Zaawansowane techniki sieciowe

Analysis of CT Logs to Detect Phishing Websites

Prowadzący:
prof. dr hab. inż. Andrzej Feliks Duda

Wykonali:

Grzegorz Febert

Miłosz Maculewicz

Paweł Kwieciński

Łukasz Łukaszewski

Warszawa 2025

Wprowadzenie

Phishing to próba wyłudzenia dostępu bądź informacji o ofierze. Atakujący wykorzystują w tym celu np. domeny łudząco przypominające prawdziwe, gdzie treść strony wygląda identycznie do oryginalnego serwisu. W ten sposób poprzez wypełnienie formularza na takiej stronie dane zostają przekazane atakującemu.

Projekt **Phishing Detector** ma na celu automatyczne wykrywanie potencjalnych stron phishingowych. Jest to możliwe na podstawie analizowanych certyfikatów TLS publikowanych w publicznych dziennikach (CT Logs) oraz porównaniu z dodatkowymi integracjami usług zewnętrznych (Google Safe Browsing, VirusTotal, PhishTank). System pozwala na jednorazowy skan lub ciągły monitoring w czasie zbliżonym do rzeczywistego.

Funkcjonalności

- Pobieranie certyfikatów zawierających słowa kluczowe z crt.sh
- Generowanie wariantów typosquattingowych (DNSTwist)
- Weryfikacja reputacji domen: VirusTotal, Google Safe Browsing, PhishTank
- Heurystyczne wyliczanie oceny ryzyka (risk score)
- Zapis wyników i alertów w bazie SQLite
- Interfejs CLI: tryb skanowania pojedynczego i ciągły monitoring

Przebieg wykonania programu

Program monitoruje nowe certyfikaty SSL związane z wybranymi słowami kluczowymi i sprawdza odpowiadające im domeny pod kątem phishingu. Oto jak to działa w skrócie:

1. **Konfiguracja i inicjalizacja**
    - W konstruktorze definiowane są listy słów kluczowych (`target_keywords`) i podejrzanych wzorców (`suspicious_patterns`).
    - Tworzone są instancje integracji: VirusTotal, Google Safe Browsing, DNSTwist i PhishTank (o ile podano odpowiednie klucze API lub narzędzia są dostępne).
    - Inicjalizowana jest baza SQLite z tabelami do przechowywania przetworzonych certyfikatów, alertów phishingowych i cache wyników VirusTotal.
2. **Pobieranie certyfikatów (crt.sh)**
    - Dla każdego słowa kluczowego budowane jest zapytanie do crt.sh (`%.keyword.%`), zwracające najnowsze certyfikaty zawierające to słowo w nazwie domeny.

- Wyniki są deduplikowane po `cert_id`, by nie powtarzać tej samej domeny wielokrotnie.

3. **Filtrowanie przetworzonych**
   - Każdy certyfikat ma unikalne `cert_id`. Jeśli już był przetworzony (jest w bazie `processed_certs`), jest pomijany.
4. **Analiza certyfikatu**
   - Z pola `name_value` pobierana jest domena (zazwyczaj główna lub alternatywna nazwa w certyfikacie).
   - Sprawdzane jest, czy domena zawiera którekolwiek słowo kluczowe; jeśli nie, rezygnujemy.
   - Dla domeny wywoływane są integracje:
     - **VirusTotal**: reputacja domeny, liczba silników zgłaszających malware/phishing.
     - **Google Safe Browsing**: czy URL jest oznaczony jako zagrożenie.
     - **PhishTank**: czy domena znajduje się w bazie znanych phishingów.
     - **DNSTwist**: generowanie wariantów domen pod kątem typosquattingu.
   - Sprawdzane są lokalne wzorce regex w nazwie (np. „secure”, długie ciągi cyfr, podejrzane TLD).
   - Na podstawie tych danych obliczany jest wynik ryzyka (`risk_score`), sumując punkty za: dopasowane słowa kluczowe, wzorce, długość domeny, wykrycia w integracjach, liczbę wariantów z DNSTwist itp. Jeśli wynik ≥ 40, uznawane jest za podejrzane.
5. **Zapis wyników**
   - Każdy przetworzony cert jest zapisywany w tabeli `processed_certs`.
   - Jeśli domenę uznano za podejrzaną, tworzony jest nowy wiersz w `phishing_alerts` z detalami (score, które słowa i wzorce, wyniki VT, GSB, PhishTank, liczba wariantów).
6. **Generowanie alertu**
   - Gdy wykryto podejrzaną domenę, w logach wypisywane są ostrzeżenia (WARNING) z podsumowaniem: domena, score, dopasowane słowa, wyniki integracji i link do raportu VT itp.
7. **Tryb jednorazowego skanu vs monitoringu**
   - `run_enhanced_scan()`: wykonuje jeden przebieg powyższych kroków, a potem wyświetla krótkie podsumowanie (ile nowych certyfikatów, ile podejrzanych, lista alertów z ostatnich 7 dni).
   - `monitor_certificates_enhanced(interval)`: w pętli co zadany interwał (domyślnie 300 s) powtarza skan, pomijając już przetworzone certyfikaty, loguje bieżące wyniki i pauzuje między przebiegami. Obsługuje przerwanie i błędy w pętli.
8. **Cache i ochrona przed limitami**
   - Wyniki VirusTotal są cachowane w SQLite przez 24 godziny, by unikać przekroczenia limitów API.

    - Między zapytaniami do crt.sh jest niewielkie opóźnienie (np. `time.sleep(1)` między słowami kluczowymi).
    - DNSTwist jest najpierw sprawdzany, czy dostępny w systemie.
9. **Logowanie**
    - Używany `logger`, który zapisuje do pliku i na konsolę. Dzięki temu operator widzi przebieg skanu, alerty i ewentualne błędy integracji.

Moduły

DNSTwistChecker

- Sprawdza dostępność narzędzia dnstwist
- Generuje warianty typosquattingowe domeny
- Obsługa wyjątków i logowanie błędów

GoogleSafeBrowsingChecker

- Sprawdza adres URL w Google Safe Browsing API
- Obsługuje typy zagrożeń: MALWARE, SOCIAL_ENGINEERING, UNWANTED_SOFTWARE
- Zwraca wynik i listę wykrytych typów zagrożeń

VirusTotalChecker

- Sprawdza reputację domeny w VirusTotal API
- Implementuje rate limiting dla darmowego planu (15 s)
- Cache w SQLite na 24 godziny
- Zapisuje szczegóły skanowania (wyniki, kategorie, link)

PhishTankChecker

- Sprawdza URL w bazie PhishTank (API HTTP)
- Zwraca, czy adres jest w bazie, czy zweryfikowany i czas zgłoszenia

Baza danych

SQLite z tabelami: processed_certs (co już sprawdzono), phishing_alerts (wszystkie wykryte incydenty), virustotal_checks (cache VT), gsb_checks (cache GSB), phishtank_checks (cache Phishtank), dnstwist_results (cache DNSTwist)

Instalacja

Pobranie repozytorium

git clone https://github.com/lukmiik/zaawansowane-techniki-sieciowe-ct-logs.git

cd zaawansowane-techniki-sieciowe-ct-logs/

Wirtualne środowisko i zależności

python3 -m venv venv

source venv/bin/activate

pip install -r requirements.txt

Dnstwist (CLI)

pip install dnstwist

Konfiguracja

1. Ustawienie zmiennych środowiskowych API:
2. export VIRUSTOTAL_API_KEY=your_vt_key
3. export GOOGLE_SAFE_BROWSING_API_KEY=your_gsb_key
4. cd src

Użytkowanie

Pojedynczy skan

python main.py --single

Ciągły monitoring

python main.py --monitor --interval 300

Przykłady użycia

- Wykrycie domeny phishingowej paypal-secure-login.com:
    - Słowo kluczowe: paypal → +10 pkt
    - Podejrzany pattern: -secure-login → +20 pkt

- Typosquatting: 12 wariantów → +20 pkt
- Razem: 50 pkt → alert