b4327
go: downloading github.com/chromedp/cdproto v0.0.0-20250803210736-d308e07a266d
go: downloading github.com/go-json-experiment/json v0.0.0-20250725192818-e39067aee2d2
go: downloading github.com/gobwas/ws v1.4.0
go: downloading github.com/go-json-experiment/json v0.0.0-20251027170946-4849db3c2f7e
go: downloading github.com/chromedp/sysutil v1.1.0
go: downloading github.com/gobwas/httphead v0.1.0
go: downloading github.com/gobwas/pool v0.2.1
go: downloading golang.org/x/sys v0.34.0
go: added github.com/chromedp/cdproto v0.0.0-20250803210736-d308e07a266d
go: added github.com/chromedp/chromedp v0.14.2
go: added github.com/chromedp/sysutil v1.1.0
go: added github.com/go-json-experiment/json v0.0.0-20251027170946-4849db3c2f7e
go: added github.com/gobwas/pool v0.2.1
go: added github.com/gobwas/ws v1.4.0
```

### 3. Karşılaşılan Teknik Zorluklar ve Çözümler

Geliştirme sürecinde, basit HTTP istekleriyle aşılamayan iki kritik problemle karşılaşılmış ve çözülmüştür.

#### 3.1. Problem: WAF ve Protokol Hataları (Trellix Örneği)

Yüksek güvenlikli sitelerde (örneğin Trellix.com), standart tarayıcı ayarlarıyla gidildiğinde sunucu bağlantıyı reddetmiş ve net::ERR_HTTP2_PROTOCOL_ERROR hatası alınmıştır. Bu durum, sitenin bot tespit sisteminin (WAF) devreye girdiğini göstermektedir.

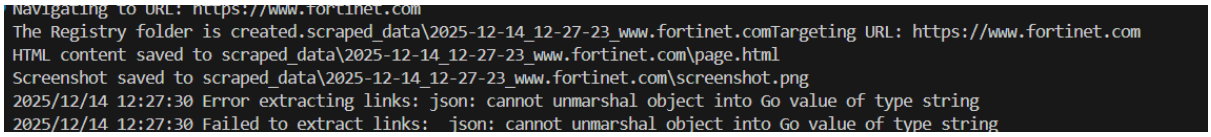



**Çözüm:** Tarayıcı konfigürasyonuna "Gizlilik Modu" eklenmiştir:

1. **User-Agent Manipülasyonu:** Standart bir Windows/Chrome kullanıcısı taklidi yapıldı.
2. **Disable HTTP/2:** Protokol hatalarını önlemek için HTTP/2 devre dışı bırakıldı.
3. **Timeout:** Yükleme süresi uzatıldı.

#### 3.2. Problem: Veri Tipi Uyuşmazlığı (Fortinet Örneği)

Sitedeki linkleri çekerken, bazı sitelerin (örneğin Fortinet) linkleri sadece metin (string) olarak değil, SVG nesnesi (object) olarak sunduğu görülmüştür. Go dili, gelen veriyi doğrudan metin dizisine ([]string) çevirmeye çalıştığında json: cannot unmarshal object hatası alarak çökmüştür.




**Çözüm:** Veri çekme yöntemi değiştirilmiştir. Tarayıcı tarafında (JavaScript) çalışan özel bir filtre yazılarak, SVG nesneleri ayıklanmış ve veriler Go tarafına güvenli bir JSON paketi (Stringify) olarak taşınmıştır.

## 4. Programın Çalışması ve Çıktılar

Geliştirilen araç, komut satırı üzerinden URL parametresi alarak çalışmaktadır. Aşağıda **Atlassian** sitesi üzerinde yapılan başarılı bir test görülmektedir. Program siteye bağlanmış, HTTP 200 (OK) kodunu doğrulamış ve verileri kaydetmiştir.

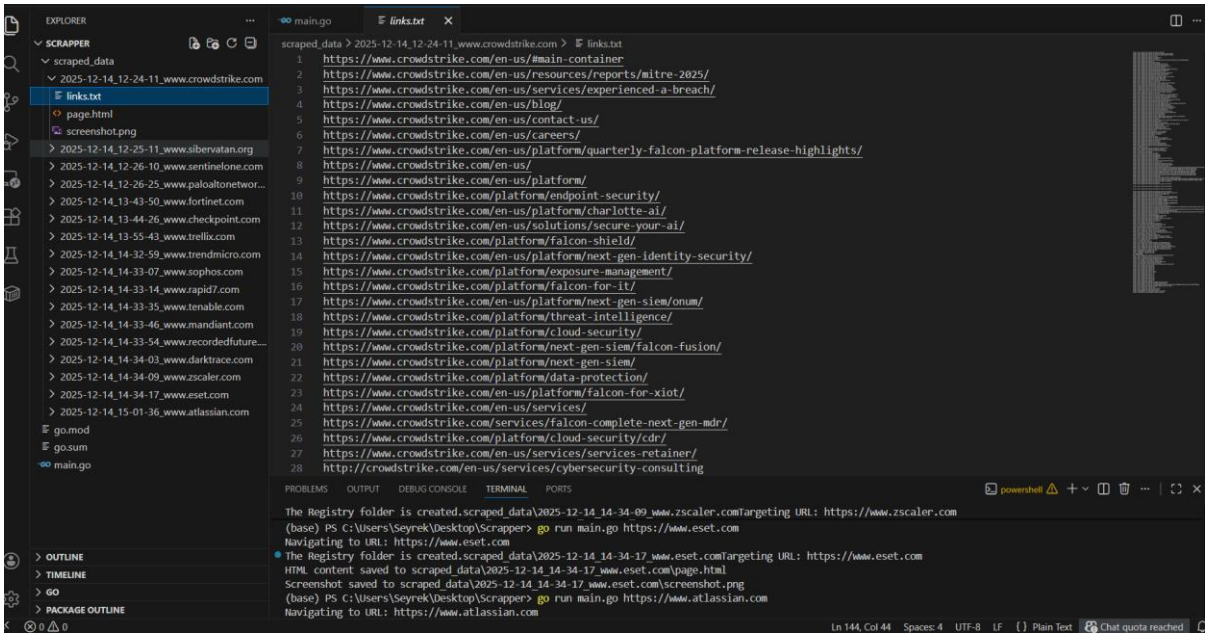
```
(base) PS C:\Users\Seyrek\Desktop\Scrapper> go run main.go https://www.atlassian.com
Navigating to URL: https://www.atlassian.com
The Registry folder is created.scraped_data\2025-12-14_15-01-36_www.atlassian.comTargeting URL: https://www.atlassian.com
Request network: 200 (OK)
Request SUCCESSFUL: Site is accessible.
HTML content saved to scraped_data\2025-12-14_15-01-36_www.atlassian.com\page.html
Screenshot saved to scraped_data\2025-12-14_15-01-36_www.atlassian.com\screenshot.png
Links saved to 336 links in scraped_data\2025-12-14_15-01-36_www.atlassian.com\links.txt
```

## 5. Elde Edilen Veriler ve Raporlama (Kanıtlar)

Proje kapsamında sektörün önde gelen 15 siber güvenlik firması taranmıştır. Her tarama için Tarih_SiteIsmi formatında dinamik klasörler oluşturulmuştur.

### 5.1. Klasör Yapısı ve Ekran Görüntüleri

Aşağıdaki görselde; CrowdStrike, ESET, Zscaler gibi firmalar için oluşturulan klasörler ve alınan ekran görüntüleri yer almaktadır.




### 5.2. Ek Görev: Link Analizi (links.txt)

Sitelerden toplanan URL adresleri, links.txt dosyasında listelenmiştir. Aşağıda CrowdStrike taramasından elde edilen bağlantı listesi görülmektedir.

### 6. Proje Teslimi ve Bakım

Proje tamamlandıktan sonra kodun taşınabilirliğini sağlamak ve gereksiz dosyaları temizlemek için go mod tidy komutu kullanılmış, go.sum dosyası güncellenerek proje teslim haline getirilmiştir.

```
(base) PS C:\Users\Seyrek\Desktop\Scrapper> go mod tidy
go: downloading github.com/ledongthuc/pdf v0.0.0-20220302134840-0c2507a12d80
go: downloading github.com/orisano/pixelmatch v0.0.0-20220722002657-fb0b55479cde
```

### 7. Sonuç

Bu çalışma ile Go dilinin ağ programlama yetenekleri kullanılarak, hata toleransı yüksek, bot korumalarını aşabilen ve CTI süreçlerine entegre edilebilecek modüler bir araç geliştirilmiştir.