Multifaktorielle Benutzerauthentifizierung durch PAM und Fingerabdruck für Linux Infrastrukturen

Claus Vielhauer, Jana Dittmann, Christian Helmholz

Fakultät für Informatik, AMSL
Otto-von-Guericke Universität Magdeburg
Universitätsplatz 2
D-39016 Magdeburg
{Claus.Vielhauer|Jana.Dittmann}@iti.cs.uni-magdeburg.de

Abstract: Dieser Beitrag befasst sich mit der Problematik der Benutzerauthentifizierung in Rechnernetzen. Es wird ein neues Verfahren für Linux-Infrastrukturen vorgestellt, welches im Vergleich zu gängigen Verfahren, beispielsweise Single Sign-On Systemen wie Kerberos, verschiedene Modalitäten wie Besitz, Wissen und Sein unterstützt. Dazu kann innerhalb der Infrastruktur personen-, rechner- und applikationsbezogen konfiguriert werden, welche Modalitäten und welche Sensoren (biometrische und nicht-biometrische) für die Erfassung dieser Modalitäten zum Einsatz kommen. Es wird die zugrunde liegende Systemarchitektur vorgestellt, welche in einen ersten Prototyp, unter Einsatz eines biometrischen Fingerabdruckvergleichs, implementiert wurde. Erste Testergebnisse werden präsentiert, welche die Anwendbarkeit des gewählten Ansatzes aufzeigen.

1 Einführung

In der Geschichte haben Verfahren zum Nachweis der Authentizität von Personen oder Gegenständen eine lange Tradition. Das Wort authentisch stammt vom lateinischen Wort *authenticus* und bedeutet echt, den Tatsachen entsprechend und daher glaubwürdig. So signierten bereits vor mehr als 2000 Jahren chinesische Töpfer, die von ihnen hergestellten Vasen und Krüge mit ihrem Daumenabdruck, um deren Echtheit zu garantieren. Ebenso dienten schon im Mittelalter königliche Siegel der Authentizität von Urkunden und Briefen. Auch heute noch werden in Ämtern und Behörden zu diesem Zweck Dienstsiegel benutzt. In der jüngeren Vergangenheit wurden technische Verfahren entwickelt, welche zur automatisierten Authentifizierung von Personen eingesetzt werden können.

In der Einführung zu diesem Artikel werden wir in diesem Kapitel zunächst verschieden Modalitäten für den Zweck einer solchen Benutzerauthentifizierung diskutieren und die Zielsetzung des vorliegenden Artikels beschreiben. Das zweite Kapitel befasst sich mit einer kurzen Einführung zum Stand der Technik im Bereich der Authentifizierung in vernetzten Systemen und zeigt offene Probleme auf. Im Kapitel 3 wird dann das von uns entworfene, neue Konzept für die Problemstellung vorgestellt, welches im Rahmen unserer Aktivitäten prototypisch implementiert wurde. Ergebnisse aus ersten Tests mit diesem Prototyp werden im vierten Kapitel präsentiert und mit einer Zusammenfassung und Ausblick schließt der Beitrag im Kapitel 5.

1.1 Modalitäten zur Authentifizierung: Besitz, Wissen und Sein

Bei der Authentifizierung von Personen wird deren vorgegebene Identität überprüft, oder die Zugehörigkeit zu einer Personengruppe bestimmt. Dies dient meist der Gewähr bestimmter Rechte oder Privilegien. Eine Identifizierung ist der Versuch, einer unbekannten Person ihre tatsächliche Identität zuzuordnen.

Sowohl zur Authentifizierung als auch zur Identifizierung von Personen wird mindestens eine der folgenden drei Klassen verschiedener Modalitäten benutzt [Bi03]

- **Besitz** (z.B. durch Schlüssel, Smartcards, Eintrittskarte)
- **Wissen** (z.B. durch Passwörter, Zahlenkombinationen (Personal Identification Number, PIN), Geburtsdatum)
- **Sein** (durch passive biometrische Merkmale, z.B. Fingerabdruck, Gesichtsgeometrie, Iris oder aktive biometrische Merkmale, z.B. Unterschrift, Sprache, Gangart)

Werden in einem Authentifizierungsvorgang Modalitäten aus mehr als einer dieser drei Klassen herangezogen (beispielsweise bei Einsatz von Smartcard und PIN), so wird dies als multifaktorielle Benutzerauthentifikation bezeichnet [Hl04].

Im Bereich der Informationstechnik wurde spätestens mit der Einführung von Mehrbenutzersystemen und verteilten Anwendungen eine Benutzerauthentifizierung notwendig. Bis heute werden hier vor allem Passwörter zur Zugangs- und Zutrittskontrolle eingesetzt, wobei hier Probleme durch die zunehmende Verbreitung zu sichernder Systeme, sowohl im privaten als auch im beruflichen Bereich, zunehmend signifikant werden. Wesentliche Problematiken der wissensbasierten Authentifizierung sind die Merkfähigkeit der Benutzer, willentliche oder unbeabsichtigte Weitergabe sowie die Wahl schwacher Passwörter.

Verfahren, die Merkmale aus den Klassen Besitz und Sein verwenden, können diese Probleme in bestimmten Fällen lösen. Schwierigkeiten bereiten hierbei jedoch verteilte Systeme, da die Authentizität und Integrität einer entfernten Eingabe (z.B. Fingerabdruck) in vernetzten Infrastrukturen gewährleistet sein muss. Zudem ist die Verwendung von biometrischen Daten im Bereich öffentlicher Netze auch deshalb problematisch, weil zur Anonymisierung von biometrischen Referenzdaten bisher nur ansatzweise geeigneten Verfahren (z.B. Einweg-Hashfunktionen, siehe [VSM02], [UPPJ04], [MRLW01]) zur Verfügung stehen und der breite Einsatz solcher Verfahren in der Praxis noch aussteht. Ein Missbrauch der Daten kann somit nicht ausgeschlossen werden.

Die Wahl einer geeigneten Authentifizierungsmethode und der zugehörigen Verfahrensparameter erfordert in der Regel einen Kompromiss zwischen dem erreichbaren Sicherheitsniveau und dem Benutzungskomfort. Die Verwendung von Wissen bedarf einer interaktiven Eingabe und wird durch mögliches Ausspähen gefährdet. Merkmale aus der Klasse Besitz sind komfortabler in ihrer Handhabung, könnten aber kopiert, weitergegeben oder gestohlen werden. Bei einer geeigneten Wahl der Verfahrensparameter ist mit Hilfe biometrischer Methoden unter Umständen zwar eine nahezu fehlerfreie Zuordnung zwischen Personen und ihren jeweiligen Authentifizierungsmerkmalen möglich, d.h. es treten keine Falscherkennungen auf. Auf Grund der relativ hohen Falsch-Rückweisungsrate (FRR) sinkt bei einer solchen Konfiguration jedoch meist der Benutzungskomfort, da Eingabewiederholungen wahrscheinlicher sind. Bei der biometrischen Identifizierung von Personen sind Fingerabdrücke, neben Körpermaßen (z.B. Größe) und Abbildungen des Gesichtes (z.B. Photo, Phantombild), die oft benutzte biometrische Merkmale. So wurden vor gut 100 Jahren erstmals Verbrecher anhand ihrer Fingerabdrücke überführt [Ve02]. Nach den technischen Entwicklungen der letzten Jahre stehen heute gute, relativ preiswerte Fingerabdruck-Scanner zur Verfügung. Damit sind Fingerabdrücke auch außerhalb der Kriminalistik, wie zum Beispiel für die Benutzerauthentifizierung an Computersystemen, von großem Interesse. Es gibt bereits eine Reihe von Forschungsarbeiten auf dem Gebiet der rechnergestützten Verarbeitung von Fingerabdrücken, mit deren Hilfe sich einige Standardtechniken für dieses Aufgabengebiet herausgebildet haben (siehe z.B. [Ho98], [MMJP03], [Pr01]).

In der Praxis hat diese Methode zur Benutzerauthentifizierung aber noch eine geringe Bedeutung. Es gibt einige Hersteller, die einzelne Sensoren sowie kombinierte Geräte wie Tastaturen, Mäuse oder USB-Memory-Sticks mit integriertem Fingerabdrucksensor anbieten. Da sich diese Firmen im Allgemeinen neben dem Vertrieb der Hardware auch um die Entwicklung und Vermarktung einer jeweils eigenen proprietären Authentifizierungssoftware bemühen, gibt es keine einheitlichen Schnittstellen für die jeweiligen Gerätetreiber und Softwarekomponenten. In der Regel stehen keine Hard- oder Software bezogenen Informationen zur Verfügung, welche die Entwicklung eines unabhängigen Authentifizierungssystems für den eigenen Fingerabdruck-Scanner ermöglichen.

Neben den kommerziellen gibt jedoch auch freie Software Projekte (wie z.B. Fingerprint Verification System [FVS04] und Free Fingerprint Imaging Software [FFIS04]), die sich mit der Verarbeitung von Fingerabdrücken beschäftigen. Ihr Ziel ist nicht die Entwicklung eines eigenen Authentifizierungssystems, sondern die Implementierung und Bereitstellung einer Bildverarbeitungssoftware bzw. einer entsprechende Programmbibliothek. Aus diesem Grund enthalten sie weder Gerätetreiber für bestimmte Sensoren noch Ansätze zur Lösung der Authentifizierungsproblematik in verteilten Infrastrukturen.

1.2 Anforderungen an Linux-Infrastrukturen

In diesem Artikel wird der Prototyp eines Rahmenwerkes zur multifaktoriellen Benutzerauthentifizierung beschrieben, mit dessen Hilfe ein skalierbares Sicherheitsniveau realisierbar ist. Zielplattform sind dabei zentral administrierte Unix-, insbesondere Linux-Infrastrukturen (z.B. Rechnerpool eines Unternehmens oder einer Universität). Solche Systeme sind oftmals Teil öffentlicher Rechnernetze, so dass die im Rahmen einer Benutzerauthentifizierung notwendige Kommunikation abgesichert werden muss, z.B. durch kryptographische Methoden. Die Ausstattung der Rechner mit entsprechenden Sensoren und Eingabegeräten kann sich individuell unterscheiden. Jeder Benutzer sollte die Möglichkeit haben, seine Authentifizierungsmerkmale selbständig und ohne Unterstützung durch den Systemadministrator, mit Hilfe des jeweiligen Arbeitsplatzrechners, zu verwalten. Auf Grund der heterogenen Natur eines solchen Systems ist zur Aufrechterhaltung eines definierten Sicherheitsniveaus eine adaptive Auswahl der zur Benutzerauthentifizierung herangezogenen Merkmale notwendig. Der Ablauf einer Benutzerauthentifizierung hängt somit unter anderem vom Benutzer, seinen hinterlegten Authentifizierungsinformationen, sowie dem verwendeten Rechner ab.

Besitzt ein Rechner beispielsweise keinen Fingerabdruck-Scanner, so kann dieses Sicherheitsdefizit durch die Abfrage mehrerer Passwörter ausgeglichen werden. Hat ein Benutzer keine biometrischen Merkmale beim System hinterlegt, so kann er stattdessen eine Smartcard verwenden. Eine Kombination mehrerer Authentifizierungsmerkmale kann verhindern, dass einzelne kompromittierte Informationen zu einem Sicherheitsrisiko werden. Die Interaktion der Anwendungen mit dem Authentifizierungssystem basiert auf einer allgemein akzeptierten Programmierschnittstelle, um so eine möglichst breite Unterstützung aktueller und auch zukünftiger Software-Komponenten zu gewährleisten. Dies ermöglicht außerdem die Kompatibilität zu Rechnersystemen, die einen anderen Authentifizierungsmechanismus verwenden.

2 Stand der Technik: Benutzerauthentifizierung für Linux Infrastrukturen

Sowohl Linux als auch die verschiedenen Unix-Varianten sind Mehrprozess- und Mehrbenutzersysteme. Ihr Sicherheitskonzept verwendet Benutzer (User IDs) und Benutzergruppen (Group IDs), um den Zugriff auf Dateien und Systemressourcen (z.B. Gerätedateien, Shared Memory) zu beschränken. Um sicher zu stellen, dass diese Mechanismen korrekt auf reale Personen umgesetzt werden, ist eine zuverlässige Benutzerauthentifizierung unerlässlich. In diesem Kapitel sollen kurz die wichtigsten heute eingesetzten technischen Methoden zu dieser Problematik vorgestellt und kritisch verglichen werden. Am Ende wird gezeigt, welche Anforderungen von den heutigen Systemen noch nicht ausreichend erfüllt werden, wodurch sich der Forschungsbedarf als Grundlage unserer Arbeiten ergibt.

2.1 Lokale Benutzerauthentifizierung

Informationen über Benutzer und Benutzergruppen werden nicht nur zur Authentifizierung, sondern auch während der Ausführung einiger Anwendungsprogramme benötigt. Intern verwendet das Linux Betriebssystem hierzu anstelle der Namen eindeutig zugeordnete Nummern, so genannte IDs, die den Benutzer und dessen Gruppenzugehörigkeiten kennzeichnen.

Die jeweiligen Zuordnungen sind neben weiteren Informationen in den Dateien */etc/passwd* und */etc/group* hinterlegt, welche im POSIX.1-Standard als Benutzerdatenbank bezeichnet werden [He99]. Der Zugriff auf diese Daten wird über spezielle Funktionen der Standard-Programmbibliothek (z.B. *glibc*) realisiert und ist jedem Benutzer erlaubt. Ursprünglich enthielt die Datei */etc/passwd* auch die Passwörter der Benutzer.

Üblicherweise werden diese dort zwar nicht im Klartext abgelegt, sondern durch Einweg-Hashfunktionen verschlüsselt, jedoch kann auch dieses Vorgehen ein Ausspähen der Passwörter nicht wirkungsvoll verhindern. In modernen Systemen speichert man die Passwörter deshalb in der für normale Benutzer unzugänglichen Datei */etc/shadow*. Ideale Einweg-Hashfunktionen sind Funktionen, die aus einer Eingabe beliebiger Länge eine Ausgabe fester Länge generieren (Kompression). Von der Ausgabe lässt sich nicht direkt auf die Eingabe schließen (Unumkehrbarkeit). Außerdem werden zwei legitimen Eingaben stets unterschiedliche Ausgaben zugeordnet (Kollisionsfreiheit). Typische Einweg-Hashfunktionen zur Verschlüsselung von Passwörtern auf Unix-Systemen sind zum Beispiel MD5 oder SHA ([KPS02] und [GS96]).

Bei der Benutzerauthentifizierung wird das eingegebene Passwort mit Hilfe der Einweg-Hashfunktion verschlüsselt und anschließend mit den hinterlegten Daten verglichen. Zwar ist die Datei */etc/shadow* durch Einschränkung der Zugriffsrechte vor dem Zugriff normaler Benutzer geschützt, trotzdem ist auch hier eine Verschlüsselung sinnvoll. Zum einen verhindert dies, dass der Systemadministrator, der ja Superuser-Rechte besitzt, auf die Klartext-Paßwörter zugreifen kann. Für die Sicherheit des Rechnersystems ist dies zwar irrelevant, bietet aber für den Benutzer ein gewisses Mindestmaß an Datenschutz, falls er sein Passwort auch für andere Zwecke einsetzt. Zum anderen müsste besonders darauf geachtet werden, dass Sicherheitskopien der Benutzerdatenbank physisch sicher (z.B. in einem Tresor) lagern, damit die gespeicherten Klartext-Passwörter geschützt sind.

Trotz der Verschlüsselung sollten Passwörter niemals öffentlich zugänglich sein. Gelingt es, das verschlüsselte Passwort eines Benutzers in Erfahrung zu bringen, so ist es eine übliche Angriffsmethode, alle potentiellen Passwörter ebenfalls zu verschlüsseln und mit dem Passwort des Benutzers zu vergleichen ([He99] und [An01]). Bei einer Übereinstimmung, entspricht das Klartext-Passwort des Benutzers der entsprechenden Eingabe der Einweg-Hashfunktion. Ein erfolgreicher Angriff ist damit nur eine Frage der Zeit. Da viele Benutzer schwache Passwörter wählen, dass heißt Passwörter, die aus realen Wörtern oder Wortteilen zusammengesetzt sind, kann der Vorgang durch Verwendung von Wörterbüchern, bei der Wahl potentieller Passwörter, noch beschleunigt werden. Dieses Vorgehen wird auch als Brute-Force-Suche bezeichnet.

2.2 Netzwerk-Informationsdienste

Bei der Administration ganzer Rechnerpools können Netzwerk-Informationsdienste sehr hilfreich sein. Einer der bekanntesten Vertreter ist NIS (Network Information Service), welcher ursprünglich von der Firma Sun Microsystems entwickelt wurde und im Jahre 1985 unter dem Namen YP (Yellow Pages) auf den Markt kam. NIS bietet potentiellen Angreifern diverse Sicherheitsdefizite und ist deshalb als unsicher einzustufen (siehe [An01] und [SEL01]). Zusammen mit Solaris 2.x wurde später NIS+ (Network Information Service Plus) eingeführt, um einen erweiterten Funktionsumfang und zumindest auf Seiten der Client-Rechner eine komfortablere Administration zu ermöglichen.

Ein NIS-Server stellt unter anderem folgende Informationen zur Verfügung:

- Benutzernamen, Passwörter (*/etc/passwd*, */etc/shadow*),
- Benutzergruppen (*/etc/group*),
- Rechnernamen und Netzwerkadressen (*/etc/hosts*).

Damit kann sich ein Benutzer an jedem Rechner eines Pools mit dem gleichen Passwort authentifizieren, und braucht auch Änderungen nur auf einem der Rechner durchzuführen.

2.3 Netzwerk-Dateisysteme

Innerhalb eines Rechnerpools ist die Verwendung von Netzwerk-Dateisystemen zum Zugriff auf eine gemeinsame Benutzerdatenbank zwar prinzipiell möglich, hat aber auch einige Nachteile. So muss der verteilte Zugriff auf die Datenbankdateien (*/etc/passwd*, */etc/shadow*, */etc/group*, ...) synchronisierbar sein, um inkonsistente Dateizustände durch konkurrierende Schreib- oder Lesezugriffe zu vermeiden. Die Kommunikation zwischen Client-Rechner und Dateisystem-Server sollte verschlüsselt erfolgen, um die Sicherheit und Integrität der abgelegten Daten, wie z.B. der verschlüsselten Passwörter, zu garantieren. Client und Server müssen sich dazu außerdem gegenseitig authentifizieren. Nicht alle Netzwerk-Dateisysteme werden diesen Anforderungen gerecht. Ein Beispiel dafür ist NFS (Network File System), welches sich einige Sicherheitsdefizite mit dem ursprünglichen NIS teilt, siehe dazu zum Beispiel in [SEL01].

Die Wahl des entsprechenden Netzwerk-Dateisystems sollte also wohl überlegt sein. Der entscheidende Nachteil eines solchen Lösungsansatzes ist aber das Fehlen einer lokalen Instanz der Benutzerdatenbank, so dass lokal existierende Benutzerkonten nicht mehr realisierbar sind.

2.4 Plugable Authentication Module (PAM)

Im Jahre 1995 wurde PAM von der Firma Sun Microsystems entwickelt und hat sich mittlerweile auf vielen Unix-ähnlichen Betriebssystemen, wie etwa Solaris oder Linux, als Standard zur Benutzerauthentifizierung durchgesetzt. Es handelt sich bei PAM um eine Programmbibliothek, die als Schnittstelle zwischen den Anwendungsprogrammen und unterschiedlichen Authentifizierungsmodulen dient. Auch diese sind ihrerseits wieder Programmbibliotheken. Dadurch kann die Entwicklung von Anwendungsprogrammen und Authentifizierungsmodulen getrennt erfolgen. Beide müssen lediglich die Programmierschnittstellen einhalten. Der Systemadministrator entscheidet nach Bedarf, welche Authentifizierungsmethode für welche Anwendung benutzt wird, ohne dass dafür die PAM-Bibliothek oder sogar eines der Anwendungsprogramme neu übersetzt werden müssten. Dabei wird auf die Standard-Programmbibliothek zurückgegriffen, so dass über Mechanismen wie NSS (Name Service Switch) auch die Verwendung von NIS weiter möglich bleibt. Daneben existiert aber auch ein Modul, welches NIS direkt unterstützt.

Es ist wichtig, dass die Anwendungsprogramme (z.B. */bin/login*, */bin/su*) PAM konform sind, da andernfalls die klassischen Authentifizierungsmechanismen zum Einsatz kommen.

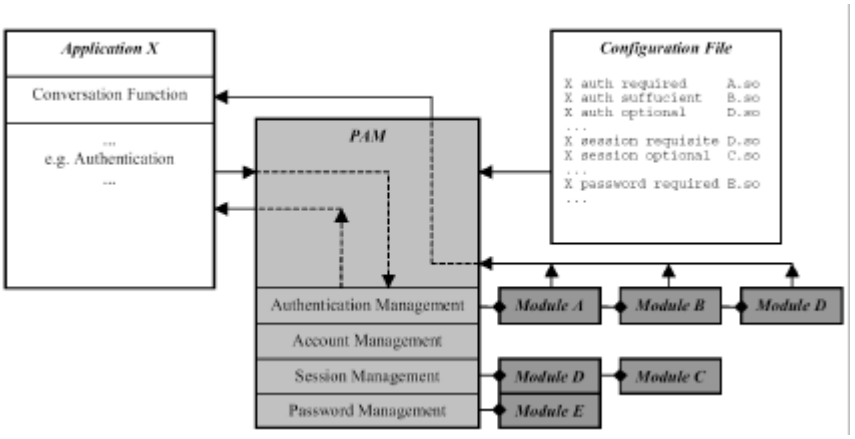

Abbildung 1: Struktur eines PAM Moduls

Jedes PAM-Modul stellt dabei mindestens einen der folgenden vier Dienste zur Verfügung:

• *Authentication Management*: Authentifizierung des Benutzers

• *Account Management*: Verwaltung und Kontrolle des Benutzerkontos. Hier wird beispielsweise das Gültigkeitsdatum eines Passwortes überprüft, und gegebenenfalls zur Eingabe eines neuen aufgefordert.

• *Session Management*: Verwaltung der aktuellen Arbeitssitzung. Dieser Dienst wird am Anfang und am Ende einer jeden Arbeitssitzung gestartet und kann beispielsweise dazu dienen, Anmeldevorgänge zu protokollieren oder benutzerspezifische Netzwerk-Dateisysteme einzuhängen.

• *Password Management*: Verwaltung des Authentifizierungsmerkmals. Dieser Dienst wird genutzt, um das hinterlegte Authentifizierungsmerkmal (z.B. Passwort) eines Benutzers zu erneuern.

2.5 Single Sign-On (SSO)

Unter Single Sign-On versteht man die Möglichkeit eines Benutzers, durch einmalige Authentifizierung am Anfang einer Arbeitssitzung, für deren gesamten Verlauf alle den jeweiligen Zugriffsrechten entsprechenden Systeme und Ressourcen benutzen zu dürfen [XO97]. Selbst der Zugriff auf andere Computer, wie etwa eine entfernte Anmeldung oder der Zugriff auf ein E-Mail-Postfach, erfordert keine erneute Benutzerauthentifizierung. Der Entwurf und die Umsetzung eines solchen Systems stellt eine anspruchsvolle Aufgabe dar, und konnte bisher in der Praxis nur bedingt realisiert werden.

Single Sign-On Verfahren weise dabei sowohl Vor- als auch Nachteile auf:

• *Benutzungskomfort*: Der Benutzer muss sich im Verlauf einer Arbeitssitzung nur einmal authentifizieren. Der wiederholte Umgang mit möglicherweise sogar unterschiedlichen Benutzernamen und Authentifizierungsmerkmalen, beim Zugriff auf verteile Systeme, entfällt.

• *Zeitersparnis*: Benutzerauthentifizierung kostet Zeit, da der Benutzer dazu mit dem Computersystem interagieren muss, und dies im Allgemeinen relativ langsam geschieht. Es können Eingabefehler auftreten, die eine Wiederholung des Vorganges erfordern. Ohne eine direkte Benutzerauthentifizierung wird der Zugriff auf Ressourcen beschleunigt. Auch für die Arbeit der Systemadministratoren kann dies von Vorteil sein, da sich die Verwaltung von Benutzerkonten und Zugriffsrechten unter Umständen vereinfacht.

• *Sicherheit*: Durch einen selteneren Umgang mit den Authentifizierungsinformationen verringert sich auch die Gefahr, dass diese durch Ausspähen in falsche Hände geraten. In der Regel kommen kryptographische Verfahren bei der indirekten Benutzerauthentifizierung zwischen den beteiligten Computersystemen zum Einsatz. Selbst verteilte Anwendungen, die ihrerseits unverschlüsselt kommunizieren, können so die Geheimhaltung der Authentifizierungsinformationen nicht mehr gefährden.

Ferner ist eine Applikations- oder Plattform-spezifische Skalierung des Sicherheitsniveaus mittels Single Sign-On nicht erreichbar, da der unberechtigte Zugriff auf ein solches System als besonders problematisch eingeschätzt werden muss. Gelingt es einem Angreifer beispielsweise, sich einmal unter einem falschen Benutzernamen erfolgreich anzumelden, oder verlässt ein Benutzer den Arbeitsplatz ohne seinen Rechner vor Fremdbenutzung zu schützen, so hat der unberechtigte Benutzer vollen Zugriff auf alle Ressourcen der berechtigten Person.

2.6 Fazit

Die klassische Benutzerauthentifizierung verwendet meist Passwörter, um den Zugriff auf Rechner und deren Systemressourcen zu beschränken. Im Laufe der Zeit wurden verschiedene Techniken entwickelt, um jenes Wissen über eine definierte Buchstabenkombination, auch der Benutzerauthentifizierung innerhalb ganzer Rechnerpools verfügbar zu machen, und dies gleichzeitig mit einer zentralen Administration der Benutzerkonten zu verbinden.

Durch Single Sign-On (siehe Abschnitt 2.5) kann primär der Komfort beim Einsatz solcher Systeme gesteigert werden. Auf Unix-Betriebssystemen ist Kerberos (siehe [KPS02], [Ke04] und [He04]) das bisher einzige Single Sign-On System, welches sich auch in der Praxis durchsetzen konnte. Kerberos benutzt Passwörter zur symmetrischen Verschlüsselung. Die Verwendung anderer Authentifizierungsmerkmale ist prinzipiell möglich, sofern sich von ihnen eindeutige Schlüssel ableiten lassen. Einige biometrische Merkmale erfüllen diese Bedingung jedoch noch nicht ausreichend. Die Verwendung austauschbarer Authentifizierungsmodule ermöglicht eine getrennte Entwicklung von Authentifizierungsmethoden und Anwendungen. PAM (siehe Abschnitt 2.4) ist das zurzeit am weitesten verbreitete und meist akzeptierte System dieser Art. Auch einige kommerzielle Anbieter von Lösungen zur biometrischen Benutzerauthentifizierung greifen darauf zurück. Deren großer Nachteil ist jedoch, dass in der Regel keine entfernte Benutzerauthentifizierung auf Basis biometrischer Verfahren möglich ist, da stets die lokalen Sensoren verwendet werden.

Es fehlen Single Sign-On Systeme, welche PAM erfolgreich einsetzen. Die meisten entsprechenden Lösungsansätze (z.B. Kerberos) basieren darauf, dass eine Client-Anwendung durch Übermittlung von indirekten Authentifizierungsmerkmalen die Authentifizierung anstelle des Benutzers durchführt. Die dabei verwendeten Anwendungen (Client und Server) sind an spezifische Kommunikationsprotokolle angepasst. Dagegen wurden PAM-Anwendungen auf eine direkte, aber universelle Interaktion mit dem Benutzer ausgelegt. Die eigentlichen Authentifizierungsmodule eines PAM-Systems haben keine Möglichkeit, direkt mit einer entfernten Client-Anwendung zu kommunizieren. Entsprechend existiert zwar ein PAM-Modul, welches eine Benutzerauthentifizierung auf Basis eines Kerberos-Servers realisiert, aber Single Sign-On wird nicht unterstützt. Es muss stets das Benutzerpasswort eingegeben werden.

Das im Rahmen unserer Arbeit entwickelte System basiert auf PAM und erlaubt dennoch eine entfernte Benutzerauthentifizierung auf Basis biometrischer Merkmale [Hl04]. Die individuelle Wahl der dabei verwendeten Authentifizierungsmethoden (Besitz, Wissen oder Sein) muss auf den jeweiligen Benutzer, dessen hinterlegte Merkmale sowie den benutzten Arbeitsplatzrechner abgestimmt sein. Durch dieses Vorgehen lässt sich trotz heterogener Rechnerkonfigurationen ein einheitlich sehr hohes Sicherheitsniveau für den gesamten Rechnerpool realisieren. In den folgenden beiden Kapiteln wird das zugrunde liegende Konzept beschrieben und erste Testergebnisse präsentiert.

3 Konzept

In Abbildung 2 ist das von uns entwickelte System zur multifaktoriellen Benutzerauthentifizierung gezeigt. Es besteht aus einem zentralen Authentifizierungsdienst (Trust Center) und mehreren Client-Rechnern, die mit Hilfe eines PAM-Moduls (Conversation Module) auf diesen zentralen Dienst zurückgreifen, um ihre Benutzer zu authentifizieren. Jeder Client-Rechner bietet seinerseits einen Dienst (Sensor Daemon) an, der den Zugriff auf die Sensoren (z.B. Fingerabdruckscanner, Kamera) dieses Rechners gestattet.

Das PAM-Modul dient dabei lediglich zur Interaktion zwischen Trust Center und dem Benutzer. Die eigentliche Benutzerauthentifizierung wird durch das Trust Center realisiert. Eine Verwaltung der Benutzerkonten ist mit dem PAM-Modul nicht möglich, da die vorgegebene Programmierschnittstelle nicht ausreicht, um die notwendigen, relativ komplexen Interaktionen zu realisieren. Deshalb wird hierfür ein eigenes Anwendungsprogramm (Admin Tool, nicht in Abbildung 2 gezeigt) bereitgestellt. Der Zugriff auf die jeweiligen Dienste, sowie die Kommunikationskanäle zwischen den Modulen werden durch kryptographische Verfahren (angelehnt an SSL, Details siehe [Hl04]) abgesichert.

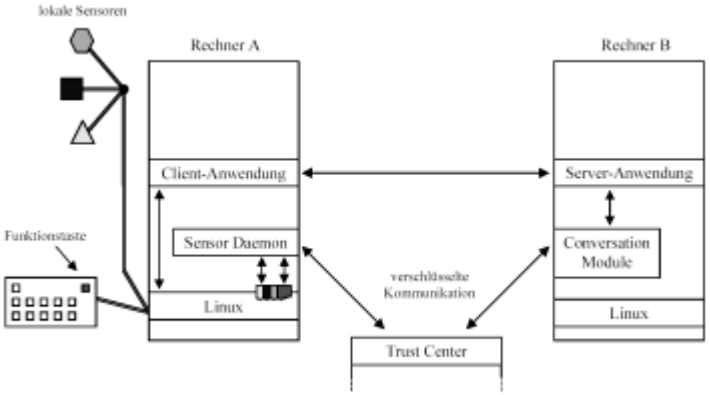

Abbildung 2: Architektur unseres PAM-basierten Authentifizierungssystems

Im endgültigen Beitrag folgen an dieser Stelle eine formale Beschreibung des von uns entwickelten Protokolls, sowie ein Beispiel für den Datenfluss in dem Szenario eines Rechner-übergreifenden Authentifizierungsvorgangs.

Im Allgemeinen sollte es nur dem Trust Center erlaubt sein, mit Hilfe des jeweiligen Sensor Daemons auf die Sensoren eines Rechners zuzugreifen. Nur die privilegierten Anwendungen der Client-Rechner (Conversation Module und Admin Tool) haben Zugang zum Trust Center. Die Bestimmung des physikalischen Ursprunges einer Arbeitssitzung erfolgt mit Hilfe eines von uns definierten und so genannten *Watchwords* (engl. Kennwort, Parole, Losung, [Hl04]).

Ein *Watchword* dient dazu, das entsprechende PAM-Modul eines Servers darüber zu informieren, von welcher Rechnereinheit die Authentifizierungsinformationen eines Benutzers bezogen werden können. Es besteht deshalb aus dem Namen des Ursprungsrechners bzw. dessen Netzwerkadresse, einer TCP Portnummer und einem zufälligen Einmal-Passwort. (siehe Abbildung 3).

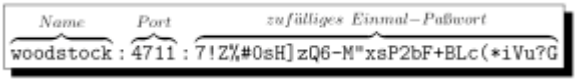

Abbildung 3: Beispiel eines *Watchword*s

Die zu Grunde liegende Idee ist, dass der Benutzer diese Eingabe nur mit Hilfe der Tastatur (Betätigung einer vordefinierten Funktionstaste) an seinem lokalen Arbeitsplatzrechner initiieren kann und dies durch einen modifizierten Tastaturtreiber im jeweils lokalen Betriebssystem realisiert wird, welcher alle Tastatureingaben des lokalen Rechners überwacht und ggf. bei Druck der entsprechenden Taste das angeforderte *Watchword* generiert und in die Tastatureingabe einschleust. Seine Gültigkeitsdauer ist begrenzt und es ist zu einem Zeitpunkt für jeden Arbeitsplatz nur ein *Watchword* gültig. Wird also die Übertragung eines neuen *Watchword*s ausgelöst, so verlieren damit alle vorherigen ihre Gültigkeit und wurde ein *Watchword* bereits einmal benutzt, so wird es danach ungültig.

Im Falle einer lokalen Benutzerauthentifizierung ist keine Eingabe des *Watchword*s notwendig, da das PAM-Modul diesen Sonderfall erkennt und dem Trust Center die reale Position des Benutzers mitteilt. Ist der Ursprung eines Authentifizierungsversuches jedoch ungewiss, so bittet das Trust Center den Benutzer mittels PAM-Modul um die Eingabe des *Watchword*s.

Nachdem dem System mittels *Watchword* bekannt ist, von welchem Rechner aus die Benutzerauthentifizierung angefordert wird, kann das Trust Center anhand der eingestellten Systemkonfiguration bestimmen, ob die Authentifizierung mittels Biometrie (Sein), Besitz oder Wissen erfolgt. Das Trust Center agiert somit in diesem Szenario als eine zentrale Authentifizierungsautorität und beherbergt zudem personenbezogene Daten der Benutzer. Ein unberechtigter Zugriff auf diese Informationen gefährdet die Sicherheit aller angeschlossenen Client-Rechner. Aus diesem Grund muss das Trust Center stets auf einem physisch sicheren Knoten des Rechnernetzes ausgeführt werden, dessen Hardware nur dem Systemadministrator zugänglich ist.

Im derzeitigen Stadium ist das von uns entworfene Demonstrationssystem in der Lage, wissensbasierte sowie biometrische Authentifizierung, basierend auf Fingerabdruckverfahren, zu unterstützen. Dabei kann die gewählte Authentifizierungsmethode Benutzer-, Rechner- und Dienst-abhängig in der Trust Center Konfiguration eingestellt werden. Weiterhin sind Kombinationen von Wissens- und Sein-basierten Authentifizierungsmodalitäten möglich.

Das von uns implementierte Fingerabdruck-Verifikationsverfahren stellt eine Beispielimplementierung dar und basiert auf Minutienbeschreibungen als gängige Merkmale, kombiniert mit einem neuartigen, iterativen Vergleichsverfahren basierend auf Polarkoordinaten. Details zur Implementierung des biometrischen Verfahrens finden sich in [Hl04].

4 Testergebnisse

Zur Beurteilung des von uns entwickelten Authentifizierungssystems sind Tests durchgeführt worden. Zum einen waren dies ein manueller Funktionstest, der unter Laborbedingungen mit Hilfe einer speziellen Test-Infrastruktur realisiert wurde und zum anderen waren dies erste statistische Untersuchungen des entwickelten Fingerabdruckverfahrens, um eine Vergleichbarkeit der Erkennungsgenauigkeit mit anderen Authentifizierungsverfahren zu ermöglichen. An dieser Stelle erfolgt eine Zusammenfassung der jeweiligen Erkenntnisse.

4.1 Funktionstest

Mit Hilfe des Funktionstestes soll insbesondere die generelle Realisierbarkeit einer entfernten Benutzerauthentifizierung auf Basis lokaler Sensoren und Eingabegeräte nachgewiesen werden. Die dazu von uns verwendete Infrastruktur besteht aus einem Pool von drei Rechnern. Zwei von ihnen (Snoopy und Woodstock) benutzen das Betriebssystem Linux (Kernel-Version 2.4.21). Auf dem dritten Rechner ist Microsoft Windows installiert.

Die Rechner sind über ein gemeinsames Ethernet an einen Router angeschlossen, der allen Knoten des lokalen Subnetzes den Zugriff auf das Internet ermöglicht. Alle aus dem Internet eingegangenen Kommunikationsanforderungen werden an den Rechner Woodstock weitergeleitet, wie in der in Abbildung 4 aufgezeigten Infrastruktur. Im Rahmen der Tests wurden alle spezifizierten Funktionen erfüllt, insbesondere war es nach einer lokalen Anmeldung auf dem Rechner Snoopy möglich, den dort lokal angeschlossenen Fingerabdruck-Scanner zur entfernten Benutzerauthentifizierung gegenüber dem Rechner Woodstock einzusetzen. Dies funktionierte selbst dann, wenn die Kommunikationsverbindung bereits über mehrere andere Rechner getunnelt wurde.

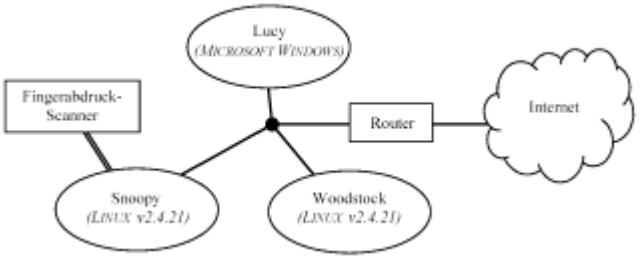

Abbildung 4: Aufbau des Funktionstests

Im endgültigen Beitrag wird an dieser Stelle ein erweiterter Erfahrungsbericht unserer Funktionstests stehen, mit einer Zusammenfassung, auf welchen Linux Systemen das System heute bereits einsetzbar ist und welche zukünftigen Tests, z.B. in sehr großen Infrastrukturen, noch geplant sind.

4.2 Bewertung des biometrischen Fingerabdruckverfahrens

Zur Einschätzung der Erkennungsgenauigkeit wurde von uns die statistische Methode der Fehlerratenermittlung angewendet, bei der über Falsch-Akzeptanz- (False-Acceptance-Rate, FAR) sowie der Falsch-Rückweisungsraten (False-Rejection-Rate, FRR) die Güte des biometrischen Verfahrens prinzipiell abschätzbar wird [Ho98]. Die Tests sind als eine erste Evaluierung zu verstehen und umfassen Testdatensätze mit jeweils 10 Aufnahmen von insgesamt 28 unterschiedlichen Fingern freiwilliger Probanden. Die Datensätze wurden personenweise jeweils in einen Test- und einen Referenzdatensatz mit jeweils 5 Abbildungen pro Finger aufgeteilt, damit waren insgesamt 700 Falsch-Rückweisungstests und 18900 Falsch-Akzeptanztests möglich. Die Testdatensatzgrösse erlaubt zwar keine statistisch signifikanten Schlüsse, zeigt jedoch dass für unseren Zweck der Evaluation der Infrastruktur das Verfahren grundsätzlich einsetzbar ist.

Das im Rahmen unserer Arbeiten entwickelte Fingerabdruckvergleichsverfahren weist im Wesentlichen vier Verfahrensparameter auf [Hl04], für welche wir im Rahmen der Tests einige repräsentative Arbeitspunkte untersucht haben:

- minimalen Anzahl zu übereinstimmender Minutien („min“)
- maximal erlaubte Positionsabweichung dargestellt in der Einheit Millimeter („pos“)
- maximal erlaubte Abweichung in der Minutien-Orientierung in Grad („ori“)
- maximales Höhen/Breiten-Verhältnis der erlaubten Positionsabweichung („width“)

Einige signifikante Erkennungsraten, die in unseren Tests ermittelt wurden, sind in Tabelle 1 abgebildet. Die Spalten „1:1“ bzw. „1:5“ unterscheiden dabei darin, ob in den Versuchen der Vergleich einer aktuellen Eingabe zu jeweils nur einem oder allen fünf verschiedenen Referenzabdrücken erfolgte. Zu erkennen ist, dass insbesondere die Werte in der zweiten Zeile und den Spalten „1:5“ einen guten Kompromiss zwischen Benutzerkomfort (FRR=0,71%, d.h. Falschablehnung im Durchschnitt in weniger als 1% aller Fälle) und Sicherheitsniveau (FAR=0,07%, d.h. geringe Wahrscheinlichkeit einer Falschakzeptanz) darstellen und die Einsetzbarkeit des Verfahrens nahe legen. Bei der Interpretation dieser Ergebnisse ist jedoch zu beachten, dass die Tabelle 1 dargestellte Parametrisierung des von uns entwickelten Verfahrens hinsichtlich der zur Verfügung stehenden Testdaten optimiert wurde. Die Überprüfung, zu welchem Grade sich auch bei einem breiteren Datenspektrum ähnliche Erkennungsgenauigkeiten ergeben, steht im Rahmen zukünftiger Arbeiten noch aus.

					1:1		1:5	
Nr.	min	pos [mm]	ori[°]	width[%]	FRR[%]	FAR[%]	FRR[%]	FAR[%]
1	3	0.05	5	27	16.42	0.07	0.71	0.5
2	4	0.1	6.5	30	13.57	0.02	0.71	0.07
3	5	0.2	7	37	11.41	0.05	0.71	0.34

Tabelle 1: False-Rejection (FRR) und False-Acceptance Raten (FAR) des Fingerabdruckverfahrens

5 Zusammenfassung und Ausblick

In diesem Artikel wurde ein neuartiges Verfahren zur Benutzerauthentifizierung in vernetzten Infrastrukturen basierend auf Linux Betriebssystemen vorgestellt und die Funktionsfähigkeit einer prototypischen Implementierung anhand erster Testergebnisse plausibel gemacht. Das vorgestellte Verfahren stellt einen ersten Versuch dar, eine multifaktorielle Benutzerauthentifizierung über PAM zu realisieren. Es zeichnet sich gegenüber alternativen Ansätzen wie Single Sign-On u.a. dadurch aus, dass der Authentifizierungsmechanismus spezifisch hinsichtlich Benutzern, Rechnern und Diensten skalierbar ist und das System dabei – abhängig von der Ausstattung der Geräte mit biometrischen Sensoren – entweder wissensbasierte oder biometrische Benutzerauthentifizierung, als auch eine Kombination davon, durchführen kann. Zudem ist die aufgezeigte Implementierung kompatibel zum Linux-Standardmechanismus zur Benutzerauthentifikation, PAM. Als biometrisches Referenzverfahren ist in dem System ein Fingerabdruckvergleich implementiert und die Testergebnisse motivieren eine Weiterentwicklung des Ansatzes in ein funktionales System.

Zukünftige Arbeiten werden im Wesentlichen auf drei Ziele ausgerichtet sein; zum einen wird derzeit die Möglichkeit geprüft das Verfahren zwecks Weiterentwicklung als Open Source Software Projekt freizugeben. Weiterhin muss die Genauigkeit des spezifischen Fingerabdruckverfahrens in umfangreicheren Tests untermauert werden. Zudem ist beabsichtigt, das Systems künftig um den Mechanismus der besitzbasierten Authentifizierung (z.B. durch Smartcards) zu erweitern.

Literaturverzeichnis

[An01] Anonymous (rev. by Ray, J.): Maximum Linux Security. Sams Publishing, 2001,
[Bi03] Bishop, M.: Computer Security. Addison-Wesley, Boston, U.S.A, 2003
[FFIS04] Free Fingerprint Imaging Software. http://ffpis.sourceforge.net, abgefragt im Juni 2004
[FVS04] Fingerprint Verification System. http://fvs.sourceforge.net, abgefragt im Juni 2004
[GS96] Garfinkel, S.; Spafford, G.: Practical Unix & Internet Security. O'REILLY, 1996
[He99] Herold H.: Linux-Unix-Systemprogrammierung. Addison-Wesley, 1999
[He04] Heimdal. http://www.pdc.kth.se/heimdal/, abgefragt im Juni 2004

[Hl04] Helmholz, C.: Multifaktorielle Benutzerauthentifizierung mit Fingerabdruck in Linux-Infrastrukturen. Diplomarbeit an der Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, 2004

[Ho98] Hong, L.: Automatic Personal Identification Using Fingerprints. Dissertation submitted to Michigan State University, 1998

[Ke04] Kerberos: The Network Authentication Protocol. http://web.mit.edu/kerberos/, abgefragt im Juni 2004

[KPS02] Kaufman, C.; Perlman, R.; Speciner, M.: Network Security, Prentice Hall, 2002

[MMJP03] Maltoni, D.; Maio, D.; Jain, A.K.; Prabhakar, S.: Handbook of Fingerprint Recognition, Springer, New York, U.S.A., ISBN 0-387-95431-7, 2003

[MRLW01] Monrose, F.; Reiter, M.K.; Li, Q.; Wetzel, S.: Using voice to generate cryptographic keys. Proceedings of Odyssey 2001, In: Proceedings of the Speaker Verification Workshop, 2001

[Pr01] Prabhakar, S.: Fingerprint Classification and Matching Using a Filterbank. Dissertation submitted to Michigan State University, 2001

[SEL01] Stern, H.; Eisler, M.; Labiaga, R.: Managing NFS and NIS. 2nd Edition, O'Reilly, 2001

[UPPJ04] Uludag, U.; Pankanti, S.; Prabhakar, S.; Jain, A.K.: Biometric Cryptosystems: Issues and Challenges, In: Kundur, D.; Lin, C.-Y.; Macq, B.; Yu, H.: (Eds.), Proceedings of the E-EE, Special Issue on Enabling Security Technology for Digital Rights Management, Vol. 92, No. 6, 2004

[Ve02] Vec, M.: Die Spur des Täters. Methoden der Identifikation in der Kriminalistik. (1879-1933), Nomos Verlag, 2002

[VSM02] Vielhauer, C.; Steinmetz, R.; Mayerhöfer, A.: Biometric Hash based on Statistical Features of Online Signatures. In: Proceedings of the IEEE International Conference on Pattern Recognition (ICPR), Quebec City, Canada, Vol. 1, 2002

[XO97] X/Open Single Sign-On Service – Pluggable Authentication Modules. The Open Group, 1997