

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	5
3.1	Verfahren	5
3.2	Resultate	5

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.



Danger: Malware Ahead

Google Chrome has blocked access to www.google.com.

Content from dictionary.cambodia.com, a known malware distributor, has been blocked on this page. Visiting this page now is dangerous. Your Mac with malware.

Malware is malicious software designed to steal information, such as identity theft, financial loss and more.

[Learn more](#)

[Go back](#)

[Advanced](#)

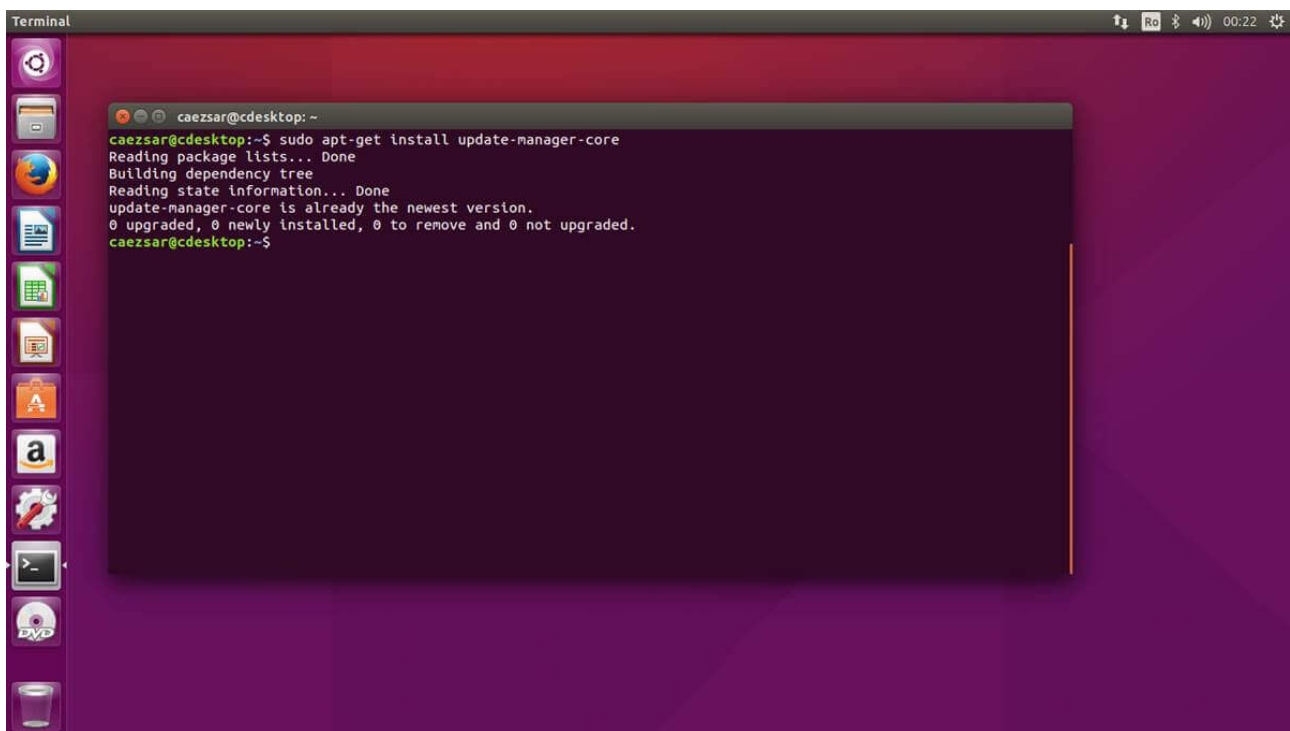
☒ Improve malware detection by sending usage data to Google
[Privacy policy](#)

Kapitel 3

Testergebnisse der Implementierung

3.1 Verfahren

3.2 Resultate



The screenshot shows a terminal window titled "Terminal" on a Linux desktop with a purple background. The terminal displays the following text:

```
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:~$
```