“Survey: Black-Hat Search Engine Optimization (SEO) Practices for Websites”

Proseminar “Technische Informatik” unter der Leitung von Georg Wittenburg
Damla Durmaz, betreut von Norman Dziengel
Institut für Informatik - Freie Universität Berlin, Takustrasse 9, 14195 Berlin, Deutschland

0. Abstract

Diese Seminararbeit beschäftigt sich mit Methoden der Suchmaschinenoptimierung am Beispiel von Google, die bewusst gegen die Richtlinien einer Suchmaschine eingesetzt werden, um eine höhere Platzierung in den Rankings zu erhalten, auch als „Black-Hat SEO“ bezeichnet. Einige dieser Methoden werden noch immer benutzt und können, wenn sie unentdeckt bleiben, die Qualität der Suchergebnislisten beeinträchtigen. Sobald Google jedoch die Verwendung einer Black-Hat-Methode feststellt, wird die betreffende Seite aus der Datenbank gelöscht. Für Suchmaschinenoptimierer ist die Kenntnis dieser Black-Hat-Techniken wichtig, um sich gegen sie zu schützen oder sie nicht versehentlich einzusetzen. Neben der Vermittlung von wichtigen Kenntnissen zur Google Suchmaschinen-Architektur werden einige Black-Hat-Techniken vorgestellt, wie sie entdeckt werden, ihre Effektivität, Popularität und im Anschluss einige Schutzmaßnahmen. Es stellt sich heraus, dass in den meisten Fällen die Techniken mehr schaden können als auf lange Sicht die Platzierung zu erhöhen.

1. Einleitung

„Die Zahl der Internet-Nutzer steigt weltweit auf 1,2 Milliarden. 2010 werden voraussichtlich 1,5 Milliarden Menschen online sein“[1]. Dadurch wächst automatisch die Anzahl an Webseiten-Inhabern. Zudem ist das Internet vor allem für Unternehmen ein wichtiges Medium, da es neben vielen anderen konventionellen Werbemitteln eine wichtige Plattform für die Kommunikationspolitik ist und somit kommerziellen Zwecken dient. Doch durch die Größe des Internets und die damit verbundene Informationsflut ist es mittlerweile schwer, auf Suchmaschinen zu verzichten. Die Frage „Wie gestalte ich meine Webseite effizienter und bekomme mehr Leser?“ gewinnt also zunehmend an Bedeutung. Eine Seite, die über eine Suchmaschine nicht zu finden ist, wird meistens gar nicht gefunden. Die Optimierung kann also über Erfolg oder Misserfolg einer Seite entscheiden. Das Thema „Suchmaschinenoptimierung“ ist somit das Topthema unter Webseiten-Entwicklern. Dabei bedeutet „Suchmaschinenoptimierung“ nicht, eine Suchmaschine zu optimieren, sondern Webseiten für Suchmaschinen zu optimieren. Die wohl bekannteste Suchmaschine ist „Google“, gefolgt von „Yahoo!“ und „T-Online“. Jede Suchmaschine hat ihre eigenen Verfahren, eine Seite zu finden, von daher sollte man sich bei der Optimierung gut überlegen, von welcher Suchmaschine man „gefunden werden möchte“ und demnach optimieren.

Die Statistiken in Abbildung 1 und 2 zeigen, dass Google den größten Marktanteil besitzt und am stärksten wächst. Von daher ist es am sinnvollsten, die Suchmaschinenoptimierung auf Google zu richten, denn eine sehr hohe Platzierung in den Suchergebnislisten bei anderen Suchmaschinen wären den hohen Aufwand der Suchmaschinenoptimierung nicht wert.

Fig. 1: Prozentualer Vergleich der Nutzung der gängigsten Suchmaschinen (Quelle: http://www.webhits.de/deutsch/index.shtml?webstats.html)

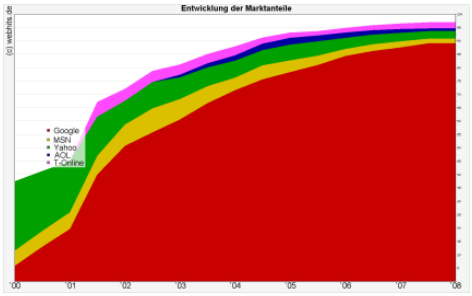

Fig. 2: Entwicklung der Marktanteile der gängigen Suchmaschinen (Quelle: http://www.webhits.de/artwork/ws_engines_historical_druck.png)

Dieses Papier beschäftigt sich mit Techniken der Suchmaschinenoptimierung, die gezielt gegen die Richtlinien der Suchmaschine die Platzierungen in den Ergebnislisten manipulieren. Diese Techniken werden auch als „Black-Hat“-Suchmaschinenoptimierung bezeichnet. Diese mögen zwar bei geringem Arbeitsaufwand die Rankings in kurzer Zeit zum eigenen Vorteil verändern, doch bei einer Übertreibung schlimme Folgen für die Seite haben, im schlimmsten Fall die Verbannung aus dem Datenbestand von Google.

Einige White-Hat Techniken – erlaubte Optimierungstechniken- sind effektiv und können erfolgsversprechend sein, doch bei zu starker Anwendung sich wiederum negativ auf die Platzierung auswirken, da sonst jeder Webseiten-Entwickler diese Methode im hohen Maße anwenden würde und somit keine realistischen Platzierungen entsprechend der Qualität von Seiten möglich wären.

Im 1. Kapitel wurde erläutert, warum die Suchmaschinenoptimierung so wichtig ist. Im 2. Kapitel wird Grundlagenwissen über Suchmaschinenoptimierung vermittelt. Zuerst wird der Begriff „Suchmaschinenoptimierung“ definiert (Kapitel 2.1) und in Kapitel 2.2 wird beschrieben, wie Google Webseiten findet

und ihre Platzierung ermittelt. Im Kapitel 2.3 werden die wichtigsten White-Hat-Techniken vorgestellt. Im 3.Kapitel werden einige Black-Hat-Techniken vorgestellt. Für jede Technik wird ihre Idee erläutert, die technischen Details hervorgehoben, gefolgt von einer zusammenfassenden Bewertung der Technik, die sich neben der Popularität und Effektivität auch damit beschäftigt, wie Google den Einsatz der Technik verhindert. Im 4. Kapitel wird die Problematik der Verwendung von Black-Hat-Techniken verdeutlicht, da Google nicht jede Art von Technik eindeutig erkennen kann. Zudem wird auch aufgeklärt, wie der versehentliche Einsatz von Black-Hat-Techniken verhindert werden kann und wie man sich vor den Techniken schützen kann.

2. Grundlegendes über Suchoptimierung

2.1 Was ist Suchmaschinenoptimierung?

„Unter Suchmaschinenoptimierung oder kurz „SEO"(„search engine optimization") versteht man allgemein alle Praktiken und Techniken, die dazu führen, dass eine Webseite oder ein Teil einer Webseite in den Ergebnislisten(auch SERP für „search engine results page" genannt) der gängigen Suchmaschinen besser, d.h. weiter vorn, gelistet wird."[2] Dieser Zusammenhang sollte nicht verwechselt werden mit der Alternative, sich hohe Platzierungen in den Ergebnislisten einer Suchmaschine zu „erkaufen", auch bekannt unter dem Stichwort „Payed Placement". Diese Verweise werden dann in den Suchergebnislisten hervorgehoben und in vordefinierten Bereichen angezeigt.

Die Logik einer Suchmaschine besteht aus zwei aufeinander bauenden Komponenten. Die eine Komponente umfasst eine Menge an kleinen, automatisch gesteuerten Programmen, die Verlinkungen in registrierten Seiten folgen, ihre Inhalte archivieren, diese analysieren und Berechnungen durchführen und diese in einer Datenbank, dem Index, zu speichern. Diese Programme werden als Robots, Spiders oder Crawler bezeichnet (s. Kapitel 2.3.1). Der Robot besucht wie ein automatisch gesteuerter Internet-Benutzer eine Webseite und verfolgt alle Links, speichert Inhalte, bis es keine weiteren internen Verlinkungen mehr findet. Bei Google werden die Daten im Googleindex, die Datenbank von Google, gespeichert. Dabei wird nicht immer ein globaler Index verwendet. Während Google neben anderen Suchmaschinen einen eigenen Datenbestand hat, gibt es Suchmaschinen, die die Suchanfrage an andere Suchmaschinen senden (z.B. Metasuchmaschinen), die Ergebnisse aufbereiten und diese dann dem Suchenden vorlegen.

Die zweite Komponente nimmt die Suchanfrage entgegen und sucht im Index mittels Vergleichen und schon vorher vergebenen Platzierungen nach Seiten, die für die Suchanfrage als „passend" eingestuft werden. Die Sortieralgorithmen, die genau diese Vergleiche machen und die Wichtigkeit einer Seite berechnen, werden von den Suchmaschinenbetreibern streng vor der Öffentlichkeit geschützt und stets an die Neuentwicklungen des WWW angepasst.

Das Ziel der Suchmaschinenoptimierung ist, auf lange Sicht eine möglichst hohe Platzierung bei Google zu erhalten. Eine einmalige Optimierung reicht demnach nicht aus, denn die Bewertungsalgorithmen ändern sich ständig. Anhand vieler Studien, Wettbewerben[3] und erfolgreichen Optimierungen gibt es Vermutungen, wie die Algorithmen ungefähr bewerten.

2.2 Ermittlung des Rankings von Webseiten durch Google

Die Platzierung einer Webseite in den Google-Suchergebnissen wird anhand bestimmter Kriterien ermittelt. Webseiten-Entwickler können neben vielen Optimierungsmethoden, die am Inhalt und der Umgebung der Seite vorgenommen werden, ihre Platzierung erhöhen, wenn hochplatzierte Seiten auf diese verweisen. Bis die Ergebnisse einer Suchanfrage bei Google jedoch angezeigt werden, sind wichtige Zwischenschritte erforderlich:

- Das Webcrawler-System
- Indexierung
- Ermitteln der Platzierung

2.2.1 Das Webcrawler-System

Um aktualisierte oder neue Seiten zu finden, werden Robots ins Internet verschickt. Dieser Schritt wird als *Crawling* bezeichnet. Der Crawler von Google, der *Googlebot*, arbeitet nach einem festgelegten Algorithmus. Er erhält eine Liste an URLs, die schon bei vorigen Besuchen des Googlebot auf Webseiten generiert wurde. Diese teils schon im Index vorhandene Seiten besucht er, erkennt, ob sie neu oder aktualisierte Webseiten sind und archiviert diese.

Die Häufigkeit des Besuches vom Googlebot hängt von einem durch Google übergebener Bewertung, dem PageRank, ab(s. Kapitel 2.2.3). Je öfter eine Seite mit neuen Inhalten aktualisiert wird, desto mehr Gründe hat der Googlebot, diese erneut zu besuchen[4]. Damit auch nicht verlinkte Webseiten in den Index aufgenommen werden können, bietet Google eine manuelle Seitenanmeldung an. Das Crawlen ist jedoch nicht der einzige Schritt bei der Datengewinnung. Auch weitere Prozesse, die das Crawlen koordinieren oder die aus den gecrawlten Seiten etwas anfangen, spielen eine große Rolle. Die verschiedenen Prozesse werden zu Modulen zusammengefasst und die Gesamtheit aller Module wird als das Webcrawler-System bezeichnet. Zu den drei Kernkomponenten dieses Systems gehören die Protokollmodule (Crawler), die Verarbeitungsmodule (Scheduler, Store Server) und die Datenspeichermodule (Dokumentenindex, Repository). Fig. 3 beschreibt graphisch die Suchmaschinenarchitektur von Google. Die rotumrandeten Formen bilden zusammen das Webcrawler-System.

Zunächst wird die Ausgangssituation beschrieben. Im *Doc Index*, eine Datenstruktur, liegen URLs mit ihren zugeordneten docIDs. Wie diese zugeordnet werden, wird im Verlauf dieses Kapitels klar. Jede Internetseite bekommt solch eine docID. Die Arbeit beginnt bei dem *URL-Server*. Dieser holt sich aus dem Doc Index eine Liste an URLs, die aus vorherigen Crawling-Durchläufen generiert wurden. Die Liste wird vom URL-Server heruntergeladen und an den Googlebot übergeben[5]. Die Übergabe der URLs wird von einer zentralen Steuereinheit, dem *Scheduler*, erledigt. Dieser teilt den Googlebots ihre Aufgaben zu und kontrolliert somit den Status des Systems. Der Googlebot besucht nun diese Seiten. Jede Seite wird mit den Einträgen im sogenannten *Dokumentenindex* verglichen. Dabei handelt es sich um eine Datenstruktur, in der Metainformationen zu allen heruntergeladenen Webdokumenten gespeichert sind.

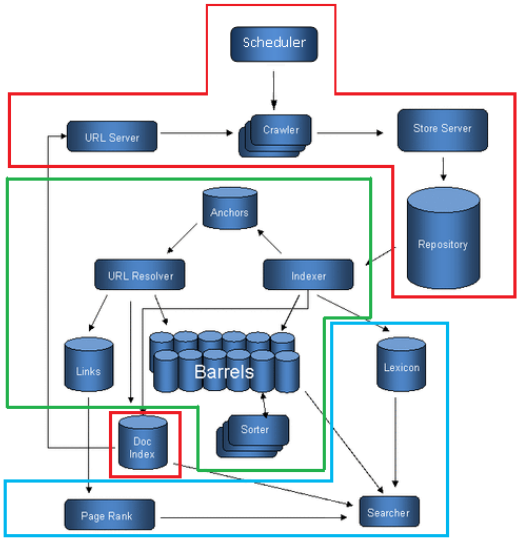

Fig. 3: Darstellung der Google Architektur[6] – rotes Feld: Webcrawler-System, grünes Feld: Indexierung, blaues Feld: Ermitteln der Platzierung

Dazu gehören unter anderem Titel, eine eindeutige ID, die sogenannte *docID*[7](die Einträge werden nach der docID geordnet), Größe, Einstellungsdatum, IP und Hostname des Servers und ein Verweis auf das Repository. Ob ein Eintrag im Dokumentindex enthalten ist, wird anhand einer Variablen überprüft. Wenn das Dokument schon einmal gecrawlt wurde, so enthält es einen Zeiger auf eine Variable, namens *docinfo*, die die URL und den Titel der Seite enthält. Ist ein Eintrag im Dokumentindex nicht enthalten(zeigt also der Zeiger nur zur URL Liste), so wird das durchsuchte Webdokument heruntergeladen.

Die heruntergeladenen Webdokumente werden an ein Programm, namens *Store Server*, weitergegeben[8]. Der Store Server liest die erhaltenen Informationen und aktualisiert den Dokumentenindex. Anschließend werden die Daten mithilfe der Programmbibliothek *zlib* komprimiert und an das *Repositoriy* weitergegeben, wo sie gespeichert werden. Währenddessen erhalten alle Webdokumente, die noch nicht betrachtet wurden, eine docID zugewiesen, die aus der URL des Webdokuments nach einem bestimmten Algorithmus erstellt wird. Das Repository ist eine Datenstruktur und speichert den Quellcode, die Größe, die URL der HTML-Datei und einen Verweis auf den Eintrag der Metainformationen des Webdokuments im Dokumentindex.

2.2.2 Indexierung

In der zweiten Phase werden die gecrawlten Seiten indiziert. Die Indexierung erfolgt durch den *Indexer* und durch die Sorter. Der Indexer holt die Webdokumente aus dem Repositoriy, dekomprimiert diese und beginnt mit der Analyse. Mithilfe eines *Parsers* werden die Webdokumente in eine Liste von Wortvorkommen auf der Seite umgewandelt. Die Häufigkeit des Wortvorkommens wird als Hit bezeichnet. Diese Liste wird auch als *Hitlist* bezeichnet und beinhaltet zudem auch Informationen über Formatierung der Wörter. All diese Hits werden nun auf viele Datenstrukturen, den sogenannten *Barrels*, verteilt. Dadurch entsteht ein sogenannter *Forward Index*, eine bestimmte Art von Indexierung, bei der die Wörter von Dokumenten, die gerade vom Parser analysiert werden, sofort abgelegt werden. So wird der Grad der asynchronen Abarbeitung der Listen erhöht.

Zudem analysiert der Indexer alle vorkommenden Links einer Seite und speichert diese und weitere Informationen in einer *Anchors-Datei*. Alle Anchors-Dateien zusammen werden in der Datenstruktur *Anchors* gespeichert.

Aus den relativen Links in der Anchors-Datei werden nun mithilfe des *URL Resolvers* absolute URLs ermittelt. Diesen werden gleichzeitig bei der Umwandlung docIDs zugewiesen und als neue Informationen zusammen mit dem Linktext dem Forward Index hinzugefügt. Währenddessen erzeugt der URL Resolver eine Datensammlung aus Links, die dann später vom PageRank-Algorithmus benutzt wird, um die Relevanz der Webdokumente zu bestimmen.

Der *Sorter* greift auf die Daten in den Barrels zu und sortiert diese nach den *wordIDs*. Bei dieser Sortierung entsteht der sogenannte invertierte Index (Inverted Index). Bei Google hat der Inverted Index den Vorteil, dass er nach Wörtern sortiert ist und nicht nach Identifikationsnummern, sodass die Suche von Wörtern innerhalb des Indexes beschleunigt wird. Anschließend erstellt das Programm „DumpLexicon“ mithilfe des Inverted Indexes zusammen mit dem *Lexicon* ein neues aktualisiertes *Lexicon*, das viele Zeiger beinhaltet, die alle auf das jeweilige Barrel, in dem sich die jeweilige wordID befindet, zeigen.

Die Rolle des Inverted Indexes und des Lexicons wird erst im dritten Schritt, im eigentlichen Festlegen der Platzierung einer Seite bei Google, deutlich, denn ein Programm namens *Searcher* benutzt den Inverted Index und das Lexicon, um zusammen mit den PageRank-Werten vom *PageRank-Server* einer Seite die Suchanfrage in der Suchmaschine zu beantworten[9].

2.2.3 Ermitteln der Platzierung

Nachdem ein Webdokument indexiert wurde, muss es bewertet werden. Bevor ein Webdokument für Google optimiert wird, ist es grundlegend, die Gewichtungsmodelle von Google zu kennen. Viele unterschiedliche Suchmaschinen benutzen ähnliche Modelle zur Ermittlung der Platzierung, doch was Google von den anderen stark unterscheidet, ist ihr Platzierungsalgorithmus, der sogenannte PageRank-Algorithmus.

Die Platzierung eines Dokuments in der Trefferliste wird als *Ranking* bezeichnet und beschreibt dabei den Grad der Ähnlichkeit eines Dokumentes für ein Suchwort. Jede Suchmaschine hat dabei unterschiedliche Gewichtungsmodelle. Der PageRank-Wert lässt sich algorithmisch mit einer mathematischen Gleichung berechnen, die nach Angaben von Google mehr als 500Millionen[10]Variablen und zwei Milliarden Ausdrücken hat. Das Ergebnis ist ein numerischer Wert. Der PageRank-Algorithmus betrachtet nicht nur die zu bewertende Seite, sondern unter anderem auch die Backlinks (also Verweise von außen) und den PageRank der Backlinks. Daraus folgt, dass sich der PageRank einer Seite automatisch erhöht, sobald die Backlinks selbst einen hohen PageRank haben.

Im einfachsten Fall lässt sich die Platzierung eines Webdokuments mithilfe der *Linkpopularität* ermitteln. Dabei handelt es sich um die Anzahl aller Backlinks plus alle internen Links und die Linkqualität[11]. Diese wird allerdings

immer von Suchmaschine zu Suchmaschine anders ermittelt. Zudem ist die Linkpopularität anfällig auf Manipulationen(z.B. Erstellen von unzähligen sinnlosen internen Links).

2.2.3.1 *Der PageRank-Algorithmus*

Im Folgenden wird die Grundidee des PageRank-Algorithmus vorgestellt, so wie er in den ersten vorgelegten Dokumentationen von Google beschrieben wird. Die mathematische Gleichung für die Ermittlung des PageRanks ist wie folgt definiert:

$$PR(A) = (1-d) + d\left(\frac{PR(D_1)}{C(D_1)} + \frac{PR(D_2)}{C(D_2)} + \cdots + \frac{PR(D_n)}{C(D_n)}\right) \quad (1)$$

wobei gilt:

$PR(A)$	PageRank-Wert des Webdokuments A berechnet aus allen eingehenden n Verweisen
A	*zu bewertendes Webdokument*
d	ein Dämpfungsfaktor, der zwischen 0 und 1 liegt(erfahrungsgemäß um 0,85)
$PR(D_1)$	Der PageRank des Dokuments D1, der auf A verweist
$(1-d)$	Wahrscheinlichkeit, dass ein Besucher die Seite verlässt
$C(D_n)$	Anzahl der Verweise, die von D_n ausgehen

Der PageRank, der für A ermittelt werden soll, wird also unter Berücksichtigung aller Verweise ermittelt, die auch von D_n ausgehen. Der PageRank wird nicht einmalig erfasst, sondern erfolgt iterativ. Der Dämpfungsfaktor d dient dazu, das Ergebnis zu "verfeinern"[12], denn so "vererbt" das Dokument D_i nicht seinen vollen PageRank an A. Der Term $(1-d)$ ist ein Gegenereignis und sagt aus, dass ein Benutzer nach einer Linkverfolgung auf der aktuellen Seite anhält und mit einer Wahrscheinlichkeit von $(1-d)$ diese verlässt. Im Folgenden ist ein einfaches Beispiel ohne Berücksichtigung des Faktors d:

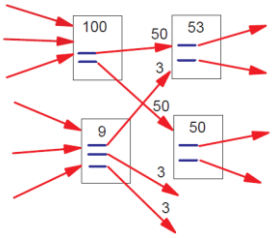

Fig. 4: Einfaches Beispiel für eine PageRank-Berechnung[13]

Die Zahlen in den Kästen in Fig. 4 stellen den PageRank-Wert des jeweiligen Dokuments dar. Das erste Dokument hat den PageRank-Wert 100, das zweite den Wert 53 usw. Die roten Pfeile stellen einen Hyperlink auf das jeweilige Dokument dar und die Werte über den Pfeilspitzen geben an, welcher PageRank-Wert „vererbt“ wird. Betrachten wir das erste Dokument. Es hat einen PageRank-Wert von 100 und verweist auf zwei weitere Seiten, somit wird den beiden Zieldokumenten der Wert 50 zugewiesen. Das zweite Dokument erhält vom ersten Dokument einen Wert von 50. Da jedoch auch das dritte Dokument auf das zweite verweist, erhält es noch zusätzlich den Wert 3, also insgesamt einen PageRank-Wert von 53. Dieses Weiterreichen eines bestimmten Wertes des eigenen PageRank-Wertes auf Zieldokumente ist die Stärke des PageRank-Verfahrens, da so teilweise sehr unterschiedliche Werte dabei herauskommen.

Man kann den PageRank als den Erwartungswert betrachten, dass die Seite bei $N = Anzahl\ der\ Seiten\ im\ Web$ Anläufen besucht wird. Durch diese Idee lässt sich die Gleichung modifizieren:

$$PR(A) = \frac{(1-d)}{N} + d\left(\frac{PR(D_1)}{C(D_1)} + \frac{PR(D_2)}{C(D_2)} + \cdots + \frac{PR(D_n)}{C(D_n)}\right) \quad (2)$$

Und so kann die Formel immer weiter modifiziert werden. Allerdings steigt damit der Rechenaufwand. Betrachten wir im Folgenden ein Beispiel[14] (siehe Fig. 5) mit einem Dämpfungsfaktor d = 0,5. Nach der Formel (1) ergeben sich folgende Berechnungen:

$$PR(A) = 0.5 + 0.5\,PR(C)$$
$$PR(B) = 0.5 + 0.5\left(\frac{PR(A)}{2}\right)$$
$$PR(C) = 0.5 + 0.5\left(\frac{PR(A)}{2} + PR(B)\right)$$

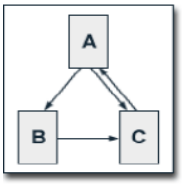

Fig. 5: Ein kleines WWW. Die Kästchen stellen jeweils Seiten dar. Berechnet werden soll ihr PageRank-Wert und zwar in mehreren Iterationen. Wie viele es genau sind, ist nicht bekannt.

In der nächsten Iteration werden die berechneten PageRank-Werte erneut benutzt. In Fig.6 werden beispielhaft 12 Iterationen durchgeführt.

Iteration	PR(A)	PR(B)	PR(C)	Sum
0	1	1	1	3
1	1	0.75	1.125	2,875
2	1.0625	0.7656	1.14844	2,977
…				
11	1.07692307	0.76923077	1.15384615	~3
12	1.07692308	0.76923077	1.15384615	~3

Fig. 6: Am Ende der 12 Iteration verändern sich die Werte kaum noch. Die Summe aller PageRank-Werte ist genauso groß wie die Summe der Werte in der Ausgangssituation 0

Die Idee, die Rankings abhängig von der Anzahl an Verweisen nach außen und auf die eigene Seite zu betrachten, wird gerechtfertigt mit dem „*Random Surfer Model*“[15]. In diesem Modell wird eine Art Internetbenutzer beschrieben, der eine Seite nach der anderen besucht, indem er immer den Verweisen folgt. Die Wahrscheinlichkeit, dass er von Seite A auf Seite B gelangt, hängt davon ab, wie viele Verweise von A überhaupt ausgehen. Damit ergibt sich dann eine Summe an Wahrscheinlichkeiten, mit der der Benutzern von Seite A, nach B, nach C, usw. gelangen wird. Mit dem Dämpfungsfaktor wird dieser Wert verfeinert, da der Benutzer nicht unendlich lange surft.

Zusammengefasst, können eingehende Links den PageRank erhöhen. Allerdings können ausgehende Links den PageRank der eigenen Seite verringern[16]. Betrachtet wird das Beispiel-WWW in Fig. 5. Die Summe der PageRank- Werte ist in jedem Iterationsschritt ungefähr 3 (vgl. Fig. 6). Genauso verhält es sich im realen WWW. Die PageRank-Werte einer Seite zusammen mit ihren verweisenden Seiten und

eingehenden Links ergeben zusammen eine Struktur, bei der die Summe aller PageRank-Werte zusammen einen konstanten Wert ergeben. Bei einem eingehenden Link gewinnt das eigene Dokument mehr PageRank dank des Backlinks. Verweist es auf ein anderes Dokument, so „vererbt“ es einen Teil seines PageRanks weiter. Da die Summe aller Werte gleich bleiben muss, wird der PageRank-Wert einer Seite somit geringer, sobald sich der Wert einer anderen Seite erhöht. Deshalb ist es ratsam, externe Links auf einer Seite mit geringerem PageRank zu setzen, als z.B. auf der Startseite der Webseite, da diese den höchsten PageRank hat.

2.2.3.2 Keyworddichte und inverse Dokumenthäufigkeit

Zwei weitere Verfahren, um Webdokumente zu gewichten, bilden die Ermittlung der Keyworddichte und der inversen Dokumenthäufigkeit.

Unter dem Begriff Keyword[17] versteht man im Allgemeinen ein Wort, der von einem Besucher in eine Suchmaschine eingegeben wird, um eine für seine Anfrage passende Webseite zu finden. Bei der *Keyworddichte* handelt es sich um ein Verhältnis zwischen der Häufigkeit eines Keywords und der Gesamtzahl der Wörter im Dokument. Je größer diese Dichte ist, desto wichtiger ist das Keyword für den gesamten Inhalt des Textes.

$$KWD = \frac{Häufigkeit\ eines Begriffs\ im\ Dokument}{Anzahl\ aller\ Wörter\ im\ Dokument}$$

Ist die Keyword-Dichte allerdings zu hoch, so besteht der Verdacht auf Keyword-Stuffing (s. Kapitel 3.1), bei der mit Absicht die Keyworddichte hochgesetzt wird. Von daher gibt es bei jeder Suchmaschine bestimmte Grenzen, meistens ca. 5%-10%[18].

Ein weiteres Kriterium ist die *inverse Dokumenthäufigkeit.* Hier wird die Relevanz eines Keywords in einem Webdokument umso höher gewertet, je seltener es in anderen Dokumenten vorkommt, was bedeutet, dass Dokument A wahrscheinlich passender ist als Dokument B, da das Keyword in A öfters auftaucht.

2.2.3.3 Lage und Format von Keywords

Wichtige Keywords würde der Autor eher zu Beginn des Textes positionieren. Vielleicht würde der Autor diese durch besondere Farben hervorheben, um ihre Relevanz zu unterstreichen. Vorteilhaft ist es auch, Keywords im Titel des Dokuments zu positionieren. Google hat für Format und Lage Klassen eingeführt, die bestimmte Voraussetzungen erwarten. Ein Keyword, der z.B. unter den ersten 100 Wörtern auftaucht, würde eine höhere Bedeutung haben, als wenn erst ab dem 250. Werden mindestens zwei Suchbegriffe eingegeben, wird das Proximity-Verfahren eingesetzt. Die Idee des Verfahrens beruht auf die empirische Beobachtung, dass zwei Keywords, deren Abstand zueinander viel geringer ist, dem zugrundeliegenden Text eine höhere Bedeutung schenken[19].

2.3 Grundlegende Techniken der Suchmaschinenoptimierung

2.3.1 Aufnahme in den Index von Google

Ein Weg zur Indexierung ist die automatische Aufnahme in den Google-Index. Im Kapitel 2.2.1 wurde beschrieben, wie der Googlebot neue Seiten findet. Ein Weg zur Indexierung wäre eine hohe Linkpopularität und ein guter PageRank-Wert[20]. Eine zweite Methode ist die manuelle Anmeldung der Seite bei Google (unter http://www.google.de/addurl). Die dritte Variante wird von Google unter den Namen „Google Sitemaps“ angeboten. Eine Sitemap ist im Allgemeinen „eine Art Inhaltsverzeichnis einer Webseite“[21] und enthält eine Liste aller Links auf der Webseite. Es gibt die Möglichkeit, schon fertig formatierte Sitemap-Dateien zu bearbeiten und diese an den Webserver zu senden. Nützlich ist Google-Sitemaps deshalb, weil es gute Statistik-Optionen für den Lebenslauf einer Seite anbietet.

2.3.2 Zu Beginn der Suchmaschinenoptimierung – Ziele definieren

Zu Beginn der Optimierung müssen passende Keywords gewählt werden. Dabei kann diese Wahl von großer Bedeutung für den zukünftigen Erfolg oder Misserfolg der eigenen Webseite spielen. Eine beliebte Methode zum Finden von passenden Keywords ist, sich die ersten zwei wichtigsten Keywords auszusuchen, die die eigene Webseite beschreiben und diese bei Google einzugeben. Anschließend kann der Optimierer die hochplatzierten Seiten auswählen und sich über die Metainformationen dieser über die gewählten Keywords informieren. Zudem gibt es auch Keyword-Tools, die kostenlos erhältlich sind (z.B. Overture/Yahoo Search Marketing). Dabei wird ein Suchbegriff eingegeben und es werden Keywords aufgelistet, die nach Relevanz geordnet sind. Sinnvoll sind auch Kombinationen von mehreren Keywords, da damit ihre inverse Dokumenthäufigkeit steigt, denn es ist unwahrscheinlicher, dass sie genau so auch woanders aufzufinden sind.

2.3.3 Onpage-Optimierung

„Unter der Onpage-Optimierung fasst man alle Optimierungsmethoden zusammen, die man direkt an der eigenen Site vornehmen kann, die nicht von außen beeinflussbar sind und somit ausschließlich in den eigenen Verantwortungsbereich fallen.[22]“

Im *Dokumententitel* (<title>-Tag) sollten alle wichtigen Keywords enthalten sein. Zusätzlich sollte auch das Proximity-Verfahren bedacht werden(siehe Kapitel 2.2.3.3). Es wäre also effektiver, mehrere Keywords, die im Titel vorkommen, möglichst nah beieinander zu setzen, um so ihre Relevanz zu untermauern.

Im *Dokumentkörper*(<body>-Tag) ist der gesamte Fließtext und muss zum Inhalt die passenden Keywords enthalten. Dabei sollte stets auf die *Keyworddichte* geachtet werden. Wenn die Dichte zu klein ist, verliert das Dokument an Relevanz. Wenn die Dichte allerdings zu hoch ist, besteht die Gefahr, dass Google das als Spam bewertet(s. Kapitel 3.1).

Zudem eignet es sich auch besonders gut, Keywords mit Formatierungstag hervorzuheben, wie z.B. die *Heading-Tags*. Wenn ein Keyword z.B. in einer Überschrift mit der Formatierung <h2> vorkommt, so hat das positive Auswirkungen auf die Relevanz der Seite[23].

Des Weiteren können auch Grafiken auf der eigenen Webseite verwendet werden, allerdings sollten wichtige Informationen trotzdem als reinen Text vorliegen, da Google Informationen aus einer Grafik nicht lesen kann. Optional empfiehlt es sich, das Alt-Attribut `<img`

src="Wiese.png" alt="Wiese"> zu benutzen, mit dessen Hilfe kurz der Inhalt einer Grafik beschrieben werden kann. Effektiv dabei wäre natürlich die Verwendung von Keywords im Alt-Attribut. Zusätzlich kann auch das Title-Attribut verwendet werden mit <img src="Wiese.png" alt="Wiese" title="Wiese">. Der Text im Title-Attribut erscheint dann dem Benutzer, wenn er den Mauscursor über die Grafik bewegt.

2.3.4 Offpage-Optimierung

Alle Optimierungstechniken in diesem Kapitel werden als Offpage-Optimierungen bezeichnet. Hier hat der Webseitenbetreiber keinen direkten Einfluss auf die Optimierung, es betrifft also nicht Inhalt, sondern die Umgebung der Seite[24].

Die *Verzeichnisstruktur* ist ein guter Optimierungsansatz. Die Startseite einer Webseite (Root-Ebene) wird am stärksten gewichtet, während die Gewichtung von Unterebene zu Unterebene sinkt[25]. Je mehr Unterebenen eine Seite besitzt, desto seltener werden die Inhalte vom Googlebot erfasst aufgrund der niedrigeren PageRank-Werte der Unterseiten.

Mehrere Verweise von einer gleichen Domain werden öfters sogar dann nur als einen Verweis gezählt[26]. Dieser Aspekt wird als Domainpopularität bezeichnet und sollte bei der Optimierung genauso bedacht werden wie die Linkpopularität. Die IP-Popularität ist ein weiterer Optimierungsansatz. Diese betrachtet die Anzahl der Backlinks unter der Berücksichtigung der IP-Adresse.

3. BLACK-HAT SUCHMASCHINENOPTIMIERUNG

Das Nutzen von Black Hat-Techniken kann zur Löschung aus dem Google-Index führen, wenn Google die Praktizierung erkennt. Im Folgenden sollen einige, teils populäre Black – Hat – Techniken detailliert vorgestellt werden. Doch vorher soll geklärt werden, was Spam ist. Den meisten Internetbenutzern ist unter Spam das massenhafte Versenden von nutzlosen Nachrichten via E-Mail bekannt. Im Zusammenhang mit der Suchmaschinenoptimierung wird unter Spam alles verstanden, was für den menschlichen Benutzer nutzlos ist und nur dem Zweck der Rankingmanipulation dient[27].

3.1 Keyword-Stuffing

3.1.1 Grundlegende Idee

Beim Keyword-Stuffing wiederholt der Webseiten-Betreiber seine Keywords übermäßig oft in seinem Webdokument, um die Keyworddichte zu erhöhen und eine bessere Gewichtung zu bekommen.

3.1.2 Technische Details

Jedes HTML-Dokument hat ein Grundgerüst:

```
<html>
<head>
    <title>Titel</title>
</head>
<body>
<meta name="description" content="Beschreibung">
</body></html>
```

Im <head>-Bereich werden unter anderem die Metaangaben und der Titel der HTML-Seite definiert. Die Metainformationen beinhalten wichtige Aussagen über die Seite, wie Autor, Sprache oder Erstellungsdatum. Der description-Tag dient eigentlich vor allem der Suchmaschine als kurze Beschreibung der Webseite. Im Kapitel 2.3.3 wurde beschrieben, wie im HTML-Code eine Seite optimiert werden kann, zum Beispiel durch das Auftauchen von Keywords im Titel, in den Metaangaben oder in der Überschrift. Bei einem HTML-Code über das Thema „Kredit mit geringen Zinsen“, bei der der Webseiten-Betreiber Keyword-Stuffing durchführen möchte, würde z.B. folgendermaßen aussehen:

```
<html>
<head>
<meta name=”keywords” content =“Kredit geringe
Zinsen“>
<title> Kredit geringe Zinsen</title></head>
<body> <h1> Kredit geringe Zinsen</h1>
<img src=“bild.gif“ alt=“ Zinsen“ title=“
Kredit“/></body></html>
```

Hierbei denkt der Webseiten-Entwickler, dass durch das ständige Wiederholen der Keywords im gesamten Dokument eine hohe Keyword-Dichte, ein gutes Design und somit eine bessere Gewichtung zustande kommt.

Manche Webseiten-Betreiber gehen einen Schritt weiter und benutzen das Keyword-Stuffing bei Konkurrenten. Dabei nutzen sie die öffentliche Kommentarfunktion, Gästebücher und Forenbereiche und überfluten diese mit den entsprechenden Keywords. Das hat teilweise enorm negative Folgen für den Betroffenen, da er dadurch unter den Verdacht von Keyword-Stuffing gerät.

Bei einer anderen Variante, dem *Hidden-Text*, handelt es sich um eine ziemlich veraltete Technik, um die Keyword-Dichte zu erhöhen. Dabei wird mithilfe des <font>-Tags die Farbe eines Textes an die Hintergrundfarbe angepasst. So kann der Leser meistens die ganzen Keywords lesen. Eine weitere Möglichkeit, den Text zu verstecken, besteht darin, die Textgröße so klein wie möglich einzustellen, sodass der Besucher den Text übersieht. Kombiniert der Webseiten-Betreiber diese Größe noch mit der gleichen Hintergrundfarbe, so wird der Text ganz einfach nicht wahrgenommen. Manchmal werden auch Texte in sehr kleiner Schriftgröße hinter Bildern versteckt, die massenhaft Keywords in sinnloser Reihenfolge enthalten. Eine verbesserte Variante des Hidden-Textes ist das Verwenden von Farbnuancen, die der Besucher kaum bis gar nicht wahrnimmt. So wird oft die Kombination „Hellgrauer Text auf weißen Hintergrund“ benutzt und wird vom Besucher nicht wahrgenommen.

3.1.3 Zusammenfassende Bewertung

3.1.3.1 Wie verhindert Google den Einsatz?

Google ist diese Art von Spam-Methode lange bewusst. Von daher spielt bei der Gewichtung von Webseiten der Meta-Bereich keine Bedeutung mehr. Sie wird nicht direkt zur Gewichtung benutzt, sondern dient viel mehr der zusammenfassenden Beschreibung einer Seite.

Um den Einsatz dieser Technik zu entdecken, schaut Google sich das Verhältnis von auftauchenden Substantiven und Nichtsubstantiven an und untersucht, ob dieser einen Sinn ergibt. Z.B. besteht ein deutscher Satz aus einem Nomen, Prädikat und einem Objekt. Wenn z.B. zu wenig oder gar keine Prädikate existieren, so ist die Seite unter Spam-Verdacht, weil wahrscheinlich viele Keywords hintereinander

gereiht. Zudem hat Google auch eigene Keyworddichte-Grenzen, die bei ca. 8% liegen[28].

Eine Variante zur Formatierung der Webseiteninhalte bilden die „Cascading Style Sheets“ (kurz: CSS) ist für Google allerdings ein Problem. Bei der Verwendung von CSS-Stylesheets können diese Formatierungen in einer externen Datei gespeichert werden, bei der Google unter Umständen Probleme hat, diese korrekt zu interpretieren[29]:

```
CSS-Anweisung:
.hidden {display: none}
HTML-Tag:
<a href=”extern.htm” class=“hidden”>Text ist nicht
mehr sichtbar</a>
```

Google muss auch diese Dateien indexieren und sie auslesen, was nicht immer korrekt abläuft. Allerdings gab es im Jahr 2005 Reaktionen aus Seiten von Google[30], wo festgesetzt wurde, dass bestimmte CSS-Formatierungen nicht mehr gelesen werden, wie z.B. `display none` und andere Wege, Texte zu verstecken.

Problematischer ist es, wenn der Webseiten-Betreiber gezielte Anweisungen in der robots.txt gibt, sodass der Googlebot die CSS-Dateien nicht lesen kann. Dabei handelt es sich um eine einfache Textdatei, die dem Googlebot bzw. einem Crawler Anweisungen zum Durchsuchungsverhalten enthält. Diese Datei befindet sich logischerweise im Stammesverzeichnis einer Webseite. Bei der Aktualisierung einer Seite ist man z.B. mit der Optimierung noch nicht fertig und möchte dem Googlebot anweisen, dass er den gerade zu bearbeitenden Inhalt noch nicht lesen soll. Solche und andere Anweisungen können einem Bot mitgeteilt werden:

```
User-agent: * //alle Bots werden angesprochen
Disallow: /privat/
//Hier wird allen Bots verweigert, das Verzeichnis
„privat“ zu lesen
```

Auch den Zugriff auf die komplette Seite kann man durch Anweisungen verhindern. Es gibt noch eine weitere Möglichkeit, Dateien vor Zugriffen zu schützen mithilfe von .htaccess – Dateien (eine Konfigurationsdatei, die im Root-Verzeichnis des Webservers liegt). Allerdings sollten diese Dateien vorsichtig benutzt werden, da nicht alle Webhoster-Provider diese Variante unterstützen.

3.1.3.2 Popularität und Effektivität

Mittlerweile ist das Erkennen von Hidden-Text leicht. Trotzdem wird es immernoch häufig verwendet. Z.B. wurde am 25.Juli 2007 in einem Blog unter der Domain www.money-machen.de ein Beitrag veröffentlich, bei dem über solch einen Fall berichtet wurde. Da zudem auch die bei wachsendem WWW die Verwendung von CSS-Layouts zur schnelleren und besseren Formatierung führt, wird die Technik immer öfters eingesetzt und die Google Crawler damit ausgetrickst. Von hoher Effektivität kann man nicht behaupten, zu mindestens solange nicht, wenn Formatierungen professionell mit CSS-Layouts unternommen werden. Auf statischen HTML-Seiten ist Keyword-Stuffing schnell zu durchschauen. Zudem wirken die Texte unleserlich und die Seite unstrukturiert. Verwendet man jedoch CSS-Angaben, so kann es unter Umständen länger dauern, solche Techniken zu entdecken.

3.2 Doorway-Pages

3.2.1 Grundlegende Idee

Unter Doorway-Pages(auch Brückenseiten[31]) versteht man für spezifische Keywords erstellte Webseiten, die so stark optimiert sind und nicht für Besucher gedacht sind, um für die eigene Webseite eine hohe Platzierung zu erreichen. Sie werden teilweise in Massen erstellt, um am Ende auf die eigene Seite zu verweisen.

3.2.2 Technische Details

Die Doorway-Page wird stark optimiert, vor allem im Hinblick auf die Verwendung von Keywords, sodass der Suchmaschine vorgetäuscht wird, dass es sich um hochqualitativen Inhalt handelt. Der Inhalt einer Doorway-Page ist meistens völlig sinnlos und besteht nur aus Wiederholungen der Keywords. Ziel soll nicht mal sein, Interessenten für die Doorway-Page zu finden, sondern schlichtweg eine hohe Platzierung zu erreichen. Nachdem eine hohe Platzierung erreicht wurde, wird anschließend von diesen Seiten aus ein Link auf die eigentliche Webseite des Webseiten-Betreibers gesetzt, um so eine Steigerung des PageRank-Wertes und der Linkpopularität zu erzielen (wenn der Betreiber sogar unterschiedliche Domains auf unterschiedlichen Servern verwendet auch IP-Popularität).

Üblicherweise enthält die Doorway-Page auch nur einen Link auf die Webseite, aber kein Link von der Webseite führt zur Doorway-Page, sodass die Doorway-Page logischerweise nur von einer Suchmaschine gefunden werden kann. Nun gibt es die Möglichkeit, dass ein Besucher, der eine Doorway-Page anklickt und auf die eigentliche Internetseite verlinkt wird, die Doorway-Page erst gar nicht bemerkt. Mit einem bestimmten Meta-Tag kann das auch realisiert werden:

```
<meta http-equiv=“refresh“ content=“0“;
URL=“decision.htm“>
```

Der Befehl `http-equiv=“refresh“` bedeutet, dass der Browser nach einer bestimmten Zeit dazu aufgefordert wird, eine neue Seite zu laden. Die Ziffer im Attribut `content` gibt die Zeit in Sekunden an und in `URL` ist die Zieladresse, die URL der eigentlichen Webpräsenz. Bei der Verwendung diese Technik sollte allerdings darauf geachtet werden, dass die Dauer der Weiterleitung nicht zu kurz sein sollte, da das für eine Suchmaschine sofort ein Hinweis auf eine Doorway-Page wäre, was negative Folgen für den Verantwortlichen der Doorway-Page hätte. Um dieser Erkennung zu umgehen, kann man die Verzögerungszeit auf z.B. 10 Sekunden setzen und zudem eine JavaScript-Methode ein:

```
window.location.replace(„datei.htm“)
```

Dabei wird durch das Anklicken des Verweises sofort die Seite datei.htm im gleichen Browserfenster aufgerufen und die URL des alten Verweises wird aus dem Verlauf gelöscht.

Im Folgenden soll beispielhaft am Keyword „Kredit“ gezeigt werden, wie so eine Doorway-Page aufgebaut sein kann:

```
<meta name=“keywords“ content=“Kredit“>
<meta name=”description” content=”Kredit
Kreditwürdigkeit Kreditunwürdigkeit”>
<title>Kredit</titel>
<a href=“indexseite.html“><img src=“mainlogo.gif“
width=”200” height=”100” alt=“ Kredit” title=”
Kredit” border=”0”> Kredit</a>
```

Es wird ein Keyword-Tag erstellt, der das Keyword beschreibt. Im Description-Tag wird massenweise das Keyword wiederholt, zudem sollten im Dokumentkörper wenig bis keine Bilder vorhanden sein, damit die Ladezeiten nicht zu hoch werden. Natürlich kann man sich die Mühe sparen und existierende Doorway-Pages kopieren und nur noch die Inhalte verändern. Im Netz gibt es zudem Doorway-Page-Generatoren, z.B. kann ein Betreiber sehr leicht unter http://www.ghpt.de/index.html?http://www.ghpt.de/doorway/index2.html unter Angabe bestimmter Informationen Doorway-Pages generieren lassen, die nur auf den Server hochgeladen werden müssen.

3.2.3 Zusammenfassende Bewertung

3.2.3.1 Wie verhindert Google den Einsatz?

Suchmaschinen haben heutzutage festgestellt, dass viele Doorway-Pages, die nicht einmal über die eigentlichen Webseiten erreichbar sind, die Qualität der Ergebnislisten verringern. Da Doorway-Pages ähnlich wie Spam nutzlosen Inhalt präsentieren und nur dem Zweck dienen, die Platzierung einer Seite zu erhöhen, werden Doorway-Pages als Spam-Techniken des Black-Hat SEO eingestuft. Gängige Suchmaschinen und Google haben heutzutage zudem einen Weg gewählt, Doorway-Pages zu entdecken, indem sie nur höchstens zwei Seiten einer Domain in den Suchergebnislisten anzeigen, sodass es nicht dazu kommen kann, dass eine Seite, mit 100 Doorway-Pages die ersten 100 Plätze in der Trefferliste einnehmen kann. Ein weiterer Hinweis, dass unter anderem Google in der Lage ist, Doorway-Pages zu erkennen, ist ein bekannter Fall, der im Februar 2006 stattgefunden hat. Das Unternehmen BMW setzte Doorway-Pages ein und schaffte es, stetig dieselben Plätze zu belegen und das über einen langen Zeitraum, Die Doorway-Pages waren vor allem auf die Keywords „Gebrauchtwagen“ oder „Jahreswagen“ optimiert[32]. Teile der Seite von BWM wurden sofort aus dem Index gelöscht und zusätzlich dazu wurde der PageRank-Wert insgesamt von Google manuell verringert. Nach Absprachen sollen zwar die Seiten wieder aufgenommen worden sein, aber der Fall zeigt, dass Google durchaus in der Lage ist, Doorway-Pages zu erkennen.

3.2.3.2 Popularität und Effektivität

Das Erstellen und Pflegen von Doorway-Pages, die nicht so schnell aus dem Index geworfen werden sollen bzw. gar nicht sollten, erfordern teilweise einen höheren Arbeitsaufwand. Doorway-Pages eignen sich bei Webseiten, bei denen nicht ständig das Layout geändert werden soll (z.B. Firmenseiten), allerdings gibt es auch hier einige Tricks, den Googlebot auf sich aufmerksam zu machen, ohne dabei auf Black-Hat Techniken zurückgreifen zu müssen(siehe Kapitel …). Von daher sind Doorway-Pages eher lastig, als dass sie wirklich nützen, denn um höhere Plätze anzustreben, ist es meistens nötig, um die 1000 Doorway-Pages zu erstellen. Die Arbeit (auch Generieren braucht seine Zeit) würde sich dann für eine minimale Optimierung und für die dabei aufkommen Zeit- und Geldkosten nicht lohnen. Zudem vermeiden Suchmaschinen mittlerweile generell Doorway-Pages, weil sie die Qualität der Trefferlisten beeinträchtigen.

Suchmaschinen können Doorway-Pages schnell erkennen, sodass es sich kaum noch lohnt, diese zu benutzen. Zudem setzen die Algorithmen von Google zur Gewichtung einer Seite mittlerweile Wert auf die Linkpopularität einer Seite und da Doorway-Pages schlecht verlinkt sind, ist der weiter“vererbte“ PageRank sehr gering. Eine Doorway-Page kann aber in einigen Fällen auch sinnvoll sein und sogar von Suchmaschinen – eventuell unter Absprache – zugelassen werden. Wenn eine Seite zum Beispiel stark auf JavaScript oder Flash basiert und dadurch der Googlebot kaum bis keine Informationen herauslesen kann, können Doorway-Pages, die nicht übertrieben sinnlos gestaltet sind, benutzt werden, um trotzdem noch einen Besucherstrom über Google zu erhalten. Aber grundsätzlich sind heutzutage die Erfolgschancen extrem gering, sodass es nicht wert ist, das Risiko zu begehen und aus dem Index ausgeschlossen zu werden, nur um Doorway-Pages zu benutzen, die sowieso kaum positive Auswirkungen haben.

3.3 IP-Cloaking

3.3.1 Grundlegende Idee

Beim *Cloaking* übermittelt der Webserver einem Crawler andere Daten als sie eigentlich für die Besucher festgelegt sind. Ob es sich um einen Besucher oder um Crawler handelt, kann anhand der Kennungsdaten des Aufrufenden der Webseite erkannt werden.

Beim *IP-Delivering* wird anhand der IP-Adresse der Anfragetyp(Crawler oder normaler Besucher) ermittelt und spezifischer Inhalt an verschiedene IP-Adressen versendet. IP-Delivering kann für seriöse Zwecke benutzt werden, wenn z.B. ein Webseiten-Betreiber eine Seite mit vielen Flash-Animationen führt, die vom Googlebot nicht gelesen werden kann. Allerdings wird diese Methode auch für unseriöse Zwecke benutzt. In diesem Fall spricht man von „IP-Cloaking“. Da IP-Cloaking oft zu unwichtigen Suchergebnissen führen kann, wird sie von Google und anderen Suchmaschinen abgelehnt und bei Anwendung dementsprechend bestraft. Die Motivation beim IP-Cloaking liegt darin, auf bestimmte Keywords stark optimierte Seiten zu erstellen und über diese Besucher auf die eigene Seite zu locken, dessen Inhalt jedoch wenig mit den Keywords zu tun hat. Ein weiterer Anlass für das Verwenden dieser Methode ist die Tatsache, dass Seiten, die viele Grafiken besitzen und vor allem z.B. auf Flash oder JavaScript aufbauen, wenig Text besitzen, sodass diese Seiten für die Googlebots nicht.

3.3.2 Technische Details

Ein normaler Besucher bekommt von der Technik nichts mit, da diese im Hintergrund abläuft. Google hat eine Cache-Funktion, d.h. der Besucher kann sich den Cache von Google ansehen und erfährt, wie der Googlebot eine Seite zuletzt gesehen hat. So könnte ein Besucher erfahren, ob der Crawler die gleiche Seite gesehen hat wie der Besucher oder aber eine veränderte Seite. Allerdings funktioniert das nicht immer, da der Webseiten-Betreiber über einen einfachen HTML-Tag die Aufnahme in den Cache verhindern kann:

```
<meta name="Googlebot" conten="noarchive">
```

Der Client sendet an Google eine Anfrage(request) und teilt dabei seine User Agent-Informationen mit. Dabei kann es sich beim Clienten sowohl um einen einfachen Besucher über einen Internetbrowser handeln, aber auch um irgendein Programm oder ein Crawler.

Beim User Agent handelt es sich um ein Client-Programm, der eine Schnittstelle zwischen einem Netzwerk und dem Benutzer darstellt. Internetbrowser oder E-Mail-Programm sind z.B. User Agents. Sie können gezielt Inhalte aus einem Netzwerk darstellen und Befehle vom Benutzer entgegennehmen. Der User Agent enthält in seinem Header Informationen darüber, auf welchem Betriebssystem es läuft, welche Version (bei einem Browser z.B. die Browserversion) und die verwendete Sprache. Allerdings gibt es keine Richtlinie, wie die User Agent-Header aussehen müssen. Außerdem kann die User Agent vom Benutzer selbst verändert werden, z.B. über die Registry des Betriebssystems. Im unteren Beispiel ist die User Agent eines Firefox-Browsers:

```
Mozilla/5.0 (Windows; U; Windows NT 6.0; de;
rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5
```

Neben Informationen wie der Name und dem laufenden Betriebssystem werden unter anderem Angaben zur Sprache und zu Sicherheitseinstellungen (U, N oder I) sowie zur Versionsnummer gegeben.

Üblicherweise wird Cloaking mithilfe eines Skriptes durchgeführt. Jedesmal, wenn ein User Agent Daten von einem Webserver aufrufen möchte, wird die CGI-Umgebungsvariable HTTP_USER_AGENT initialisiert[33]. Wenn ein Webserver CGI (ein Standard für den Datenaustausch zwischen Webservern und Clienten) unterstützt, so enthält es unter anderem Umgebungsvariablen, die dem Webserver über die Anfragen und über den Status des Webservers informieren. Nachdem die Umgebungsvariable nun gesetzt wurde, enthält es die Informationen des Aufrufenden, z.B. den Namen des User Agents. Das Skript ermittelt nun anhand der Informationen in der Variable, ob es sich um einen Besucher oder um die User Agent eines Crawlers handelt. Heißt der User Agent z.B. „Googlebot“ statt „Mozilla/4.0“ bzw. besitzt es Informationen, die einem Bot zugeordnet werden können, so wird ein anderer stark optimierter Inhalt gesendet als einem Besucher.

Allerdings ist es leicht, den User Agent Namen zu verändern. Von daher wird das Verfahren um die IP des Anfragenden erweitert. Neben der Überprüfung der User Agent Informationen wird die IP-Adresse des Anfragenden verglichen mit IP-Adressen aus Datenbanken, die im Netz veröffentlicht werden. Profis gehen einen Schritt weiter und machen sogenannte *Tracerroutes[34]*. Dabei wird überprüft, über welchen Server im Netz die Anfrage weitergeleitet wird. Noch detailreichere Informationen über die anfragende IP-Adresse liefert die Seite http://www.ripe.net/, bei der teilweise sogar Telefonnummern von Personen gefunden werden können.

3.3.3 Zusammenfassende Bewertung

3.3.3.1 Wie verhindert Google den Einsatz?

Da IP-Cloaking als eine Spam-Methode eingestuft wird, wird sie dementsprechend bei Verwendung von Google bestraft durch einen manchmal dauerhaften Rauswurf aus dem Index oder mit starker Senkung der Platzierung.

Hat Google bei einer Seite Verdacht auf IP-Cloaking, wird ein weiterer Googlebot versendet, der in seinem User Agent-Header als normaler Benutzer getarnt ist. Dieser ruft die verdächtige Seite erneut auf und bei vorliegender Verwendung wird dem Bot eine andere Seite als die vorher besuchte Seite angezeigt, sodass Googlebot Seiten nur noch miteinander vergleichen muss.

Mittlerweile werden kompliziertere Methoden eingesetzt, um IP-Cloaking aufzudecken. Dabei werden sogenannte Referrer eingesetzt, von der aus ein Besucher durch Verfolgung eines Hyperlinks zur aktuellen Seite gelangt ist. Beim Aufruf einer Seite von einem Webserver wird der Referrer ebenfalls an den Webserver gesendet und kann unter den Server-Logfiles eingesehen werden. Suchmaschinen benutzen immer öfter die Idee von Referrern, um Cloaking bzw. IP-Cloaking aufzudecken. Dabei wird eine Seite zuerst vom Googlebot aufgerufen und kurze Zeit später wird eine Anfrage mithilfe von veränderten Referren gemacht, damit es scheint, als ob ein Besucher die Seite über Google gefunden hat bei Eingabe von Suchwörtern wie „abc“ (siehe folgendes Beispiel[35]).

```
Googlebot:
crawl-66-249-66-243.googlebot.com - -
[07/Feb/2008:13:10:35 +0100] "GET /news/234-neue-
msn-suche-online.html HTTP/1.1" 200 8223 "-"
"Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
Suchmaschine, getarnt als Besucher
74.125.16.67 - - [07/Feb/2008:13:34:52 +0100]
"GET /news/234-neue-msn-suche-online.html
HTTP/1.1" 200 8223
"http://www.google.com/search?q=abc" "Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.7)
Gecko/20060909 Firefox/1.5.0.7"
```

So erscheint es, als ob ein normaler Besucher die Seite über Google gefunden hat, sodass der optimierte Inhalt angezeigt wird. Bei Entdeckung von Cloaking bzw. IP-Cloaking kann Google die betreffende Seite evtl. sofort aus dem Index löschen und nicht selten ist dieser Rauswurf unbegrenzt.

3.3.3.2 Popularität und Effektivität

Google rät grundsätzlich, IP Cloaking zu vermeiden und stattdessen mithilfe von speziellen Description-Tags im HTML-Dokument die nicht anzeigbaren Elemente zu beschreiben. Wie effektiv IP-Cloaking ist, hängt unter anderem von den Kenntnissen des Webseiten-Entwicklers ab. Oft scheitern Black-Hat SEO’s schon an der Identifizierung des Anfragenden, indem nur mithilfe der User Agent gearbeitet wird. Um nun noch mithilfe der IP den anfragenden zu bestimmen, müssen diese ständig in Datenbanken ausgelesen werden. Da Google ihre IP-Adressen immer öfter ändert und mittlerweile auf Referrerbasis arbeitet, wird es umso schwerer, die Seite auf einen Besuch des Googlebots vorzubereiten. Ebenso wie bei Doorway-Pages erfordert das ständige Pflegen der optimierten Seite einen hohen Arbeitsaufwand, sodass es fraglich ist, ob sich der Aufwand lohnen wird. Das hohe Risiko stellt den größten Nachteil von IP- Cloaking dar. Da sich viele Entwickler durch die Fülle an Konkurrenten bedroht fühlen, existiert IP- Cloaking heutzutage wenn auch vermindert immer noch.

3.4 Bait-and-Switch

3.4.1 Grundlegende Idee

Beim Bait-and-Switch handelt es sich um eine rudimentäre Form von Cloaking. Eine für Google optimierte HTML-Seite wird bei Google angemeldet. Nachdem diese im Index

vorhanden ist, wird die Seite mit einer Seite ausgetauscht, die für menschliche Besucher stark optimiert ist.

3.4.2 Technische Details

Für diese Methode sind zwei Schritte notwendig. Zu Beginn wird eine Webseite erstellt, die stark für Google optimiert ist und darauf wird diese manuell angemeldet. Nach der Indexierung wird im zweiten Schritt unter der gleichen URL der Seite allerdings die Webseite gegen eine neue Seite ausgetauscht, die z.B. Flash enthält, die sonst vom Bot nicht lesbar. Die ursprüngliche Seite hatte dank ihrer starken Optimierung eine hohe Platzierung bekommen und durch den Austauscht „erbt“[36] die neue Seite diese Platzierung. Solange bis erneut ein Googlebot die Seite besucht, wird unter der URL die neueingefügte Seite zu finden sein. Nach dem erneuten Besuch stellt der Googlebot fest, dass die Seite aktualisiert wurde, und der Index wird mit den neuen Daten ersetzt bzw. die Platzierung wieder angepasst, sodass es dann zu einem starken Fall des Rankings kommen kann.

3.4.3 Zusammenfassende Bewertung

3.4.3.1 Wie verhindert Google den Einsatz?

Im Gegensatz zu den bisher genannten Black-Hat-Techniken kann diese Technik von Google nicht erkannt werden, denn Google kann nicht sehen, ob es sich um einen Manipulationsversuch oder wirklich um eine Aktualisierung einer Webseite handelt. Deshalb kann diese Methode auch nicht bestraft werden. Allerdings kann es zum Ausschluss aus dem Index kommen, wenn zum Zweck der höheren Platzierung die Kopie einer anderen, gut optimierten Seite als eigene Seite benutzt wird. Wenn dann im zweiten Schritt mit der eigentlichen Seite sogar durch Werbeanzeigen Geld verdient wird, so kann dies bestraft werden, falls jemand die Verwendung der Methode erkennt.

3.4.3.2 Popularität und Effektivität

Diese Spam-Methode war früher sehr beliebt, ist aber mittlerweile nutzlos. Sobald der Googlebot die Seite erneut besucht, wird ein neuer PageRank ermittelt und die Seite wird wieder nach unten platziert. Der Einsatz hat erst dann Sinn, wenn eine Seite für eine sehr kurze Dauer hochplatziert werden soll. Allerdings ist fraglich, ob der Aufwand dafür gerecht ist. Die optimierte Seite muss angemeldet und zuerst indiziert werden, was unter Umständen auch mehrere Wochen dauern kann. Dann muss die Seite ausgetauscht werden mit der eigentlichen Seite. Zudem werden die Goolgebots mittlerweile immer häufiger versendet[37], sodass das Austauschen der Seiten schnell bemerkt wird. Stattdessen wird heute lieber IP-Cloaking verwendet, welche zwar von Suchmaschinen immer schneller enttarnt werden kann, aber trotzdem bei guter Identifikation der Anfragenden immernoch einen längeren Erfolg versprechen kann als Bait-and-Switch.

3.5 Logfile-Spam

3.5.1 Grundlegende Idee

Beim Logfile-Spam besuchen Webseiten-Entwickler über ihre eigene Domain andere Seiten im WWW, um ihre Informationen in den Logfiles der Server zu hinterlassen, so steigert sich ihr Bekanntheitsgrad, da ihre Zugriffsinformationen in den Statistiken der Webseiten auftauchen. Wenn diese dazu öffentlich sind, ist das für die Lofgilespammer gute Werbung und zudem ein guter Backlink[38].

3.5.2 Technische Details

Logfiles sind Dateien, die automatisch erstellt werden und bestimmte Aktionen von Prozessen auf einem laufenden Betriebssystem erstellen. Auch ein Webserver erstellt automatisch eine Logfile-Datei. Diese enthält die IP-Adresse des Clienten, ggf. die Identitätsinformationen des Clientrechners und Benutzers, die Aufrufszeit, die Antwort des Servers (im Fall von 200 entspricht es einer erfolgreichen Anfrage) und die Größe der übertragenen Daten in Bytes (im unteren Beispiel[39] sind es 512 Byte).

```
183.121.143.32 - - [18/Mar/2003:08:04:22 +0200]
"GET /images/logo.jpg HTTP/1.1" 200 512
"http://www.wikipedia.org/" Mozilla/5.0 (X11; U;
Linux i686; de-DE;rv:1.7.5)
183.121.143.32 - - [18/Mar/2003:08:05:03 +0200]
"GET /images/bild.png HTTP/1.1" 200 805
"http://www.google.org/"
```

Wie zu sehen ist, wird auch der Link im Logfile gespeichert, von der aus auf die Seite zugegriffe wurde. Ein Logfilespammer nutzt gerade diese Tatsache aus, ruft die Seite über seine Domain auf, sodass dann die URL der Domain in den Logfiles steht. Obwohl Logfiles einige Probleme mit sich bringen, wie z.B. die manchmal enorme Größe der Dateien, veröffentlichen viele Webseiten-Entwickler die Statistiken ihrer Logfiles für Suchmaschinen. Die Gründe für die frei zugänglichen Logfile-Analysen sind unter anderem das Angeben, wie viele Seiten aufgerufen wurden, wie viele Besucher die Seite aufgerufen haben undnvor allem interessant für Google-Suchmaschinenoptimierer, von wo aus die Besucher kommen, denn die Herkunftdomain wird als ein echter Link angegeben. Dadurch werden diese Links nach gewisser Zeit vom Googlebot gefunden und durchsucht, sodass quasi ein weiterer Backlink auf die Seite der Logfilespammer existiert und sich damit das Ranking ihrer Seite verbessert. So erzeugt sich der Logfilespammer auf eine schnelle und einfache Art und Weise kostenlos einen Backlink. Zudem können Programme eingesetzt werden, die vom Logfilespammer eine Liste an Domains erhalten, in deren Statistiken ein Link gesetzt werden soll. Dieses Programm ruft die Seiten automatisch auf, sodass dann in all den Logfiles die Referrerinformationen der zu optimierenden Seite angegeben werden, der dann als Link markiert wird[40].

Der Schaden, der dabei angerichtet werden kann, ist ebenfalls groß, denn zum einen werden die Logfilestatistiken eines Webservers durch die Einträge einer Seite in die Logfiles verfälscht und zum anderen wird diese Spam-Variante nicht einmal pro Webserver, sondern mehrmals durchgeführt.

3.5.3 Zusammenfassende Bewertung

3.5.3.1 Wie verhindert Google den Einsatz?

Google kann Logfile-Spam nicht direkt erkennen, von daher bietet Google den Webseiten-Entwicklern die Möglichkeit, ein rel=“nofollow“-Attribut zu setzen, mit der die Verweise „markiert“ werden sollen, die nicht in die Berechnung des PageRank-Wertes einfließen sollen[41]. Allerdings hat diese

Methode einen Nachteil, denn sobald dieses Attribut automatisch gesetzt wird, sind auch andere Links davon betroffen, die der Betreiber jedoch in die Gewichtung mitberücksichtigen möchte.

Es gibt von daher eine zweite Möglichkeit, und zwar in die .htaccess-Datei alle typischen Keywords, die für Spam gebräuchlich sind, einzutragen und sie so zu konfigurieren, dass sie, sobald einer dieser Keywords im Referrer der Spammer auftaucht, den Status 403(Zugriff verboten) zurücksenden. Natürlich entsteht hier ein Problem, denn der Betreiber müsste selbst solch eine Liste erstellen und die stets aktualisieren. Im Folgenden soll ein Beispiel dargestellt werden, wie solch ein Eintrag aussehen mag:

```
RewriteEngine On
RewriteCond %{HTTP_REFERER}
^http://(www\.)?.*adult(-|.).*$ [OR]
RewriteRule ^(.*)$ %1 [R=301,L]
```

Nachdem die RewriteEngine, ein Modul zur Umleitung von Links, eingeschaltet wird, folgten Deklarationen von Bedingungen. Am Ende folgt der eigentliche Befehl, der durch „RewriteRule“ eingeführt wird. Wird eine der Bedingungen erfüllt, so wird die Anweisung in der `RewriteRule` ausgeführt. Allerdings ist es fraglich, ob diese Methode gut schützt, denn für die Logfilespammer ist es nicht schwer, sich neue Domains anzulegen.

3.5.3.2 Popularität und Effektivität

Logfile-Spamming ist eine effektive Methode. Trotz der Gegenmaßnahmen durch Google und durch den Webseiten-Entwicklers, wird diese Methode sehr häufig verwendet. Zumal die grundlegende Idee von Suchmaschinen selbst eingesetzt wird, um IP-Cloaking zu bekämpfen, ist Logfiel-Spam eine neben ihrer Effizienz auch populäre Methode. In zahlreichen Weblogs finden sich Beschwerden über Logfile-Spamming[42], die die Logfiles unlesbar machen und die Statistiken verfälschen. Wie in Kapitel 3.5.3.1 geschildert, sind die Maßnahmen gegen Logfile-Spam zwar effizient, aber nicht auf Dauer. Zudem bergen diese auch Nachteile für den Webseiten-Betreiber, der sich gegen Logfile-Spam wehren möchte. So kann er z.B. durch die Verwendung des `rel="nofollow"`-Attributs ganze Seiten blockieren.

3.6 Content Spam

3.6.1 Grundlegende Idee

Die zugrundeliegende Idee dieser Technik ist, den Inhalt von Webseiten zu stehlen, um nicht selbst den hohen Arbeitsaufwand zu leisten, der mit dem Editieren von qualitativem Inhalt verbunden ist. Diese sogenannte Dubletten[43] sind dann über anderen URLs abrufbar. Da die kopierten Inhalte dann woanders schon einmal auffindbar sind, verschlechtern sie die Suchergebnisliste, falls qualitativer Text kopiert wurde.

3.6.2 Technische Details

Um in den Rankings bei Google gut abzuschneiden, ist es wichtig neben guten Optimierungsstrategien auch guten Inhalt zu präsentieren. Das Schreiben von guten Inhalten ist jedoch arbeitsaufwändig, sodass das Stehlen von guten Inhalten anderer Betreiber diese erspart. Unter anderen wird diese Technik gerade dann angewendet, wenn mithilfe von Google AdSense Umsatz gemacht werden soll, einem Dienst von Google, bei der Google-Anzeigen auf der eigenen Seite platziert werden, Werden diese angegklickt, so bekommt Google daraus Einnahmen, von denen ein Teil an den Betreiber der Seite geht. Allerdings ist diese Variante des Conten-Spam mittlerweile erschwert worden, da Google in der Lage ist, kopierten Inhalt, auch Duplicate Content genannt, zu erkennen[44]. Von daher suchen Content-Spammer immer öfter Texte aus Foren und Gästebüchern, aber auch in öffentlichen Webkatalogen. Oft ist den Spammern der Inhalt egal, Hauptsache sie bekommen Texte. Mittlerweile werden dafür Programme, die als Content-Grabber[45] bezeichnet werden, eingesetzt. Diese besuchen automatisch mehrere Webseiten und lesen ihre Inhalte aus.

3.6.3 Zusammenfassende Bewertung

3.6.3.1 Wie verhindert Google den Einsatz?

Eine wirklich gute Maßnahme gegen diese Spam-Methode ist schwer, oft kann der Webseiten-Betreiber selbst etwas gegen diese Variante des Spams unternehmen. Mithilfe von Gleichungen, die zwei Seiten auf Duplicate Conten überprüfen, kann Google zumindestens etwas dagegen tun. Zwei Seiten werden z.B. mithilfe von Filtern verglichen, indem der prozentuale Anteil bestimmter Auszüge in einem Artikel verglichen wird mit dem im anderen Artikel. Ist dieser Anteil zu hoch, besteht Verdacht auf Content-Spam.

Ein Webseiten-Betreiber kann auch selbst etwas gegen Content-Spam tun, auch wenn diese Strategie auf lange Sicht nicht effektiv ist, weil sie zu zeitaufwendig ist. Dabei durchsucht der Betreiber die Logfiles auf seinem Webserver und sperrt die gefundenen Bots, die Inhalte klauen, permanent mithilfe der RewriteEngine in der .htaccess-Datei.

Um sich diese Arbeit etwas zu erleichtern, wird Betreibern ein kostenloses PHP-Skript auf www.bot-trap.de angeboten, welches in die Seite integriert werden kann und wie ein aktiver Virenscanner einen Spamschutz bietet und sich selbstständig mit neuen Spammerkmalen updatet.

3.6.3.2 Popularität und Effektivität

Content-Spam ist eine sehr beliebte, häufige, aber nervende Methode. Sie ist deshalb so effektiv, weil sich der Betreiber enorm viel Zeit beim Editieren der Texte spart bzw. noch effektiver ist, wenn er das Auffinden von guten Inhalten durch Skripte automatisiert. Wenn dazu die Texte, die auf bestimmte Keywords optimiert sind, nicht sinnlos und zusammenhangslos sind, sondern die gestohlenen Auszüge trotzdem irgendeinen Sinn ergeben, so kann der Betreiber, solange er nicht entdeckt wird, mit guten Platzierungen rechnen. Zudem ist diese Methode sehr populär. In zahlreichen Blogs beschweren sich Webseiten-Entiwckler über ihre überfluteten Logfiles ihrer Webserver durch die Spammer. Wie aktuell das Thema tatsächlich ist, zeigt der Weblog unter http://blog.bernd-distler.net/2008/11/10/content-spam, wo der Autor diese Art von Spam feststellte. Ein möglicher Grund für die hohe Aktualität dieser Methode ist das Problem, Spam wirklich sicher bekämpfen zu können. Wenn Google eine Seite findet, kann es von der Seite nicht wirklich behaupten, ob sie wirklich Content-Spam enthält oder ob der Autor sich nur „Ideen“ geholt.

4. GOOGLES REAKTION AUF BLACK-HAT

Google versucht permanent, die Qualität ihrer Suchergebnislisten zu pflegen. Es gibt viele Funktionen, die es ermöglichen, nicht nur gute Texte auf der Seite zu erstellen, sondern auch das Layout der Seiten schöner zu gestalten. Leider werden einige dieser Funktionen von Google nicht unterstützt bzw. bereiten dem Googlebot Probleme, aus diesen besonders gestalteten Elementen Informationen zu lesen. Leider werden einige dieser Funktionen bei Black-Hat Methoden eingesetzt, sodass es Google erschwert wird, diese zu entdecken und zu ahnden. Das grundlegende Wissen, eine Seite für den Googlebot lesegerecht zu gestalten ist also wichtig.

4.1 Welche Techniken erkennt Google? Wo hat Google Probleme?

Ein großes Problem für den Googlebot ist die Skriptsprache *JavaScript*. Jeder Inhalt, der von JavaScript umschlossen ist, kann vom Googlebot nicht gelesen werden und wird ignoriert[46]. Falls Webseiten-Entwickler trotzdem JavaScript verwenden wollen, so sollten sie über ihre Sitemaps einen normalen Link zum Inhalt setzen, der sich im JavaScript-Bereich befindet. Im JavaScript-Bereich kann zusätzlich ein <noscript>-Tag benutzt werden, sodass Verweise, die sich in diesem Bereich befinden, nochmal als normalen Link auftauchen.

Ein weiteres Problem für den Googlebot sind Content Management Systeme (CMS), Foren und dynamische Webseiten. Viele Webseiten-Entwickler arbeiten oft mit diesen Funktionen, testen allerdings nicht, wie sich diese Seiten gegenüber Google zeigen, da nicht immer alle technischen Anforderungen aufeinander abgestimmt sind. Ansonsten führt das dazu, dass einige Seiten nicht gecrawlt oder gar nicht indiziert werden. Vor allem dynamische URLs stellen ein großes Problem für Google dar, da sie oft unzählige Parameter beinhalten(z.B. `id=` oder `sid=`). Generell problematisch sind dabei dynamische Seiten, die auf Cookies oder Session-IDs beruhen, um Besucher besser identifizieren zu können.

Ein noch viel größeres Problem als dynamische URLs sind die *.htaccess-Dateien* von Webservern. Diese sind für viele Black-Hat-Methoden eine wichtige Stelle, um sich gegen diese zu wehren. Allerdings kann großer Schaden angerichtet werden, wenn diese Datei fehlerhaft geschrieben ist. Mithilfe der RewriteEngine kann man über diese Datei seine Seite vor Content-Spam, vor IP-Cloaking und anderen Black-Hat Techniken schützen. Allerdings passiert es nicht selten, dass nur darauf geachtet wird, dass die Seite am Ende fehlerfrei läuft anstatt wirklich zu überprüfen, ob die Anweisungen in der .htaccess-Datei ausgeführt werden[47]. Der Leichtsinn mancher Betreiber kann für Google ein großes Problem sein. So kann es passieren, dass die Seite zwar funktioniert, aber aufgrund fehlerhafter .htaccess-Dateien im Hintergrund Content-Spam oder untypische IPs zugelassen werden. Auf korrekte Dateien sollte demnach geachtet werden, sowohl von ihrer Lauffähigkeit als auch in ihren Anweisungen, denn wenn Schaden angerichtet wird, ist dieser meistens groß und die Fehlerquelle schwer auffindbar.

Ein weiteres Problem für Google stellt *Flash* dar[48]. Flash ist eine sehr beliebte Funktion, Seiten anspruchsvoll und für den Menschen optimiert zu gestalten. Allerdings ist jede Art von Flash, Mutlimedia, Java-Applets, Videos und Musik für Googlebots nicht lesbar, da die Inhalte dieser Funktionen auf dem Rechner des Clienten generiert werden und dort gelesen werden müssten. So kommt es schnell zum Missbrauch der Funktionen in Form von Doorway-Pages oder IP-Cloaking. Da Betreiber trotz der Verwendung von Flash eine hohe Platzierung erreichen wollen, erstellen sie optimierte Brückenseiten, die Doorway-Pages (s. Kapitel 3.2) oder Seiten, die extra für den Googlebot optimiert wurden(IP-Cloaking, s. Kapitel 3.3). Mithilfe bestimmter HTML-Tags können zwar Inhalte der Flash-Bereiche zusammengefasst werden, allerdings reicht es einer Suchmaschine meistens nicht aus, anhand dieser reduzierten Informationen eine sinnvolle Platzierung zu ermitteln.

Eine weitere Methode, um Suchergebnisse zu manipulieren oder gezielt Schaden anzurichten, ist das *Cross Site Scripting[49]* (auch XXS genannt).

Bei dynamischen Webseiten, die meistens über PHP-Skripte laufen, werden die Inhalte dynamisch gestaltet, also bei jedem Aufruf neu generiert. Die Anfragen der Besucher werden anhand der Parameter in der URL gelesen:

```
get    http://www.irgendeinedomain.de/script.php&page-id=0815
```

Solche Anfragen werden z.B. beim Ausfüllen von Formularen an den Webserver gesendet. Wird allerdings statt der korrekten page-id ein Scriptcode versendet, der dann ungeprüft auf dem Webserver bzw. gelagert wird oder, falls die Daten an der Browser des Besuchers zurückgesendet werden, im Browser des Benutzers gelesen wird, so kann Schadcode ausgeführt werden. Die Person, die diesen Schadcode nun auf den Server des Betroffenen übertragen hat, kann das nun ausnutzen, z.B. Doorway-Pages zwischenschalten, IP-Cloaking durchführen oder im schlimmsten Fall dafür sorgen, dass die Seite durch Einsatz von Black-Hat-Techniken, die das Opfer nicht einmal selbst angewendet hat, aus dem Index gelöscht wird.

4.2 Schutz vor Black Hat SEO

4.2.1 Wie schütze ich mich vor versehentlichem Einsatz von Black Hat-Techniken?

Um sich wirklich effizient vor versehentlichem Einsatz von Black Hat-Techniken zu schützen, ist es zuerst wichtig, diese zu kennen. Oft ist Betreibern nicht klar, dass z.B. zu oft verwendete Keywords zum Keyword-Stuffing tendiert. Mithilfe eines Tools kann man z.B. die Keyworddichte einer Seite überprüfen lassen, um festzustellen, wie hoch diese auf bestimmte Keywords ist. Einer dieser Tools ist unter http://www.ranks.nl/tools/spider.html zu finden.

Zudem gibt es weitere nützliche SEO-Tools, wie z.B. das SEO-Tool IBP von www.axandra.de. Es bekommt gute Kritik und kann sowohl automatisch als auch halbautomatisch Positionen von Keywords analysieren, Link-Optimierungen vornehmen oder Suchmaschinen emulieren, die die eigene Seite betrachten und dem Betreiber darstellen, wie die Seite vom Googlebot gesehen wird.

Um nicht versehentlich Doorway-Pages einzusetzen, ist es zum einen wichtig, sich gegen XXS zu schützen, auf das im

Folgenden noch eingegangen wird, denn ansonsten kann ein Betreiber nicht „aus Versehen“ Doorway-Pages erstellen, es sei denn, er will es. Falls jedoch ein Verzicht auf Doorway-Pages unmöglich ist, so sollten diese nicht automatisch generiert werden lassen, sondern wirklich gut gestaltet werden und ihr Inhalt sollte dem Inhalt der eigentlichen Seite passen, da sie sonst den Verdacht auf Spam wecken. Das gleiche gilt auch für IP-Cloaking und Cloaking.
Auf http://www.seochat.com/ werden viele verschiedene Tools angeboten, die z.B. die Linkpopularität optimieren oder zwei Seiten miteinander auf Content-Spam vergleichen können.

Neben den Tools genannten Tools ist es vor allem wichtig, sich gegen XSS zu schützen, denn falls ein Betreiber ersteinmal Opfer einer XSS-Attacke ist, so ist der Schaden, der dadurch angerichtet werden kann, enorm. Von daher ist es wichtig, sich vor XSS zu schützen. Der einfachste Schutz vor XSS ist, schon zu Beginn nur statische Seiten zu erstellen oder Scripts nur in Bereichen ausführen zu lassen, die passwortgeschützt sind[50]. Eine weitere Methode ist es, konsequent die Skripte zu patchen, also sich darüber zu informieren, wo die neusten Sicherheitslücken liegen, und sein Skript daran anzupassen. Dazu gibt es auch Dienste im WWW, die Skripte auf Sicherheitslücken überprüfen können, wie z.B. Chorizo (https://chorizo-scanner.com/).

Eine dritte, interessante und hilfreiche Methode ist, wie vorher erwähnt, ein Suchmaschinenemulator. So kann getestet werden, was die Suchmaschine bei einer Anfrage bekommt. Solch ein Tool ist z.B. das „Utility Header Test“, welches auf der Seite http://www.shamrock.de/tools.htm erhältlich ist. So können die Daten, die das Tool liefert mit den Daten verglichen werden, die der Browser anzeigt. Sind diese unterschiedlich, könnte der Verdacht auf XSS bestehen.

Zudem ist es wichtig, vor allem Parameter, die an den Webserver mittels der URL gesendet werden, vorher zu überprüfen und nur die zuzulassen, die auch erwünscht sind. Z.B. können so Seiten so eingestellt werden, dass sie JavaScript-Code nicht annehmen.

4.2.2 Wie schütze ich mich davor, Opfer zu werden

Das Schlimmste, was einem Betreiber passieren kann, ist Opfer von XSS zu werden. Mögliche Schutzstrategien wurden im vorigen Kapitel erläutert. In diesem Kapitel soll noch einmal kurz erläutern werden, wie sich ein Betreiber davor schützen kann, Opfer einer Black-Hat-Methode zu werden. Ein guter Schutz vor Logfile-Spam, Content-Spam und anderen Spam-Methoden ist der Einsatz der .htaccess-Datei. Mithilfe dieser Datei können alle Zugriffsdirektiven eines Webservers geregelt werden. Nicht nur das Blockieren von Unterverzeichnissen eines Webservers kann damit erfolgen, auch das Schützen mit einem Passwort gelingt mithilfe dieser Datei. Das unbefristete Aussperren von IPs ist ein wichtiger Grund zur häufigen Verwendung der .htaccess-Datei. Um z.B. Logfile-Spam auszuschließen, können mithilfe dieser Datei Bedingungen gesetzt werden (s. Kapitel 3.5.3.1). Falls diese erfüllt werden (z.B. auftauchen bestimmter Spamkeywords im Referrer), so werden Aktionen ausgelöst. Diese reichen von Umleitungen auf andere Seiten bis hin zur Blockierung oder Umleitung auf die Spamseite zurück. Ein Beispiel wären die Einstellungen für zugreifende IPs oder Zugriffsformen. Für einige http-Anfragen, wie GET, PUT oder POST können unterschiedliche Berechtigungen verteilt werden, was beim Aufruf geschehen soll, z.B. wer alles Formulare abschicken kann oder dass bei GET eine Seite nur angeschaut werden darf:

```
<Limit GET>
  # Leseberechtigungen
</Limit>
<Limit POST PUT>
  # Schreibberechtigungen in Formularen
</Limit>
```

Eine weitere Variante, sich effektiv vor Black Hat-SEOs zu schützen, kann das Erstellen der Webseite mit Google Sitemaps sein. Google Sitemaps liefert auf Anfrage Statistiken auf Spam-Verdacht der Seite, beim Rauswurf sogar detaillierte Gründe, warum es dazu kam und weitere Funktionen.

5. ZUSAMMENFASSUNG

Obwohl einige der genannten Black Hat-Techniken sehr interessant klingen und bei professionellem Einsatz sogar zu guten Platzierungen führen können, funktionieren diese Techniken nicht unbefristet, da sich die Suchmaschinen ständig anpassen, um ihre Suchergebnislisten von Spammern rein zu halten. Die Konsequenzen für den Einsatz dieser Techniken sind meistens enorm hoch, manchmal kann es zum dauerhaften Bann aus dem Index kommen. Das Wissen über Black Hat SEO ist trotzdem enorm wichtig, um sich im Netz vor teils böswilligen Spammern zu schützen oder nicht selbst aus Versehen diese Techniken zu verwenden und sich dann hinterher zu wundern, warum die Seite nicht mehr im Googleindex zu finden ist. Am Ende bleibt es jedoch jedem überlassen, auf welche Art und Weise der Betreiber seine Platzierungen erhalten möchte.

6. QUELLEN

[1] http://www.bitkom.org/46074_46069.aspx, Zugriff am 7.10.2008 um 21:59Uhr
[2] J. Winkler, *Suchmaschinenoptimierung*, Franzis Verlag GmbH: Poing, 2007 , S. 7
[3] http://www.heise.de/newsticker/Jagd-frei-auf-die-Hommingberger-Gepardenforelle--/meldung/58647, Zugriff am 25.12.2008 um 04:40Uhr
[4] F. Neumeier, *Websites optimieren für Google&Co. schnell + kompakt*, 1. Auflage, 2007, entwickler.press, S. 21
[5] http://www.beyond-media.net/architektur_google.html, Zugriff am 24.12.2008 um 12.44Uhr
[6] http://www.beyond-media.net/architektur_google.html, Zugriff am 24.12.2008 um 12.53Uhr
[7] http://seodoneright.blogspot.com/2007_10_28_archive.html, Zugriff am 26.12.2008 um 14:35Uhr
[8] J. Winkler, *Suchmaschinenoptimierung*, Franzis Verlag GmbH: Poing, 2007 , S. 29
[9] M. Glöggler, *Suchmaschinen im Internet*, 2003, Berlin: Springer-Verlag, S.27
[10] http://www.google.de/corporate/tech.html, Zugriff am 4.1.2009 um 01:30Uhr
[11] J. Winkler, *Suchmaschinenoptimierung*, Franzis Verlag GmbH: Poing, 2007 , S. 22
[12] M. Glöggler, *Suchmaschinen im Internet*, 2003, Berlin: Springer-Verlag, S.83
[13] http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf, Zugriff am 30.12.2008 um 12.24Uhr, S. 4
[14] http://www.aifb.uni-karlsruhe.de/Lehre/Sommer2006/kdtm/stuff/Google-PageRank-V1.0.pdf, Zugriff am 10.1.2009 um 08:230Uhr, S. 20
[15] http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf, Zugriff am 30.12.2008 um 12.33Uhr, S. 5

[16] http://www.aifb.uni-karlsruhe.de/Lehre/Sommer2006/kdtm/stuff/Google-PageRank-V1.0.pdf, Zugriff am 11.1.2009 um 13:08Uhr, S. 38
[17] J. Winkler, *Suchmaschinenoptimierung*, Franzis Verlag GmbH: Poing, 2007 , S. 19
[18] http://www.seo-suchmaschinenoptimierung.at/keyworddichte.html, Zugriff am 12.01.2009 um 14:35Uhr
[19] http://www.beyond-media.net/lage_naehe.html, Zugriff am 12.01.2009 um 15:01Uhr
[20] J. Winkler, *Suchmaschinenoptimierung*, Franzis Verlag GmbH: Poing, 2007 , S. 109
[21] J. Winkler, *Suchmaschinenoptimierung*, Franzis Verlag GmbH: Poing, 2007 , S. 78
[22] http://www.drisol.com/informationen/seo-lexikon/onpage/, Zugriff am 12.02.2009 um 15:23Uhr
[23] S. Erlhofer, *Suchmaschinen-Optimierung für Webentwickler*, Galileo Press GmbH, S. 225
[24] http://www.drisol.com/informationen/seo-lexikon/offpage/, Zugriff am 31.12.2008 um 15:25Uhr
[25] S. Erlhofer, *Suchmaschinen-Optimierung für Webentwickler*, Galileo Press GmbH, S. 246
[26] J. Winkler, *Suchmaschinenoptimierung*, Franzis Verlag GmbH: Poing, 2007 , S. 23
[27] http://www.tecchannel.de/webtechnik/entwicklung/1766517/google_optimierung_die_schmutzigen_tricks/index.html, Zugriff am 1.1.2009 um 17:10Uhr
[28] http://www.tecchannel.de/webtechnik/entwicklung/1766517/google_optimierung_die_schmutzigen_tricks/index3.html, Zugriff am 1.1.2009 um 17:22Uhr
[29] M. Glöggler, *Suchmaschinen im Internet*, Springer, Berlin, 2003, Berlin: Springer-Verlag, S. 201
[30] http://www.webdesign-in.de/mts/google-bestraft-display-none-visibility-hidden/, Zugriff am 1.1.2009 18:03Uhr
[31] http://www.cms-ranking.de/doorwaypages.html, Zugriff am 2.1.2009 um 23:10Uhr
[32] http://www.heise.de/newsticker/Google-sperrt-nun-auch-deutsche-Webseiten-mit-versteckten-Suchwoertern-aus--/meldung/69230, Zugriff am 2.1.2009 um 23:54Uhr
[33] http://www.gutefrage.net/frage/kann-mir-wer-erklaeren-was-http_user_agent-macht-bzw-was-man-damit-fuer-suchmaschinenoptimierung-machen-muss, Zugriff am 12.01.2009 um 02:14Uhr
[34] http://www.drweb.de/magazin/ip-cloaking/, Zugriff am 12.02.2009 um 02:26Uhr
[35] http://www.sistrix.de/news/724-referrer-spam-von-google-und-microsoft.html , Zugriff am 12.020.2009 um 19:25Uhr
[36] http://www.tecchannel.de/webtechnik/entwicklung/1766517/google_optimierung_die_schmutzigen_tricks/index7.html, Zugriff am 13.01.2009 um 00:05Uhr
[37] http://www.tecchannel.de/webtechnik/entwicklung/1766517/google_optimierung_die_schmutzigen_tricks/index7.html, Zugriff am 12.01.2009 um 22.58Uhr
[38] http://www.tecchannel.de/webtechnik/entwicklung/1768122/google_optimierung_die_verbotenen_spam_methoden/index7.html, Zugriff am 13.01.2009 um 00:40Uhr
[39] http://de.wikipedia.org/wiki/Logdatei, Zugriff am 13.01.2009 um 00:54Uhr
[40] J. Winkler, *Suchmaschinenoptimierung*, Franzis Verlag GmbH: Poing, 2007 , S. 119
[41] http://www.tecchannel.de/webtechnik/entwicklung/1768122/google_optimierung_die_verbotenen_spam_methoden/index7.html, Zugriff am 13.01.2009 um 03:45Uhr
[42] http://uwe.vg/2008/06/27/referrer-spam-nervt-muellt-alles-zu-und-nervt/, Zugriff am 13.01.2009 um 04:00Uhr
[43] M. Glöggler, *Suchmaschinen im Internet*, Springer, Berlin, 2003, Berlin: Springer-Verlag, S. 202
[44] http://www.tecchannel.de/webtechnik/entwicklung/1766517/google_optimier, Zugriff am 14.01.2009 um 12.23Uhr
[45] http://blocati.de/2006/08/17/spammer-content-grabber-und-anderes-gesindel-unerwuenscht/, Zugriff am 13.01.2009 um 18:05Uhr
[46] F. Neumeier, *Websites optimieren für Google&Co. schnell + kompakt*, 1. Auflage, 2007, entwickler.press, S. 88
[47] F. Neumeier, *Websites optimieren für Google&Co. schnell + kompakt*, 1. Auflage, 2007, entwickler.press, S. 92
[48] F. Neumeier, *Websites optimieren für Google&Co. schnell + kompakt*, 1. Auflage, 2007, entwickler.press, S. 93
[49] F. Neumeier, *Websites optimieren für Google&Co. schnell + kompakt*, 1. Auflage, 2007, entwickler.press, S. 106
[50] F. Neumeier, *Websites optimieren für Google&Co. schnell + kompakt*, 1. Auflage, 2007, entwickler.press, S. 107