Developing a visually pleasing attack map to classify attacks

Jonas Thierjung[1]

Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik 4
s6jsthie@uni-bonn.de

Abstract. Dieser Bericht beschreibt die Entwicklung einer sogenannten *Attackmap* im Rahmen der Projektgruppe *Malware Boot Camp*. In diesem Projekt werden die Herausforderungen und der grundlegende Aufbau einer Attackmap untersucht. Dabei wird besonders das Problem beleuchtet, wie man valide Daten für die Attackmap generieren kann. Auch werden verschiedene Tools, welche Open-Source sind, verglichen. Des Weiteren werden auch schon bestehende Attackmaps betrachtet und gegenübergestellt. Zuletzt wird die eigene Entwicklung vorgestellt.

1 Einführung

IT-Angriffe wie DDOS, Spam oder Ransomware sind kein lokales Phänomen und nicht durch Landesgrenzen beschränkt. Weltweit passieren jede Sekunde Angriffe verschiedenster Art. Eine Möglichkeit diese darzustellen und vermittelbar zu machen, sind die sogenannten Attackmaps.
Attackmaps beschreiben Karten sowie Statistiken, welche dem Nutzer durch Animation einen Überblick über IT-Angriffe verschiedenster Art liefern. Oftmals wird der Fokus meist auf weltweite Aktivitäten verschiedenster Botnetze gelegt. Die Angriffe werden dabei in Echtzeit dargestellt und visualisiert.
Ein Beispiel für eine Attackmap ist die vom IT-Sicherheitsdienstleiter *Kaspersky* [1]. Wie in Figur 2 gezeigt, besteht die Attackmap aus einer Darstellung der Weltkarte mit einer Tabelle der Statistiken, sowie animierte Linien, welche einzelne Angriffe vom Ausgangspunkt bis zum Ziel darstellen.

2 Stand der Forschung

In diesem Kontext gibt es zu diesem Thema kaum bis gar keine Quellen oder Veröffentlichung seitens der Fachliteratur. Einzig zu nennen ist das Projekt Starmine [2] aus dem Jahr 2006. In diesem beschreiben die japanischen Forscher eine Visualisierung, welche Cyberangriffe auf eine Weltkarte mittels einer 3-dimensionalen Grafik darstellt. Zu sehen ist dies in Figur 1.
Weitere Literatur konnte im Rahmen dieser Arbeit nicht gefunden werden.

3 Vergleich Attackmaps

Im Rahmen dieser Projektarbeit wurden sich auch bestehende Arbeit angeschaut, charakterisiert und bewertet. Dafür wurden sich ein Subset von Attacksmaps betrachtet. Diese sind wie vorher schon erwähnt die Attackmap von Kaspersky[1],

Fig. 1. Hier die gezeigte Attackmap von Kaspersky

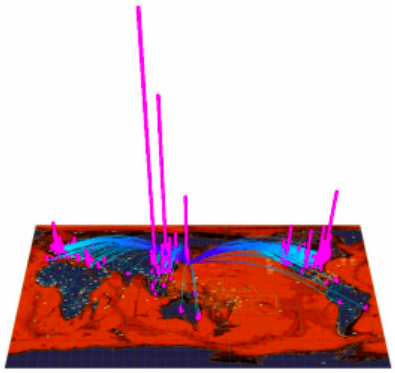

Fig. 2. Die Attackmap der japanischen Forscher arbeitet mit Verbindungslinieren und Balken zur Darstellung der Bedrohungslage

die von FireEye[3] sowie die von Jigsaw (ehemals Google Ideas)[4].
Bei allen drei Attackmaps handelt es sich um Webseiten, welche eine Javascript animierte Weltkarte darstellt und dort die Angriffe darstellt. Bei allen wird Ursprung und Ziel der Attacken durch eine Verbindungslinien gekennzeichnet.
Kasperskys Attackmap stellt dabei eine Bandbreite von Angriffe da. So erfassen die Botnetaktivitäten, Mailspam und Webantiviren an. Sie ist damit die ausführlichste Attackmmap von den Dreien. Weitere Information, also genauere Details, werden leider nicht dargestellt.
FireEyes Attackmap stellt im Gegensatz nur da, das Angriffe existieren, genauere Information, um welche Art von Angriff es sich handelt, fehlen. Einzig eine Tabelle der Top 5 von Cyberattacken betroffenen Industrien wird eingeblendet.
Die Attackmap von Jigsaw dagegen stellt nur DDOS-Angriffe da. Diese werden dabei ausführlich beschrieben. Man erkennt zum Beispiel an der Dicke der Verbindungslinien, wie hoch die Gbps(Gigabyte per second) der Angriffe ist. Auch wird die Dauer der Angriffe angezeigt.
Alle drei Projekte nennen auch ihre Datengrundlage. Als Datenquellen benutzen FIreEyes und Kaspersky ihre eigenen Quellen, die sie als IT-Sicherheitsdienstleiter besitzen. Diese sind nicht öffentlich. Eben so wenig ist die Datenquelle von Jig-

saw öffentlich. Diese verwendet als Datenquelle *Arbor Networks global threat intelligence system* [5].

Fig. 3. Die Attackmap von FireEye

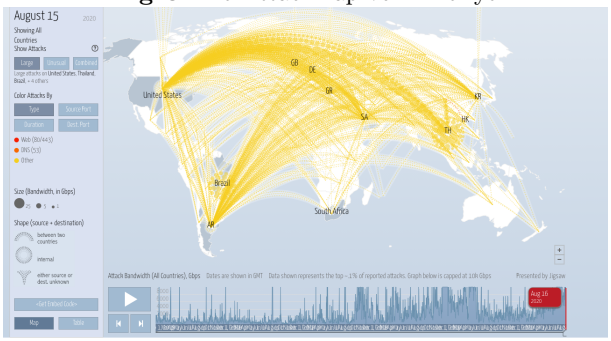

Fig. 4. Die Attackmap von Jigsaw

4 Architektur der Attackmap

Für dieses Projekt bzw. für die Attackmap stellen sich 2 wichtige Fragen. Die erste ist, wie soll die Attackmap die Angriffe visualisieren und woher kommen die Daten, welche die Informationen zu den Angriffen bereitstellen. Beide dieser Fragestellungen werden dabei von zwei verschiedenen Bereichen der Software gelöst, dem Frontend und dem Backend.
Das Frontend löst dabei die erste, das Backend die zweite Frage.
Als Basis für dieses Projekt wird ein Webserver genommen. Der Grund liegt in der Annahme, das die Attackmap eine Visualisierung sein soll, welche öffentlich zugänglich ist. Deshalb wird die Implementierung in diesem Vorhaben als Website geplant.
Damit das Frontend einheitliche Daten verarbeiten kann, muss eine feste Datenstruktur definiert werden, welche das Backend liefern soll. Dafür wurde sich hierbei entschieden, die Daten im JSON-Format auszugeben und über eine REST-API zu liefern.
Die Daten enthalten dabei die wichtigen Information, um welche Art Angriff

sich handelt, welche IP der Angreifer und das Opfer haben, sowie die Länder der beiden mitsamt Längen und Breitengraden.

5 Problematik Daten

Wie wir schon im Abschnitt *Vergleich Attackmaps* erfahren haben, sind alle Datenquellen der vorgestellten Attackmaps nicht öffentlich zugänglich. Generell stellt dies eine große Herausforderung dieses Projekt da.
So gibt es öffentliche Quellen, die statistische die Angriffe der Vergangenheit aufzeigen und darstellen. Diese eignen sich jedoch nicht für eine Attackmap, da sie keine Daten in Echtzeit liefern.
Dafür benötigt man ein System von Honeypots etc. un ein System zu Sammelen und veröffentlichen dieser Daten.
Zum Zeitpunkt des Projektes konnte keine öffentliche Quelle gefunden, deren Daten man verarbeiten konnte, um eine Attackmap darzustellen. Deswegen wird dieses Projekt mit Dummydaten verwirklicht.

6 Open Source Tools

Im Rahmen dieser Projektarbeit wurden sich Open Source Projekte angeschaut und bewertet, in wieweit sie sich für eine Implementierung eignen. Dabei wurde die drei am besten bewertete Attackmaps bei GitHub validiert. Als Bewertungskriterium wurden die GitHub-Sterne gewählt, welches jedes öffentlichen Repository besitzt.
Bei allen drei Projekten muss man betonen, dass diese nicht mehr aktuell sind und nicht gewartet werden.
Der Anfang macht das Projekt "NorseAttack-like" [6], welches eine Weltkarte als Basis hat und die Angriffe als Schallwellen darstellt. Von den vorgestellten, ist dies das neuste, jedoch besitzt es auch die wenigstens Sterne bei GitHub. Leider eignet es sich nicht für die Verwirklichung, da zum einen das Projekt nur eine dürftige Dokumentation besitzt und es mittels eines einfachen Webpackservers implementiert wird. Da dieses Projekt auch eine REST-API hinzugefügt werden soll, ist dies eher ungeeignet.
Das zweite Projekt ist "geoip-attack-map" [7], welches die zweitmeisten Sterne bei GitHub besitzt. Im Gegensatz zum ersteren Projekt wird hier auch ein vollwertiger Server samt Datenbank verwendet, jedoch wird als Kartenbasis MapBox genutzt, welche einen gültigen Apikey benötigt. Dieser ist jedoch kostenpflichtig und deswegen ist dieses Repository leider für dieses Projekt auch ungeeignet.
Das letzte Projekt ist "IPew Attack Map" [8]. Von denen drei Projekten besitzt es die meisten Sterne bei GitHub. "IPew Attack Map" ist eine einfache HTML/Javascript Implementierung einer Attackmap, welche ohne fertige Serverstruktur daherkommt. Deswegen wird diese Repository als Basis zur Entwicklung der Projektsoftware verwendet.

7 Entwicklung Software

Als Grundlage für die Entwicklung der Projektsoftware dient das Framework Laravel [9], welches auf einer Virtual Machine läuft, die durch Laravel Homestead [10] generiert wird.

Es wurde sich für diese Software entschieden, da der Projektteilnehmer bereits Erfahrung mit dem Framework vorweist und es sich für die Implementierung einer Website und einer REST-Api eignet.

7.1 Backend

Im Backend wurde eine eine einfache REST-Api implementiert, welche eine JSON-Datei ausgibt. Da wir leider keine Datengrundlage haben, wird eine Beispiel JSON ausgegeben.
Die JSON sieht dabei wie folgt aus:

```
"date": "2020-08-14T00:00:00.000Z",
"attack": [{
"origin": {
"country": "DR",
"ip": "192.168.2.1",
        "lat": 19,
        "long": -70
},
"": {},
"type": "DDOS",
"destination": {
"country": "US",
"ip": "192.14.3.9",
        "lat": 39,
        "long": -98
}
}]
```

Die Daten enthalten dabei die wichtigen Information, um welche Art Angriff sich handelt, welche IP der Angreifer und das Opfer haben, sowie die Länder der beiden mitsamt Längen und Breitengraden.

7.2 Frontend

Bei Frontend wurde wie vorherigen Abschnitt erwähnt, dass Repository "IPew Attack Map" als Basis genommen. Es wurde soweit angepasst, dass es ein Array von Angriffen bekommt und diese animiert wiedergibt. Dieses Array stammt dabei von der REST-APi.
Auch wurde angepasst, dass die Farbe der Angriffe sich je nach Angriffstyp unterscheidet.

8 Ergebnis Software

Die fertige Arbeit stellt die Angriffe als Verbindungslinien da, wobei das Ziel des Angriffes durch einen weißen Kreis dargestellt wird. Je nach Art des Angriffes, hat die Linie eine andere Farbe.
Eine Übersicht der Farbkodierungen ist in Figur 8 zu sehen.

Rechts unten befinden sich in der Attackmap ein Infokasten, welcher Ziel und Ursprung der gezeigten Angriffe zeigt, sowie Art des Angriffes und Details wie die IP-Adressen der beteiligten Akteure.

Angriffsart	DDOS	Spam	Banking	Standart
Farbe	Rot	Blau	Grün	Orange

Fig. 5. Farbkodierung der Angriffsarten

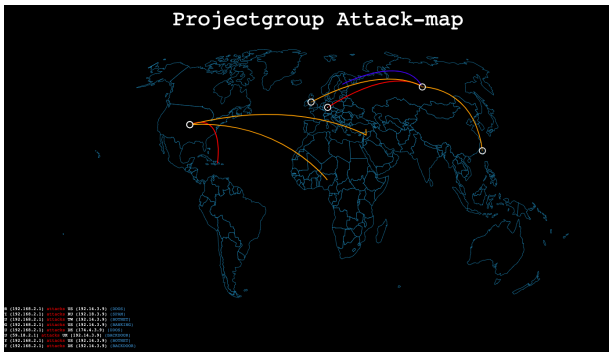

Fig. 6. Die fertige Attackmap

9 Zusammenfassung und Ausblick

Im Rahmen der Projektgruppe "Malware Bootcamp" konnte eine funktionierende Attackmap programmiert werden. Diese ist generisch gestaltet und verarbeitet Daten, welche von einer eigenen REST-Api stammen.
Das Projekt wurde dabei als Webserver implementiert mittels des Laravel Framworks. Als Basis für die Website dient dabei das Repositoy "IPew Attack Map".
Die REST-Api verarbeitet nur Dummydaten, da leider keine Datenquelle im Rahmen dieser Arbeit gefunden werden konnte, welche in Echtzeit Angriffe darstellt.
Eine zukünftige Aufgabe könnte es sein, dieses Projekt mit einer Echtzeit-Datenquelle auszustatten bzw. diese zu entwickeln und mit dieser Arbeit zu kombinieren.

References

[1] *Cybermap kaspersky*, `https://cybermap.kaspersky.com/de`, eingesehen am 14.10.2020.
[2] Y. Hideshima and H. Koike, "Starmine: A visualization system for cyber attacks.," Jan. 2006, pp. 131–138.
[3] *Threatmap fireye*, `https://www.fireeye.com/cyber-map/threat-map.html`, eingesehen am 14.10.2020.
[4] *Digital attack map*, `https://www.digitalattackmap.com/v1`, eingesehen am 14.10.2020.
[5] *Arbor networks global threat intelligence system*, `https://www.netscout.com/product/atlas-intelligence-feed-aif`, eingesehen am 14.10.2020.
[6] *Anorseattack-like*, `https://github.com/yuanzhaokang/NorseAttack-like`, eingesehen am 14.10.2020.

[7] *Geoip-attack-map*, `https://github.com/MatthewClarkMay/geoip-attack-map`, eingesehen am 14.10.2020.
[8] *Ipew attack map*, `https://github.com/hrbrmstr/pewpew`, eingesehen am 14.10.2020.
[9] *Laravel 8*, `https://laravel.com/docs/8.x`, eingesehen am 14.10.2020.
[10] *Homestead*, `https://laravel.com/docs/8.x/homestead`, eingesehen am 14.10.2020.