Anleitung Skripte Internet Census 2012

Maximilian Thünemann

August 29, 2013

1 Skripte

Folgend eine kurze Übersicht über die Funktionalität der einzelnen Skripte zum Masterseminar Internet Census 2012. Genaue Erklärungen kann man auch dem eigentlichen Skript entnehmen, da jeweils bei falscher Parametriesierung eine Nutzungsanweisung mit Beispielen ausgegeben wird.

1.1 gatherBodyData.py

Python Skript, das parallel die HTTP Antworten rekursiv aus einem beliebigen Ordner aufsteigend durchsucht. Gesucht wird hierbei nach einem bestimmten Server String und um auf bestimmte Versionen abzuzielen gibt man zusätzlich einen Regulären Suchausdruck nach Python Synthax an. Abschließend kann man angeben in welche Datei die Ergebnisse abgespeichert werden.Als letzten optionalen Parameter kann man die Anzahl der parallelen Threads angeben. Die Ausgabe enthält alle IP Adressen die eine HTTP-Seite (erkennbar durch ¡html¿*¡/html¿) mit den entsprechenden Seitenquelltexten (auch mehrfache Antworten werden aufgeführt!).

1.2 gatherdata.py

Python Skript, dass die HTTP-GetRequest Service Probes eines Ordners rekursiv nach einem Suchbegriff und einem Versionspattern durchsucht. Wie bei gatherBodydata.py kann man den Namen einer Zieldatei angeben und die Anzahl der Threads angeben. Die Ausgabe ist eine Liste, die jeweils als erstes Element ein Versionsflag enthält und dahinter alle IPs auflistet, die diesen Service aufweisen. Das selbe Serverflag kann in dieser Datei mehrfach vorkommen, da es pro Datei ausgewertet wird. Ein Refactoring ist also notwendig.

1.3 locate.py

Das Lokalisationsskript wertet die durch gatherData.py gefundenen Daten aus und lokalisiert im Rahmen der GeoIP Datenbank die IP Adresse. Diese lokalisierten Daten werden in die Ausgabedatei in Tabulatorgetrennter Form wie folgt ausgegeben: IP - Land Kürzel - Land - Stadt - Latitude - Longitude.

1.4 master.sh

Das Masterskript dient zur Ablaufsteuerung des für die Auswertungen im Rahmen des Masterseminar vorgesehenen Ablaufs (Finden - Lokalisieren - Visualisieren). Zusätzlich wird die Häufigkeit des jeweiligen Dienstes ausgewertet durch das statistics.py Skript. Diesem Masterskript wird ein Ordner übergeben, dessen Dateien rekursiv durchsucht werden sollen, sowie ein Scanname, der in den jeweiligen Ausgabedateien als Präfix dient Zusätzlich nimmt das Skript sowohl einem Serverstring als auch ein Versionspattern entgegen. Wichtig: Um eine Expansion von Wildcard Zeichen zu vermeiden (z.B. *) muss die Angabe des Versionspatterns, als auch der Serverstring in Semikola stehen.

1.5 statistics.py

Das Statistik Skript beschränkt sich zur Zeit auf die Auswertung der Häufigkeit des Vorkommens eines bestimmten Servicetyps. Hierzu werden diesem Skript die Ergebnisse des gatherData.py Skriptes übergeben.

1.6 unpack.tar/unzpaq.tar

Diese Skripte gehen rekursiv in das Wurzelverzeichnis und entpacken alle tar bzw. zpaq Dateien. Angegeben werden muss bei diesem Skript das Wurzelverzeichnis, von dem ausgehend die Dateien entpackt werden sollen.

1.7 visualize.py

Dieses Skript benutzt die Ausgabedatei des Lokalisationsskriptes, um die Lokalisierten IPs in eine Weltkarte einzuzeichnen. Hierzu wird eine vorgefertigte Bibliothek namens nugsel-worldmap genutzt. Die Ausgabe erfolgt in Form einer Vektorgrafik im Format svg. Wichtig: Diese Auswertung schlägt bei einer sehr hohen Anzahl von Lokalisierten IP Adressen fehl (>2.000.000) bzw. gibt extrem Speicherintensive Vektorgrafiken aus (die selbst den Chrome überfordern).

1.8 gatherips.py

Diese Skript dient dazu die IP-Adressen zu sammeln die in den Ergebnisdateien des gatherdata.py Skripts vorhanden sind. Diese werden aus der Resultdatei gelesen und in eine beliebige Ausgabedatei geschrieben. Diese Ausgabedatei dient dazu in Verbindung mit dem sslcheck.py Skript zu verfizieren, ob auf einem bestimmten Port wirklich SSL verschlüsselt kommuniziert wird.

1.9 sslcheck.py

Diese Skript ist unvollständig. Der ursprüngliche Einsatzzweck ist gewesen anhand einer Liste von IP Adressen zu überprüfen, ob der Webdienst der auf einer bestimmten IP-Adresse auf Port 443 lauscht auf einen SSL Handshake reagiert. Dies sollte dazu dienen zu verfizieren, ob auf einem bestimmten Port wirklich SSL verschlüsselt kommuniziert wird.

1.10 schlagwortscan.sh

Dieses Bash Skript dient dazu eine Suche nach einer Liste von Schlagworten auf einer Liste von Ports auszuführen. Hierzu wird eine Liste von Schlagworten übergeben die Zeilenweise durchlaufen wird, sowie eine Liste von Ports die ebenso sequentiell abgearbeitet wird. Zur Suche Ergebnisse wird das gatherdata.py Skript eingesetzt

1.11 gatheriXdata.py

Dieses Skript ist eine Vereinfachung von gatherdata.py. Die Funktionsweise unterscheidet sich von gatherdata.py insofern, als das die gesamte HTTP-Antwort nach dem Suchstring durchsucht wird und nicht nur das Serverfeld im HTTP-Header.

1.12 negativeFilteriXResults.py

Diese Skript dient zur Filterung der Ergebnisse des gatheriXData.py Skriptes anhand einer Liste von nicht zulässigen Strings. Die Liste der Strings wird als Argument übergeben und enthält zeilenweise die unzulässigen Zeichenketten.

1.13 positiveFilteriXResults.py

Dieses Skript filtert diejenigen Suchergebnisse des gatheriXData.py Skriptes, auf Wörter die auf einer White List stehen. Die Liste der Strings wird als Argument übergeben und enthält zeilenweise die zulässigen Zeichenketten.

2 Bibliotheken

PyGeoIP `https://github.com/appliedsec/pygeoip`

nugsl-worldmap `http://law.nagoya-u.ac.jp/en/appendix/software/worldmap/`