Westfälische Hochschule

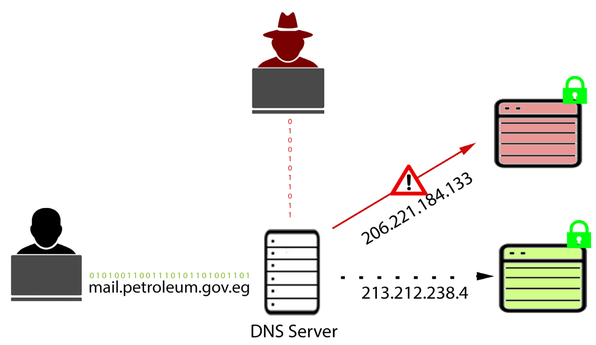
Fachbereich Informatik & Kommunikation
Show & Tell 2021

Domain Hunter

System zum Aufspüren von Domain Hijacking

Modul: Softwareprojekt (Bachelor)

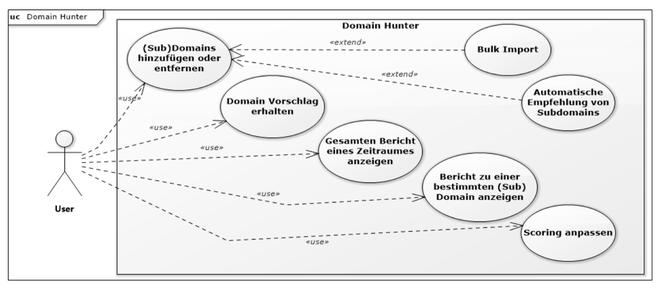
Team: Arne Thiel (WI), Julien Förderer (MI), Michael Kalamarski (WI), Nico Hegemann (WI)



Änderung eines DNS-Records durch einen Attacker

Problemstellung

- Wie läuft Domain Hijacking ab?
- Wie lässt sich Domain Hijacking identifizieren?
- Wie sammelt man möglichst viele Domains?
- Welche Informationen lassen sich über Domains ermitteln?
- Wie verlässlich sind die gewonnenen Daten?
- Wie viele False Positives entstehen durch Clouddienstleister?
- Wie lassen sich Domains automatisiert überwachen?



Anwendungsfalldiagramm

Idee und Konzept

- Ein Expertensystem, das hochskalierbar regelmäßig Domains über das Domain Name System auflöst
- Dauerhafte Speicherung der Daten in einer PostgreSQL-Datenbank, mit dem Ziel, möglichst eine große Anzahl an Vergleichswerten zu erhalten
- Nutzung von weiteren Quellen zur Anreicherung von IP-Adressen
- Abweichungen durch permanentes Abgleichen vorhandener Stammdaten identifizieren und bewerten

Technische Umsetzung

Die Implementierung wurde in der Programmiersprache Python umgesetzt. Des Weiteren wurde eine Datenbank in dem objekt-relationalen Datenbankmanagementsystem PostgreSQL realisiert. Zur Darstellung wurde ein selbst erstelltes kommandozeilen-basiertes Menü angelegt. Um Domains zu überwachen und Hijacking zu erkennen, können diese via CSV-Datei importiert werden. Durch einen Scheduler werden in regelmäßigen Abständen DNS-Abfragen auf Domains und Whois-Abfragen auf die entsprechenden IP-Adressen ausgeführt. Die zurückgegebenen Werte werden im Anschluss mit bereits vorhandenen Daten verglichen. Die Abfrageergebnisse werden durch einen Scoring-Algorithmus bewertet und in die Datenbank geschrieben. Bewertete Datensätze können gefiltert und per Kommandozeile, CSV- oder PDF-Datei exportiert werden.

Benutzeroberfläche (CLI) auf der Kommandozeile

```python
def findSubdomainsDNS(fqdn, n_threads, subdomains):
    """ Funktion die mit Hilfe von Multithreading die Liste
    von möglichen Subdomains ab arbeitet
    @param int n_threads: menge der threads die an diesem Problem arbeiten.
    @param list subdomains: liste von möglichen subdomains die geprüft werden sollen
    """
    global subdomainSet
    subdomainSet = set()
    begin, end, chunk = 0, 0, 0
    thread_list = []
    overhead = len(subdomains) % n_threads
    chunk = int(len(subdomains) / n_threads)
    """jeder thread bekommt seinen anteil an zu prüfenden domains"""
    for tid in range(n_threads):
        if(overhead != 0):
            end = begin + chunk + 1
            overhead -= 1
        else:
            end = begin + chunk
        worker = threading.Thread(target=scanSubdomains, args=(fqdn, tid, begin, end, subdomains))
        thread_list.append(worker)
        worker.start()
        begin = end

    for thread in thread_list:
        thread.join()
    return subdomainSet
```

Quellcode-Ausschnitt des subdomain.py Moduls

Team

Arne.Thiel@studmail.w-hs.de
Julien.Foerderer@studmail.w-hs.de
Michael.Kalamarski@studmail.w-hs.de
Nico.Hegemann@studmail.w-hs.de

Weitere Informationen

Betreuung

Prof. Dr. Christian Dietrich
Fachgebiet: Angewandte Informatik, IT-Sicherheit
Institut für Internet-Sicherheit
Sandra Michalik, B.Sc.
Raphael Springer, B.Sc.