

FACHHOCHSCHULE NORDWESTSCHWEIZ
DISKRETE STOCHASTIK, HS19

Bloom Filter

BONUSAUFGABE

Stefan Gruber
Roger Kreienbühl
Florian Thiévent

3. Dezember 2019

Inhaltsverzeichnis

1	Idee des BloomFilters	2
1.1	Vorteile	2
1.2	Nachteile	2
2	Beispiel aus der Praxis	3
2.1	Google Chrome	3
3	Testergebnisse der Implementierung	4
3.1	Verfahren	4
3.2	Resultate	4

Kapitel 1

Idee des BloomFilters

1.1 Vorteile

1.2 Nachteile

```
1 // Hello.java
2 import javax.swing.JApplet;
3 import java.awt.Graphics;
4
5 public class Hello extends JApplet {
6     public void paintComponent(Graphics g) {
7         g.drawString("Hello, world!", 65, 95);
8     }
9 }
```

Kapitel 2

Beispiel aus der Praxis

2.1 Google Chrome

Der weitverbreitete Browser Google Chrome benutzt Bloom Filter in seiner Malicious URL Implementierung. Dabei werden URL's die von Usern eingegeben werden durch die Browser Engine geprüft und bei einem positiven Match der User mittels einer Meldung darauf aufmerksam gemacht.

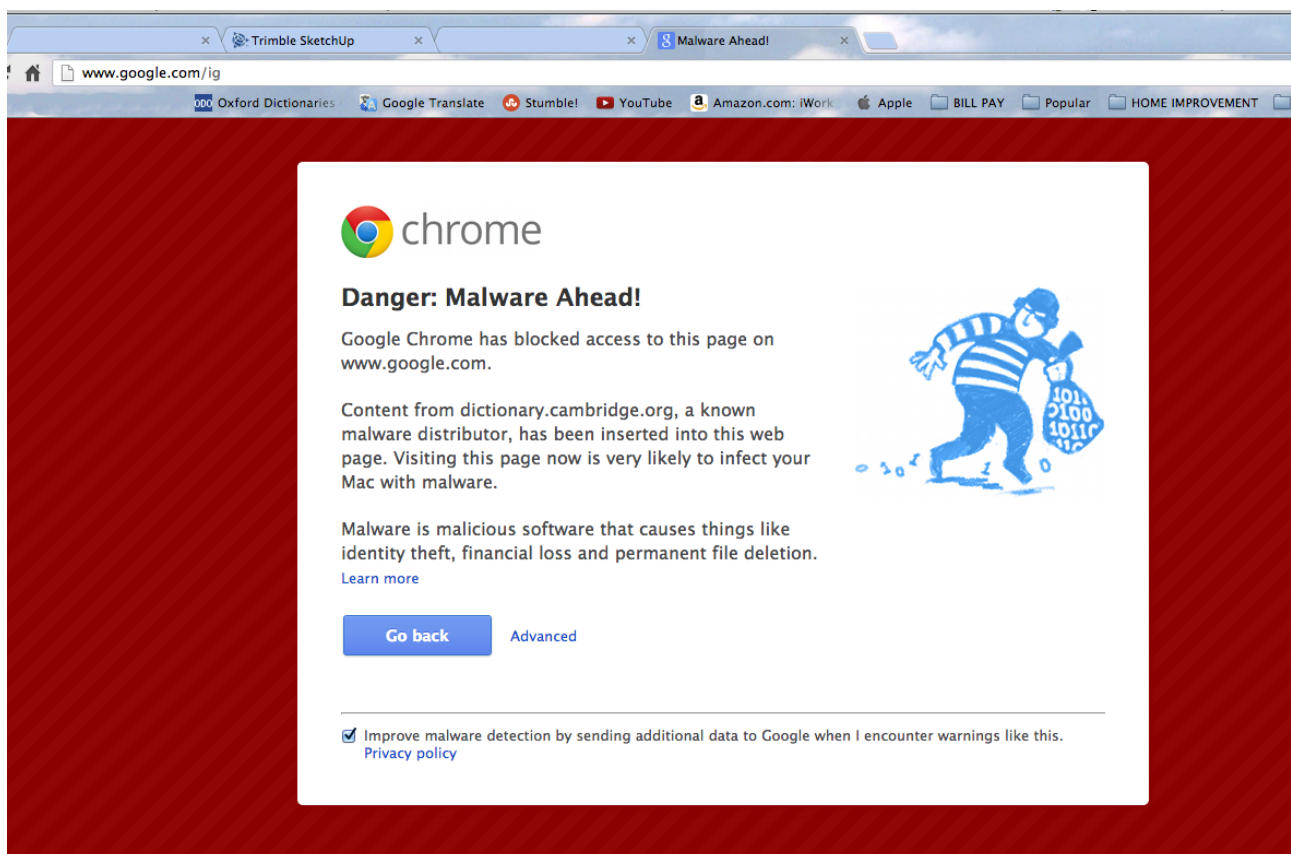


Abbildung 2.1: Google Chrome Malware Hinweis

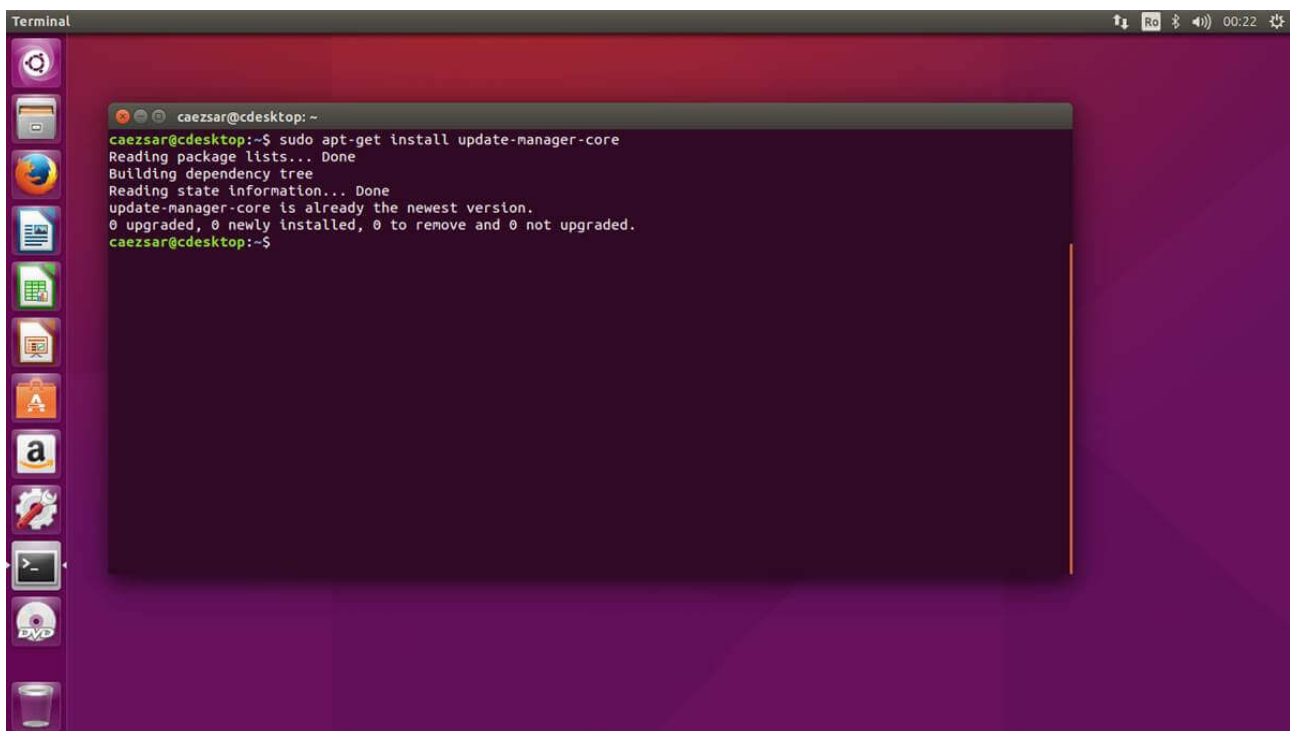
Die Verwendung eines Bloom Filters bietet sich hier an, da der verwendete Speicherplatz sehr klein ist und die Kommunikation zur Malicious URL API von Google sich damit einen enorm kleinen Footprint hat. Das bedeutet, dass die Abgleiche auch bei einer langsamen Internetverbindung performant durchgeführt werden können.

Kapitel 3

Testergebnisse der Implementierung

3.1 Verfahren

3.2 Resultate



The image shows a terminal window on a Linux desktop. The terminal output is as follows:

```
caezsar@cdesktop: ~  
caezsar@cdesktop:~$ sudo apt-get install update-manager-core  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
update-manager-core is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
caezsar@cdesktop:~$
```