Deutschland
Land der Ideen
Ausgewählter Ort 2008

GEORG-SIMON-OHM
HOCHSCHULE NÜRNBERG

Benutzerfreundliche, flexible und sichere Konfiguration von AppArmor

Peter Trommler

Fakultät Informatik

Georg-Simon-Ohm-Hochschule

Nürnberg

Agenda

- Motivation
- Erklärte Sicherheitsprofile (ESPE)
- Sicherheit
- AppArmor Prototyp
- Ausblick

Motivation

- Dateiviewer mit Netzzugriff
- Emails mit Viren/ Würmern
- Trojanische Pferde

- Lösung: Zugriff einschränken

Zugriffskontrollmechanismen

- Linux Security Modules Schnittstelle
  - Linux AppArmor
  - Security Enhanced Linux
- Solaris Capabilities

- Konzept: Referenzmonitor
- Zugriffskontrolle für Programm (nicht nur Benutzer)

Konfiguration der Zugriffskontrolle

- Welchen Zugriff benötigt die Anwendung?
- Welcher Zugriff ist gefährlich?
- Welcher Zugriff ist angemessen?

- Wie kann der Benutzer diese Fragen beantworten?

Benutzerfreundlich, flexibel und sicher

- Benutzerfreundlich
  - Benutzer muss nichts tun
  - Benutzer muss etwas tun, das er versteht
- Flexibel
  - Benutzer kann unter verschiedenen Profilen wählen
  - Kooperation des Programmautors nicht erforderlich
- Sicher
  - Schutz vor Manipulation der Plattform (AppArmor)
  - Schutz vor gefälschten Profilen

Vertrauen in Experten

- Experte spezifiziert Zugriffsrechte (Policy)
- Benutzer vertraut dem Experten
- Alternative Policies: Wie wählt der Benutzer?

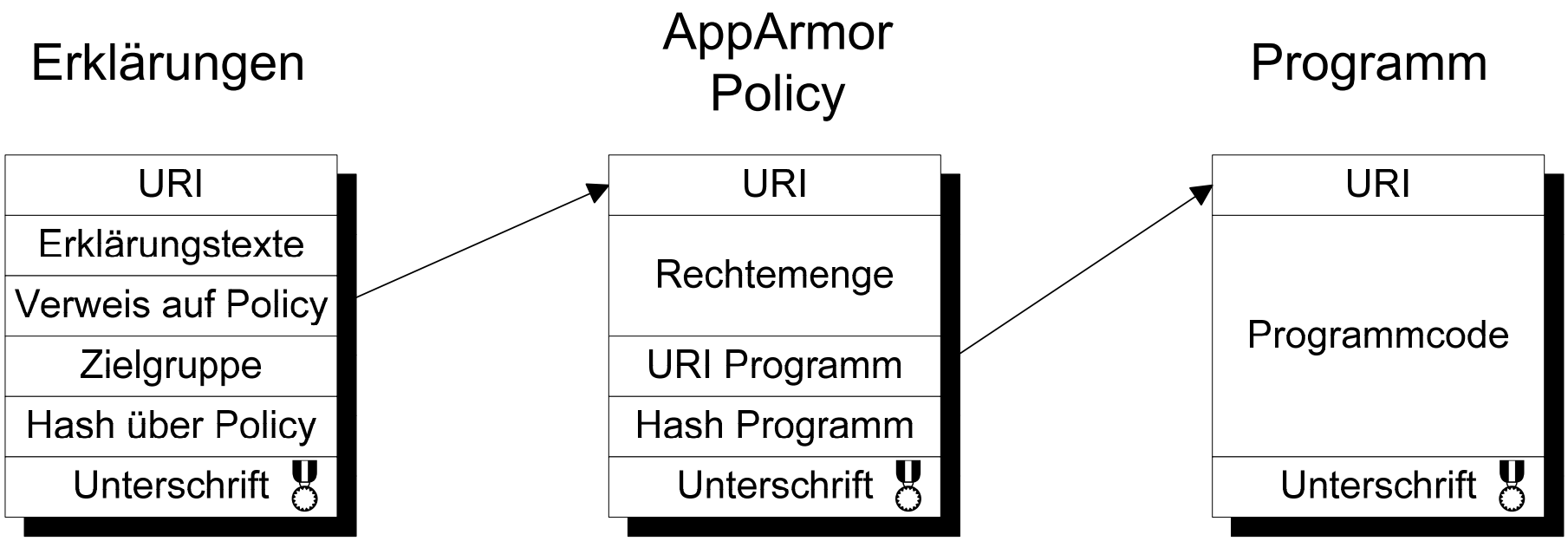
Erklärte Sicherheitsprofile

- Policy in natürlicher Sprache erklärt
- Experte liefert Erklärung
- Zielgruppenspezifische Erklärungen
- Benutzer wählt Policy an Hand der Erklärung
- Gewählte Policy wird installiert

Programm, Policy und Erklärung

- Eine Datei
- Alle jeweils separat
- Programm und Policy in einer Datei
- Policy und Erklärung in einer Datei

Sicheres Verbinden



Bedeutung der Cryptographie

- Programm
  - signiert durch Urheber
  - Integrität und Authentizität
- Policy
  - geschrieben für eine Anwendung
  - nicht geschrieben für ein Programm (Versionen!)
  - Integrität und Authentizität

Bedeutung der Cryptographie (Forts.)

- Erklärung
  - erklärt eine Policy
  - Hash stellt Integrität der Policy sicher
  - Signatur: Integrität und Authentizität der Erklärung

Policy Datei

- Autor
  - Name
  - Public Key
- URI
  - Identifikator der Policy
- Programm
  - Name, Quelle
  - Hash über Programm (opt.)
- Policy-Dateien
  - Hauptdatei
  - Includes

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<espe:policyfile xmlns:espe="http://www.informatik.fh-
     nuernberg.de/espe">
   <espe:author>
      <espe:lastname>Doe</espe:lastname>
      <espe:pubkeyURI>http://www.example.com/~john/crt.pem
      </espe:pubkeyURI>
   </espe:author>
   <espe:details>
  <espe:URI>http://www.example.com/proto.epo</espe:URI>
   </espe:details>
   <espe:program>
      <espe:programName version="1.0">ESPE
     Prototyp</espe:programName>
      <espe:programURI>http://www.fh-
     nuernberg.de/espe/prototyp.zip</espe:programURI>
      <espe:programHash
     algorithm="sha1">123...456</espe:programHash>
   </espe:program>
   <espe:policy>
      <espe:profile><![CDATA[ ...]]></espe:profile>
      <espe:include name="i1"><![CDATA[...]]></espe:include>
   </espe:policy>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ...
   </Signature>
</espe:policyfile>
```

Erklärungsdatei

- Erklärer
  - Name
  - Public Key
- Details
  - Identifikator der Erklärung
- Policy
  - URI
  - Hash über Policy
- Erklärungen
  - Zielgruppe
  - Sprache

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"
    ?>
<espe:explanation xmlns:espe="http://www.fh-
    nuernberg.de/espe">
   <espe:author>
      <espe:lastname>Meinelt</espe:lastname>
      <espe:pubkeyURI>http://www.fh-
     nuernberg.de/mm_cert.pem</espe:pubkeyURI>
   </espe:author>
   <espe:details>
    <espe:URI>http://www.fh-
     nuernberg.de/espe/proto.eex</espe:URI>
   <espe:details>
   <espe:policy>
      <espe:policyURI>http://www.fh-
     nuernberg.de/espe/proto.epo</espe:policyURI>
      <espe:policyHash
     algorithm="sha1">ad1adfe6...7e621385c28f38</espe:
     policyHash>
   </espe:policy>
   <espe:text level="expert" language
     ="en"><![CDATA[Example Expert]]> </espe:text>
   <espe:text level="user" language="en">Example
     User</espe:text>
   <Signature
     xmlns="http://www.w3.org/2000/09/xmldsig#">
        ...
   </Signature>
</espe:explanation>
```

Versionsdatei

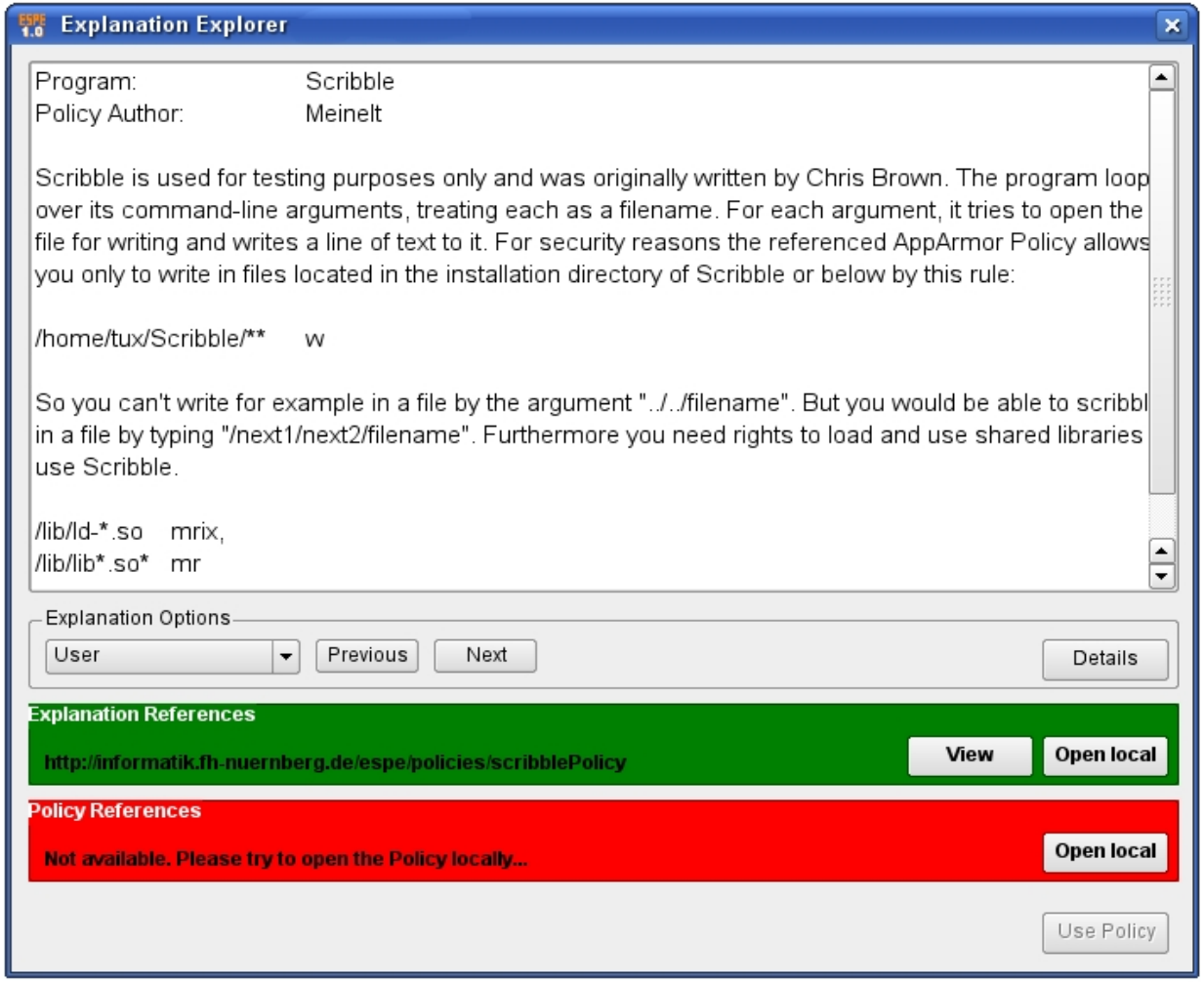
- Programm-Autor
  - Name
  - Public Key
- Details
  - Identifikator der Versionsdatei
- Programm
  - Name
- Versionen
  - Versionsnummern
  - Hash

```
<? xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<espe:updatefile xmlns:espe="http://www.informatik.fh-
    nuernberg.de/espe">
  <espe:author>
     <espe:firstname>Bill</espe:firstname>
     <espe:lastname>Apple</espe:lastname>

    <espe:pubkeyURI>http://www.billapple.local/cert.pem</es
    pe:pubkeyURI>
  </espe:author>
  <espe:details>

    <espe:URI>http://www.billapple.local/updatefile.eup</es
    pe:URI>
  </espe:details>
  <espe:program>
     <espe:programName>ESPE Prototyp</espe:programName>
  </espe:program>
  <espe:hashes>
     <espe:hash algorithm="sha1"
    version="0.8">7...9</espe:hash>
     <espe:hash algorithm="sha1" version="1.0">9...1
    </espe:hash>
     <espe:hash algorithm="sha1"
    version="1.1">d5f...99a</espe:hash>
  </espe:hashes>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
          ...
  </Signature>
</espe:updatefile>
```

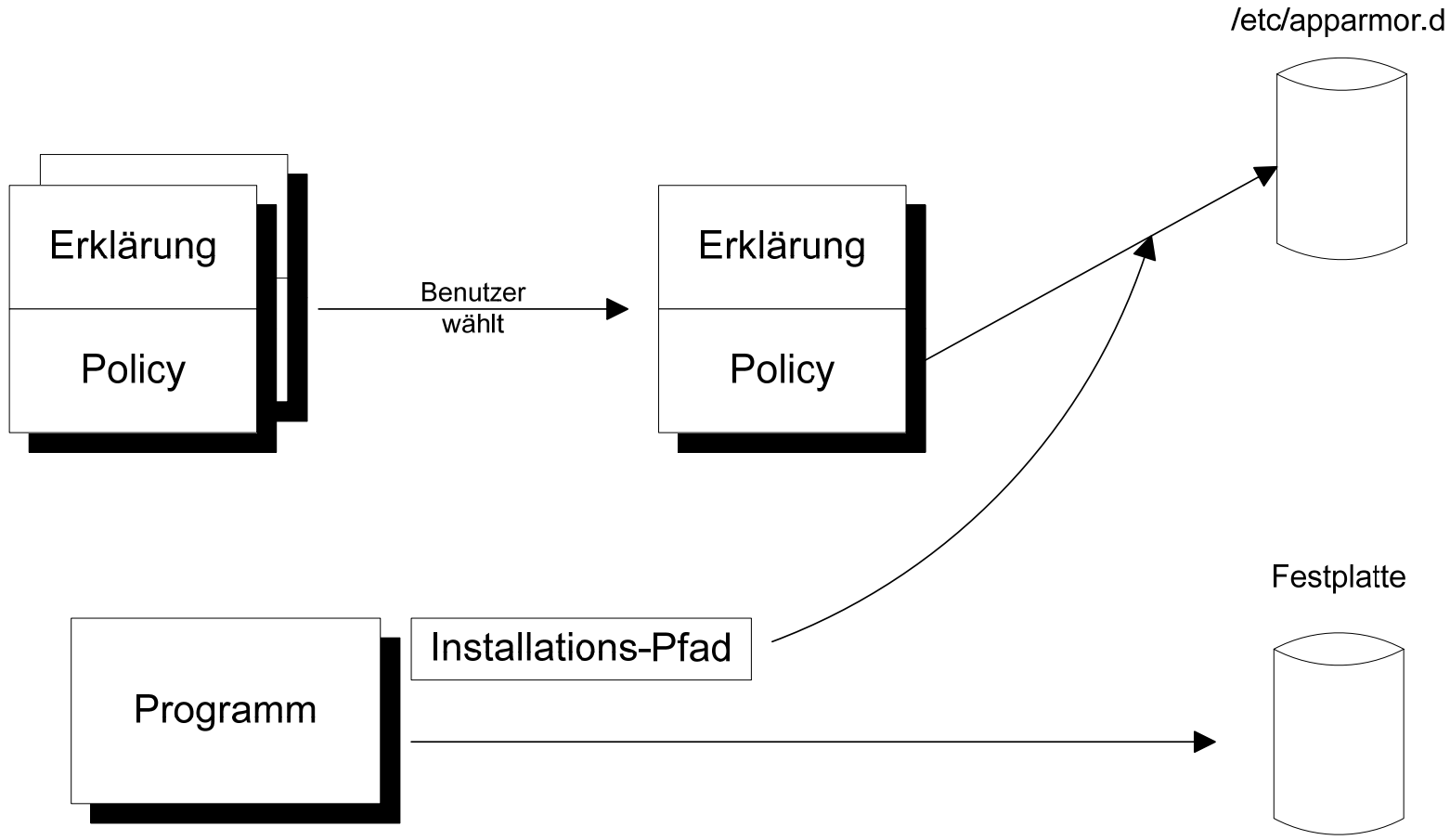
Prototyp


Explanation Explorer
Program: Scribble
Policy Author: Meinelt
Scribble is used for testing purposes only and was originally written by Chris Brown. The program loop
over its command-line arguments, treating each as a filename. For each argument, it tries to open the
file for writing and writes a line of text to it. For security reasons the referenced AppArmor Policy allows
you only to write in files located in the installation directory of Scribble or below by this rule:
/home/tux/Scribble/** w
So you can't write for example in a file by the argument "../../filename". But you would be able to scribbl
in a file by typing "/next1/next2/filename". Furthermore you need rights to load and use shared libraries
use Scribble.
/lib/ld-*.so mrix,
/lib/lib*.so* mr
Explanation Options
User
Previous
Next
Details
Explanation References
http://informatik.fh-nuernberg.de/espe/policies/scribblePolicy
View
Open local
Policy References
Not available. Please try to open the Policy locally...
Open local
Use Policy

Installation eines Programms

- Benutzer
  - Programm, Policies und Erklärungen bereitstellen
  - Erklärung auswählen und damit Policy
  - Programmpfad festlegen (AppArmor)
- ESPE
  - Prüfen der Signaturen (Erklärung und Programm)
  - Prüfen des Hashes (Policy)
  - Extrahieren der Policy und Includes aus CDATA-Abschnitten
  - Kopieren der Policy in AppArmor-Verzeichnis

Installation (Forts.)



Offene Punkte

- Beispiel: Dateien im „Web-Ordner“
  - Policy: Zugriff auf alle mit Wildcard oder alle Dateien
  - Benötigt: eigener Web-Ordner (Benutzername)
  - Lösung: konfigurierbare Policies
- Erweiterung von ESPE um Wizard
- Beispiel: Editor
  - Policy: Zugriff auf alle Dateien
  - Benötigt: Zugriff auf die zu bearbeitende Datei
  - Lösung: vertrauenswürdiger Dialog
  - Allgemein: Policy für Programmaufruf (nicht nur Programm)
- Neues Linux Security Module

Zusammenfassung

- Programme mit eingeschränktem Zugriff
- Problem: Konfiguration der Zugriffskontrolle
- Lösung: Erklärungen
- Offen
  - Anpassung an lokale Konfiguration
  - Dynamisches Profil abgeleitet aus Benutzerinteraktion
- Prototyp mit AppArmor