ASN-Skript: Datensammler haben ausgeschnüffelt

1. Ausgeschnüffelt IPFire IP-Blocking

Datensammlern wie Google, Facebook und Co. begegnet man laut der Studie »Online Tracking: A 1-million-site Measurement and Analysisauf« auf fast jeder Webseite. Insbesondere Google ist mit über 80% Verbreitung schon fast mit einem »Krebsgeschwür« vergleichbar. Hintergrund dieser hohen Verbreitung ist meist die **Bequemlichkeit** von Seitenbetreibern, die externe Ressourcen wie JavaScript oder Schriftarten gerne über Drittanbieter einbinden.

Um Datensammlern wie Google oder Facebook zu entgehen, gibt es unterschiedliche Möglichkeiten. Die meisten von euch verwenden vermutlich einen Adblocker wie uBlock Origin oder einen Pi-Hole, um Tracker wie Google Analytics oder den Facebook Like-Button zu blockieren. Eine weitere, äußerst effektive Methode ist das Blockieren von IP-Adressbereichen, auf der Basis von AS-Nummern, wie im Beitrag Google und Facebook IP-Adressen blockieren aufgezeigt.

Im dritten Teil der IPFire-Artikelserie möchte ich euch ein Skript vorstellen, mit dem sich ganze IP-Adressbereiche von Datensammlern blockieren lassen. **Bonus**: Das Skript lässt sich ebenfalls in Kombination mit iptables auf einem GNU/Linux-Rechner oder der AFWall+ auf Android anwenden. Es ist daher nicht nur für IPFire-User von Interesse.

DIESER BEITRAG IST TEIL EINER ARTIKELSERIE:

- Hardware und Netzwerkaufbau – IPFire Teil1
- DNS-Adblocker Skript für IPFire – IPFire Teil2
- ASN-Skript: Datensammler haben ausgeschnüffelt – IPFire Teil3
- Windows unter Kontrolle – IPFire Teil4

2. Das Problem: Einbindung externer Ressourcen

Den Grund, weshalb wir zur »Holzhammermethode« greifen und ganze IP-Adressblöcke von Anbietern wie Google oder Facebook sperren, möchte ich nachfolgend kurz am Beispiel von Webseiten skizzieren. Wer keinen Wert auf das »Warum« legt, der kann auch gleich zu Ziffer 3 springen.

Das Einbinden externer Ressourcen auf Webseiten verfolgt meist unterschiedliche Ziele. Für viele Webseitenbetreiber ist es schlichtweg »bequem« eine JavaScript-Bibliothek von einer externen Quelle einzubetten, als sich selbst die Mühe zu machen, die für die Funktionalität der Webseite benötigen Ressourcen selbst zu hosten. Insbesondere Webentwicklern wird die Erstellung und Anpassung von Webseiten, durch die Verwendung von vorgefertigten JavaScript-Bibliotheken und entsprechende Frameworks, erleichtert. Zu den bekanntesten Vertretern zählen jQuery, AngularJS oder die MooTools. Auch in diesem Fall werden die notwendigen JavaScript-Bibliotheken, meist aus Bequemlichkeit, über die jeweiligen Frameworks von Drittquellen eingebunden.

Angesichts dieser zunehmend angewandten Praxis, scheint den Webseitenbetreibern nicht bewusst zu sein, welches **Risiko**, für Sicherheit und Privatsphäre, mit der Einbindung externer Ressourcen

einhergeht. Laut der Studie »Exposing the Hidden Web: An Analysis of Third-Party HTTP Requests on One Million Websites« zählen die folgenden extern eingebunden Ressourcen zu den am häufigsten verwendeten:

- **JavaScript**: JavaScript ist eine Skriptsprache, die es erlaubt HTML-Dokumente während der Anzeige im Browser dynamisch zu verändern, Benutzerinteraktionen auszuwerten oder neue Inhalte nachzuladen. Zu den typischen Anwendungsgebieten von JavaScript zählen unter anderem die Datenvalidierung von Formulareingaben, Anzeige von Dialogfenstern oder die Anzeige von Suchvorschlägen während der Eingabe. JavaScript ist nicht per se unsicher, wird aber immer wieder kontrovers diskutiert, da die meisten der bekannt gewordenen Sicherheitslücken in Browsern oftmals eng mit JavaScript verknüpft sind. Leider wurde und wird JavaScript zunehmend für das Fingerprinting von Nutzern »missbraucht«, das ein seitenübergreifenden Tracking ermöglicht. In der Studie »(Cross-)Browser Fingerprinting via OS and Hardware Level Features« wird aufgezeigt, dass sich über bestimmte Merkmale wie die verwendete Bildschirmauflösung, Farbtiefe, im Browser installierter Plugins oder Schriftarten, Besucher zu 99,24 % wiedererkennen lassen / seitenübergreifend Tracken.
- **Bilder**: Insbesondere Bilder werden häufig von Drittseiten zur Auslieferung von Werbung eingebunden oder als »unsichtbare« Web-Bugs missbraucht, die gerne im Bereich des Online-Marketing zum Einsatz kommen, um den Besucher einer Webseite zu tracken.
- **Schriften**: Moderne Browser unterstützen heute das Nachladen von Schriftarten von Webseiten oder Drittquellen, meist mit dem Ziel, browser- und betriebssystemübergreifend ein einheitliches Aussehen zu erreichen. Aus Bequemlichkeit werden diese Schriften allerdings meist nicht von der aufgerufenen Webseite ausgeliefert, sondern von Drittanbietern (Google Fonts) eingebunden.

Welche Risiken, insbesondere mit der externen Einbindung von JavaScript, für die Sicherheit und die Privatsphäre eines Webseitenbesuchers einhergeht, möchte ich kurz aufzeigen.

2.1 Externes JavaScript: Ein »Autsch« für Sicherheit und Privatsphäre

Die Einbindung von (externem) JavaScript verfolgt meist unterschiedliche Ziele. Zu den häufigsten Anwendungsszenarien zählen mitunter das User-Tracking und die Auslieferung von Werbung. Insbesondere die Möglichkeit des »seitenübergreifenden Trackings« hat sich dadurch verbreitet, dass Webseitenbetreiber die Erfassung und Auswertung ihrer Besucher nicht mehr selbst durchführen (wollen), sondern auf **externe** Dienstleister zurückgreifen, die ihnen diese Arbeit »abnehmen«. Das »Outsourcing« der Besucheranalyse macht es allerdings vielfach notwendig, dass ein Webseitenbetreiber externe JavaScript-Bibliotheken einbindet, die das Besucherverhalten erfassen und analysieren.

Die Einbindung dieser externen Ressourcen ist für den Webseitenbetreiber meist bequem und mit wenig Aufwand verbunden, weshalb sich diese Praxis allgemein durchgesetzt hat. Ein weiterer Vorteil externer Tracking-Anbieter stellt die meist **kostenlose** Nutzung für einen Webseitenbetreiber dar, der im Gegenzug mit den Daten seiner Besucher »bezahlt«. Durch die Einbindung externer Dienstleister in eine Webseite geht ein Webseitenbetreiber allerdings eine »Vertrauensbeziehung« ein, ohne sich womöglich mit der daraus resultierenden Tracking-Problematik bzw. der Aufzeichnung sensibler Informationen durch Dritte ausreichend auseinandergesetzt zu haben.

Aber nicht nur hinsichtlich der Privatsphäre kann die Einbettung von »fremdem« JavaScript-Code negative Auswirkungen haben, sondern sowohl auf die Verfügbarkeit, als auch auf die Sicherheit des Webseitenbetreibers und insbesondere seiner Besucher:

- **Verfügbarkeit**: Angenommen ein Seitenbetreiber hat einen Bilder-Slider in seine Webseite über die Domain »example.de/slider.js« eingebunden. Falls nun die Domain, über die das JavaScript eingebunden wird, aus irgendwelchen Gründen nicht erreichbar ist, wird die Einbindung des Bilder-Sliders fehlschlagen. Als Folge ist die Webseite für einen Besucher möglicherweise nicht benutzbar, da über die Einbindung des externen JavaScripts wichtige Funktionen abgebildet werden. Das Einbinden kann sich allerdings nicht nur auf die Benutzbarkeit negativ auswirken, sondern ebenfalls auf die Geschwindigkeit des Seitenaufbaus. Jede Ressource, die von einer externen Domain / Quelle nachgeladen wird, bedeutet, dass der Browser eine Verbindung initiiert und die Ressource von dort »abholt«. Dieser Vorgang ist vergleichsweise langsamer, als wenn der Seitenbetreiber die Ressource bzw. das JavaScript lokal von seinem Webserver anbieten würde.
- **Sicherheit**: Weitaus gravierendere Folgen, insbesondere für die Sicherheit eines Besuchers, können allerdings dadurch entstehen, wenn sich ein Webseitenbetreiber »blind« auf den Drittanbieter verlässt. Man sollte sich vor Augen führen, dass der Betreiber von »example.de« **jederzeit** Änderungen an der JavaScript-Datei vornehmen kann oder möglicherweise »gehackt« wird. Eine böswillige Veränderung von JavaScript-Code kann unter anderem dazu »missbraucht« werden, um Cookies aus dem Kontext einer Webseite zu stehlen oder die Login-Daten (Benutzernamen & Passwort) abzugreifen. Für diesen Vorgang benötigt ein Angreifer noch nicht einmal Zugang zur Webseite des Betreibers, sondern einzig zum JavaScript, das er entweder selbst kontrolliert oder entsprechend beim Drittanbieter modifiziert hat. Fatal ist, dass der Webseitenbetreiber von dieser Veränderung meist nichts »mitbekommt« und somit eine »Vertrauensbeziehung« zum Anbieter eingeht, die insbesondere seinen Besuchern extrem Schaden kann.

Webseitenbetreiber, denen die Sicherheit und Privatsphäre ihrer Besucher am Herzen liegt, sollten daher prüfen, ob die Einbindung von externen JavaScript-Ressourcen zwingend erforderlich ist oder ob die Einbindung auch lokal über die eigene Domain erfolgen kann.

Hinweis

Wer prüfen möchte, welche externen Inhalte bzw. Ressourcen Webseiten einbinden, der kann dies mit Webbkoll tun.

2.2 Weitere Gedanken zur Thematik

Dieser kurze »Ausflug« sollte euch für die Problematik sensibilisieren, weshalb es durchaus sinnvoll sein kann, komplette IP-Adressbereiche von Anbietern zu blockieren, die es insbesondere mit der Privatsphäre eines Nutzers nicht sehr genau nehmen bzw. daraus sogar Kapital schlagen. Genauer gesagt möchten wir verhindern, dass externe Ressourcen von Anbietern wie Google oder Facebook auf Webseiten oder innerhalb von Apps nachgeladen werden, die dafür bekannt sind, das Nutzerverhalten zu tracken.

Das komplette Sperren von IP-Adressblöcken zählt gewiss zur »Holzhammermethode«, die dazu führen kann und auch wird, dass Webseiten nicht mehr funktionieren oder Smartphone-Apps den Dienst versagen. Ihr solltet daher bereit sein, Webseiten, die nicht mehr funktionieren, bspw. über

den Tor-Browser aufzurufen und euch Apps zu suchen, die bspw. auch ohne das Nachladen von Google-Bibliotheken problemlos funktionieren. Solche Apps könnt ihr insbesondere im F-Droid Store finden. Wie ihr komplett ohne Google auskommt könnt ihr im Beitrag »Tschüss Datenkrake: Ein Leben ohne Google« nachlesen.

Hinweis

In einem gesonderten Beitrag werde ich demnächst aufzeigen, welche weiteren Risiken für Sicherheit und Privatsphäre mit der Einbindung externer Ressourcen einhergeht. Ein Beitrag, der sicherlich nicht nur für Nutzer von Interesse sein dürfte, sondern insbesondere für Webseitenbetreiber, die sich Gedanken um eine sicheren und datenschutzfreundlichen Webauftritt machen.

3. ASN-Skript: Die Holzhammermethode

Mitte April diesen Jahres habe ich mir ein Bash-Skript gebastelt, dass auf Basis von ASN-Informationen iptables Blocking-Regeln (IPv4) für ausgewählte Unternehmen generiert. Nach ein paar Anpassungen veröffentlichte ich das Skript anschließend für IPFire-User. Seither sind ein paar Monate vergangen und ein Leser (maloe) aus dem Kuketz-Blog XMPP-Konferenzraum hat das ursprüngliche Skript weiter angepasst, optimiert und verbessert. Er hat aus dem Skript sozusagen ein »Schweizer-Taschenmesser« gemacht, mit dem es möglich ist, Blocking-Regeln für die IPFire, iptables und die AFWall+ zu generieren.

Seit Juli 2017 hat das Skript ein online zu Hause auf NotABug bekommen. Wer in Zukunft also bspw. Fehler findet, Verbesserungsvorschläge einreichen möchte oder ein Problem melden möchte, der findet dort eine entsprechende Anlaufstelle.

3.1 Was macht dieses Skript nun?

Das Ziel unseres Skripts ist die automatische Generierung von IP-basierten Blocking-Regeln für Unternehmen, mit denen ihr nicht in »Berührung« kommen wollt. Die Frage ist nun, wie man eine möglichst **vollständige** Liste aller IP-Adressblöcke bekommt, die zu einem Unternehmen wie Google oder Facebook gehören. Typischerweise haben ISPs, aber auch große internationale Unternehmen, eigene AS-Nummern (ASN). Hinter solch einem autonomen System (AS) verbirgt sich eine Ansammlung von IP-Netzen, welche bspw. über das interne Routing-Protokoll (IGP) miteinander verbunden sind. Internationale Unternehmen wie Google oder Facebook haben ihre eigenen ASNs, wo im Idealfall alle zugehörigen IP-Netze registriert sind.

Das Skript holt sich also auf Basis von ASN-Informationen automatisiert alle zu Google oder Facebook gehörigen IP-Adressblöcke und erstellt daraus Netzwerkobjekte und Gruppen, die sich anschließend in der IPFire GUI als Blocking-Regeln (Verweigern (REJECT)) einbinden lassen. Über entsprechende Optionsparameter des Skripts lassen sich allerdings nicht nur Blocking-Regeln für die IPFire erzeugen, sondern ebenfalls für iptables oder AFWall+.

Als Quelle für die ASN-Informationen nutzt das Skript unterschiedliche Ressourcen, die sich auch erweitern lassen:

- **Registrierte ASN**: Um alle ASN abzufragen, die von einem Unternehmen registriert wurden,

eignet sich der Online-Dienst UltraTools (Beispiel Facebook)
- **IP-Adressblöcke zu einem ASN**: Zu jedem ASN lassen sich dann, bspw. über RIPEstat, die dazugehörigen IP-Adressblöcke ausgeben

Dieser Vorgang wird vom Skript automatisch für alle jene Unternehmen ausgeführt, die ihr dem Skript als Parameter übergebt. Allerdings funktioniert das nur für Unternehmen, die eine gewisse Größe haben und international tätig sind. Ihr werdet also nicht für jedes Unternehmen Blocking-Regeln erzeugen können.

Hinweis

In der aktuellen Version des Skripts werden ausschließlich Blocking-Regeln für IPv4-Adressen generiert. Über IPv6 könnten demnach weiterhin Verbindungen zu Google oder Facebook initiiert werden. Solltet ihr das Skript also nutzen wollen, ist es sinnvoll, IPv6 in eurem Betriebssystem oder Router zu deaktivieren.

3.1 IPFire

Die Installation bzw. die Nutzung des Skripts auf der IPFire ist denkbar einfach und binnen weniger Minuten erledigt. Loggt euch zunächst mittels SSH auf eure IPFire ein und führt anschließend folgende Befehle aus:

```
cd ~
curl -O https://notabug.org/maloe/ASN_IPFire_Script/raw/master/asn_ipfire.sh
chmod 755 asn_ipfire.sh
```

Ruft anschließend die Hilfefunktion auf, um euch alle Parameter anzeigen zu lassen, die vom Skript unterstützt werden:

```
bash asn_ipfire.sh --help
```

Ausgabe der Hilfe:

```
Usage: asn_ipfire.sh [OPTION] [COMPANYs | -f FILE]
Add or remove networks to IPFire firewall Groups: Networks & Host Groups

Options:
  -a, --add   Add new company networks
  -r, --remove      Remove company networks from customnetworks &
customgroups
                    COMPANY='ALL' to remove all entries done by this script
  -f, --file FILE   Get company list from FILE
  -l, --list        List entries done by this script
      --renumber    Renumber lines of customnetworks & customgroups files
  -h, --help        Show this help

Create special output files (Non-IPFire-Mode):
  --network        Create FILE 'network_list.txt' with networks
```

```
  --network_raw    dito, but networks not consolidated
  --asn            Create FILE 'asn_list.txt' with ASNs only
  --iptable        Create FILE 'iptable_rules.txt' with iptable rules
  --afwall         Create FILE 'afwall_rules.txt' with afwall rules

COMPANY to be one or more company names, put into double quotes ('"')
        Multi company names can be comma or space separated
usage example: asn_ipfire.sh -a "CompanyA,CompanyB,CompanyC"
               asn_ipfire.sh --asn "CompanyA,CompanyB,CompanyC"

FILE = name of a file, containing one or more company names.
Company names to be separated by space or line feeds.
usage example: asn_ipfire.sh -u -f company.lst
               asn_ipfire.sh --network -f company.lst

Notes:
  Company names are handled case insensitive.
  Only entries made by asn_ipfire.sh are updated or removed.
  These entries are recognized by the 'Remark'-column in IPFire.

</file>

Mit folgendem Aufruf generiert ihr Netzwerkobjekte und Gruppen, für die
Unternehmen Google, Facebook, Twitter, Oracle und Acxiom, die sich
anschließend in der IPFire GUI als Blocking-Regeln (Verweigern (REJECT))
einbinden lassen:<code>bash /root/asn_ipfire.sh --add
"Google,Facebook,Twitter,Oracle,Acxiom"
```

Insbesondere Google, Facebook, Twitter und Oracle zählen zu den Unternehmen, die laut der Studie »Online Tracking: A 1-million-site Measurement and Analysis« auf mehr als 10% der Webseiten vertreten sind. Präziser:

- **Google**: Auf über 80% aller untersuchten Webseiten
- **Facebook**: Auf knapp 40% aller untersuchten Webseiten
- **Twitter**: Auf knapp 20% aller untersuchten Webseiten
- **Oracle**: Auf über 10% aller untersuchten Webseiten

Persönlich habe ich keine Berührungspunkte mit diesen Unternehmen, möchte also keine Services in Anspruch nehmen und möchte im Gegenzug auch nicht, dass diese Unternehmen Daten von mir erhalten. Das kann ich natürlich nur eingeschränkt kontrollieren – eben über die Blockade jeglicher IP-Adressen, die zu den Unternehmen gehören.

Öffnet nach der Ausführung des Skripts die GUI der IPFire und navigiert dort unter »Firewall → Firewallgruppen → Netzwerke«. In der nachfolgenden Abbildung könnt ihr sehen, dass die IP-Adressblöcke der Unternehmen bereits als **Netzwerkobjekte** angelegt sind:

Analog fasst das Skript jedes Unternehmen (alle IP-Adressblöcke) als **Gruppe** unter »Firewall → Firewallgruppen → Netzwerk-/Hostgruppen« zusammen:

Das Skript erledigt sozusagen die »Vorarbeit« und legt sowohl Netzwerkobjekte, als auch Gruppen an, die ihr dann nach belieben in eure Regelsätze einbauen könnt. Damit jeglicher Kontakt von euren Geräten / Netzen zu Google, Facebook, Twitter, Oracle und Acxiom unterbunden wird, könnt ihr folgende Regelsätze anlegen:

Wie ihr auf der Abbildung erkennen könnt, sind die Blocking-Regeln gleich am Anfang des Regelwerks gesetzt und werden als erstes abgearbeitet. Bei einem Treffer bekommt euer Client dann ein REJECT (Ablehnen) von der IPFire übermittelt, was für den Client bedeutet, dass das Ziel nicht erreichbar ist. Solch eine REJECT-Regel könnt ihr folgendermaßen unter »Firewall → Firewallregeln → Neue Regel erstellen« anlegen:

- **Quelle**: Wählt dort die entsprechenden Hosts, Gruppen oder Netze, denen ihr den Zugriff auf Unternehmen wie Google nicht erlauben wollt
- **Ziel**: Wählt dort das Unternehmen aus, das ihr blockieren möchtet. Das Skript hat bereits eine Gruppe (pro Unternehmen) angelegt, die ihr für diesen Zweck auswählen könnt
- **Protokoll**: Dort wählt ihr »Alle« und selektiert »Verweigern (REJECT)«
- **Weitere Einstellungen**: Damit ihr die Regel nicht ganz nach oben schieben müsst, könnt ihr als Regelposition die »1« vermerken und anschließend auf »Hinzufügen« klicken

Wenn ihr anschließend in einem Browser die Adresse »www.google.de« oder »www.facebook.com« aufruft, wird euch der Browser melden, dass die Adresse / Host nicht erreichbar ist. Mission accomplished!

Die IP-Adressblöcke sollten über das Skript regelmäßig aktualisiert werden. Über fcrontab könnt ihr das Updateverhalten auf der IPFire steuern:

```
fcrontab -e
```

Legt dort einen neuen Eintrag an, um bspw. jeden Tag um 23:35 Uhr das Skript auszuführen:

```
# 23 Uhr
# Update ASN network information
45 23 * * * bash /root/asn_ipfire.sh --add
"Google,Facebook,Twitter,Oracle,Acxiom" > /dev/null 2>&1
```

Das war es schon. Auf Wunsch könnt ihr weitere Unternehmen hinzufügen, entfernen oder über die »Remove-Funktion« auch alles wieder rückgängig machen:

```
bash /root/asn_ipfire.sh --remove ALL
```