StegFS

Peter Baumann

siprbaum@informatik.stud.uni-erlangen.de

FAU Erlangen-Nürnberg

Steganographie, was ist das ?

- Informationen in einem Trägermedium verstecken

Steganographie, was ist das ?

- Informationen in einem Trägermedium verstecken
- soll für den Betrachter möglichst unsichtbar sein

Steganographie, was ist das ?

- Informationen in einem Trägermedium verstecken
- soll für den Betrachter möglichst unsichtbar sein
- Einsatz bei Medien, die für jeden Zugänglich sind

Steganographie, was ist das ?

- Informationen in einem Trägermedium verstecken
- soll für den Betrachter möglichst unsichtbar sein
- Einsatz bei Medien, die für jeden Zugänglich sind

Häufig eingesetze Trägermedien:

- Bilder
- Tonträger (mp3, wav, ...)
- Filme (avi, mpg, ...)

Beispiel für Steganographie

Hallo Leute! Wir genießen nun endlich unsere
Ferien auf dieser Insel vor Spanien. Wetter gut, Zimmer
ok, ebenso das Essen.
Viele Grüße P. B.

Beispiel für Steganographie

Hallo Leute! Wir genießen nun endlich unsere
Ferien auf dieser Insel vor Spanien. Wetter gut, Zimmer
ok, ebenso das Essen.
Viele Grüße P. B.

Buchstabenanzahl zwischen Leerzeichen:
gerade = 0 ungerade = 1

Beispiel für Steganographie

Hallo Leute! Wir genießen nun endlich unsere
Ferien auf dieser Insel vor Spanien. Wetter gut, Zimmer
ok, ebenso das Essen.
Viele Grüße P. B.

Buchstabenanzahl zwischen Leerzeichen:
gerade = 0 ungerade = 1

01010011|01001111|01010011

Beispiel für Steganographie

Hallo Leute! Wir genießen nun endlich unsere
Ferien auf dieser Insel vor Spanien. Wetter gut, Zimmer
ok, ebenso das Essen.
Viele Grüße P. B.

Buchstabenanzahl zwischen Leerzeichen:
gerade = 0 ungerade = 1

01010011|01001111|01010011

S O S

Strukturen von Dateisystemen

Blockverwaltung:

- Blockallokierungstabelle zum Markieren belegter Blöcke, häufig als Bitliste implementiert
- zusätzliche Freispeicherverwaltung aus Performancegründen

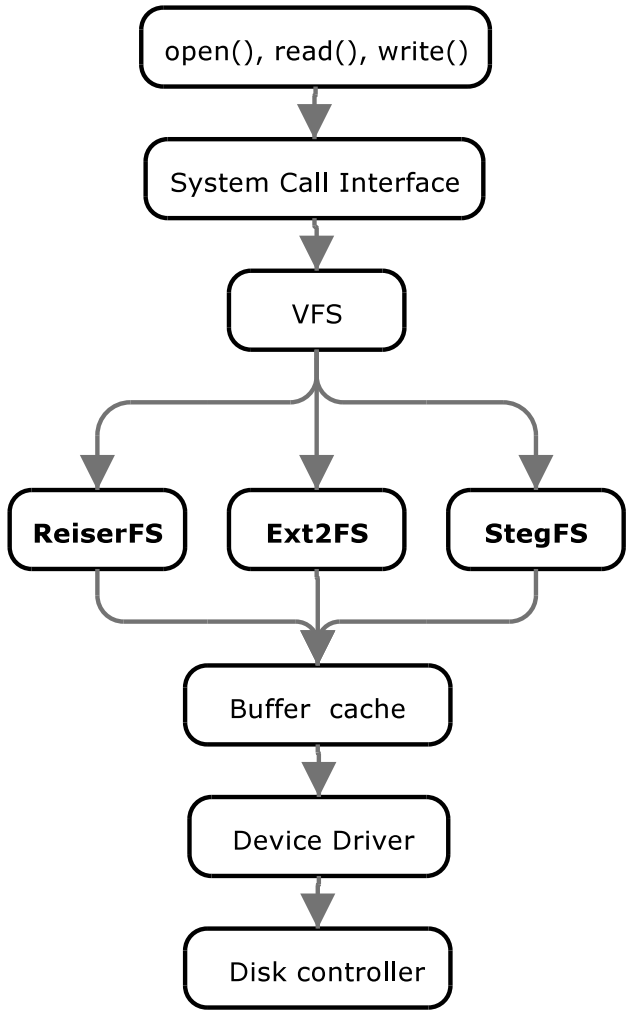
StegFS

- Daten werden in freie Blöcke der Platte versteckt
- Verwenden einer “normalen” Datenpartition, d.h. es kann wahlweise zwischen Ext2 und StegFS gewechselt werden

Anforderungen:

- Transparente Benutzung
- Kompatibilität zum ursprünglichen Dateisystem
- Sicherheit vor potentiellen Angreifern

Integration in den Kernel



Transparenz:

- Integration in den Kernel (evtl. modular)
- volle Funktionalität des Dateisystems bleibt erhalten

Kompatibilität, Sicherheit

StegFS reimplementiert Ext2 und erweitert es um folgende Funktionen:

- neue Dateien werden an zufälligen Positionen auf die Platte geschrieben
- freier Speicherplatz wird mit Rauschen gefüllt
- implementiert eine eigene Blockallokierungstabelle
- Unterstützung von verschiedenen Sicherheitsstufen mit zusätzlicher Verschlüsselung

Sicherheitsstufen (1)

StegFS kann die Daten

- ganz “normal” abspeichern (nicht versteckt, nicht verschlüsselt)
- versteckt und verschlüsselt abspeichern
  - bis zu 15 verschiedene Sicherheitsstufen
  - jede Stufe wird mit einem eigenen Key verschlüsselt
  - Stufen höherer Ordnung enthalten alle Stufen niederer Ordnung

Sicherheitsstufen (2)

- Daten aus Stufen höherer Ordnung als die aktuelle Sicherheitsstufe sind weder für den Anwender noch für StegFS sichtbar

verwendete Crypto-Algorithmen:

- RC6
- Serpent
- Algorithmen, die 128 Bit Blöcke verarbeiten

Keymanagement

Speicherung der Schlüssel in einem Array.
Zugriff auf die Spalten erfolgt mit dem Passwort

SS_1	SS_1	SS_1	$\ldots$	SS_1
	SS_2	SS_2	$\ldots$	SS_2
		SS_3	$\ldots$	SS_3
			$\ldots$	SS_4
			$\ldots$	$\vdots$
				SS_{15}

Blockallokationstabelle

Blockallokationstabelle:

```
struct stegs_btable {
    uint32_t magic1;     /* immer 0                    */
    uint16_t magic2;     /* 0 fuer Daten, 1 fuer Inode */
    uint16_t iv;
    uint32_t bchecksum;  /* Checksumme des Blocks      */
    uint32_t ino;        /* Inode-Nummer des Datei     */
}
```

- 128 Bit pro Eintrag im Gegensatz zu 1 Bit
- Einträge werden ebenfalls verschlüsselt abgespeichert
- Tabelle ist immer gleich groß
- Freie Einträge werden mit Zufallsdaten befüllt

Inodes

Unterscheidung auf verschlüsselt oder unverschlüsselt erfolgt bereits anhand der Inode-Nummer:

Ext2-Inode:

0	0	Rest der InodeNr.

StegFS-Inode:

0	1	level	Rest der InodeNr.

Schwierigkeiten (1)

Dateien nicht geöffneter Sicherheitsstufen können überschrieben werden.

Schwierigkeiten (1)

Dateien nicht geöffneter Sicherheitsstufen können überschrieben werden.

Lösung: Daten werden redundant abgespeichert

Schwierigkeiten (1)

Dateien nicht geöffneter Sicherheitsstufen können überschrieben werden.

Lösung: Daten werden redundant abgespeichert

- bis zu 14 Kopien der Datenblöcke
- bis zu 28 Kopien der Inodes

Schwierigkeiten (1)

Dateien nicht geöffneter Sicherheitsstufen können überschrieben werden.

Lösung: Daten werden redundant abgespeichert

- bis zu 14 Kopien der Datenblöcke
- bis zu 28 Kopien der Inodes

Anzahl der Duplikate ist vom Benutzer einstellbar und wird beim Schreibzugriff auf eine Datei geprüft und gegebenfalls korrigiert

Schwierigkeiten (2)

Duplikate ermöglichen Angreifern das Auffinden versteckter Dateien, indem einfach nach Datenblöcken mit selben Inhalt gesucht wird.

Schwierigkeiten (2)

Duplikate ermöglichen Angreifern das Auffinden versteckter Dateien, indem einfach nach Datenblöcken mit selben Inhalt gesucht wird.

Lösung: Verwendung verschiedener Blockschlüssel

$$BL_{key} = SS_{key} \oplus Blocknummer$$

Handling

Erzeugen einer StegFS Partition:

```
$ mkstegfs /dev/device /path/to/btab
```

Handling

Erzeugen einer StegFS Partition:

```
$ mkstegfs /dev/device /path/to/btab
```

Mounten der Partition:

```
$ mount /dev/device /mnt/mntpoint -tstegfs
   -obtab=/path/to/btab
```

Handling

Erzeugen einer StegFS Partition:

```
$ mkstegfs /dev/device /path/to/btab
```

Mounten der Partition:

```
$ mount /dev/device /mnt/mntpoint -tstegfs
   -obtab=/path/to/btab
```

Öffnen der Sicherheitsstufen:

```
$ stegfsopen /mnt/mntpoint N
```

Handling

Erzeugen einer StegFS Partition:

```
$ mkstegfs /dev/device /path/to/btab
```

Mounten der Partition:

```
$ mount /dev/device /mnt/mntpoint -tstegfs
   -obtab=/path/to/btab
```

Öffnen der Sicherheitsstufen:

```
$ stegfsopen /mnt/mntpoint N
```

Schliesen der Sicherheitsstufen:

```
$ stegfsclose /mnt/mntpoint N
```

Beispiel: Praktischer Einsatz

```
$ stegfsopen /mnt 3
```

Beispiel: Praktischer Einsatz

```
$ stegfsopen /mnt 3
$ cd /mnt/stegfs
```

Beispiel: Praktischer Einsatz

```
$ stegfsopen /mnt 3
$ cd /mnt/stegfs
$ ls
drwx---- 5 peter users 512 Jun 5 15:35 .
drwx---- 7 peter users 512 Jun 5 15:09 ..
drwx---- 2 peter users 512 Jun 5 15:35 1
drwx---- 2 peter users 512 Jun 5 15:35 2
drwx---- 2 peter users 512 Jun 5 15:35 3
```

Beispiel: Praktischer Einsatz

```
$ stegfsopen /mnt 3
$ cd /mnt/stegfs
$ ls
drwx---- 5 peter users 512 Jun 5 15:35 .
drwx---- 7 peter users 512 Jun 5 15:09 ..
drwx---- 2 peter users 512 Jun 5 15:35 1
drwx---- 2 peter users 512 Jun 5 15:35 2
drwx---- 2 peter users 512 Jun 5 15:35 3
$ cp geheimnis /mnt/stegfs/3
```

Beispiel: Praktischer Einsatz

```
$ stegfsopen /mnt 3
$ cd /mnt/stegfs
$ ls
drwx---- 5 peter users 512 Jun 5 15:35 .
drwx---- 7 peter users 512 Jun 5 15:09 ..
drwx---- 2 peter users 512 Jun 5 15:35 1
drwx---- 2 peter users 512 Jun 5 15:35 2
drwx---- 2 peter users 512 Jun 5 15:35 3
$ cp geheimnis /mnt/stegfs/3
$ stegfsclose /mnt 0
```

Beispiel: Praktischer Einsatz

```
$ stegfsopen /mnt 3
$ cd /mnt/stegfs
$ ls
drwx---- 5 peter users 512 Jun 5 15:35 .
drwx---- 7 peter users 512 Jun 5 15:09 ..
drwx---- 2 peter users 512 Jun 5 15:35 1
drwx---- 2 peter users 512 Jun 5 15:35 2
drwx---- 2 peter users 512 Jun 5 15:35 3
$ cp geheimnis /mnt/stegfs/3
$ stegfsclose /mnt 0
$ ls
drwx---- 5 peter users 512 Jun 5 15:35 .
drwx---- 7 peter users 512 Jun 5 15:09 ..
```

Zusammenfassung

- StegFS ist leicht in vorhandenes System zu integrieren

Zusammenfassung

- StegFS ist leicht in vorhandenes System zu integrieren
- es werden 15 verschiedene Sicherheitsstufen unterstützt

Zusammenfassung

- StegFS ist leicht in vorhandenes System zu integrieren
- es werden 15 verschiedene Sicherheitsstufen unterstützt
- Lesegeschwindigkeit ist vergleichbar mit Ext2

Zusammenfassung

- StegFS ist leicht in vorhandenes System zu integrieren
- es werden 15 verschiedene Sicherheitsstufen unterstützt
- Lesegeschwindigkeit ist vergleichbar mit Ext2
- Schreibgeschwindigkeit sehr stark abhängig von der Duplikatanzahl

Zusammenfassung

- StegFS ist leicht in vorhandenes System zu integrieren
- es werden 15 verschiedene Sicherheitsstufen unterstützt
- Lesegeschwindigkeit ist vergleichbar mit Ext2
- Schreibgeschwindigkeit sehr stark abhängig von der Duplikatanzahl
- kein hundertprozentiger Schutz vor überschreiben

Zusammenfassung

- StegFS ist leicht in vorhandenes System zu integrieren
- es werden 15 verschiedene Sicherheitsstufen unterstützt
- Lesegeschwindigkeit ist vergleichbar mit Ext2
- Schreibgeschwindigkeit sehr stark abhängig von der Duplikatanzahl
- kein hundertprozentiger Schutz vor überschreiben
- Benutzer kann plausibel abstreiten, versteckte Daten zu benutzen

Danke für Eure Aufmerksamkeit