Proseminar Speicher- und Dateisysteme

Wintersemester 2018/2019

Betreuer: Dr. Michael Kuhn, Jannek Squar

Access Control

Bjarne Meimerstorf | 6809565

B.Sc. Wirtschaftsinformatik

Gliederung

1. Access Control allgemein

Access Control steht grundsätzlich für die Zugangskontrolle auf Ressourcen. Dabei gibt es unterschiedliche Formen von Access Control. Die für die meisten intuitivste Art ist physische Access Control, die Zugangskontrolle auf physisch vorhandene Ressourcen. Beispiele dafür sind abgeschlossene Türen, Flughafengates oder Grenzkontrollen.

Der Prozess Access Control besteht aus drei Vorgängen, der Authentisierung Authentifizierung und Autorisierung. Bei der Authentisierung wird überprüft, ob die angegebene Identität tatsächlich stimmt. Beim Flughafengate stellt der Abgleich des Ausweises mit dem Flugticket und dem Fluggast die Authentisierung dar. Die Authentifizierung ist der Schritt der Überprüfung von Zugriffsrechten, sie würde beim Flughafengate-Beispiel stattfinden, wenn das Flugticket mit der Datenbank der Fluggäste abgeglichen wird. Der letzte Schritt der Autorisierung ist die Freigabe der Ressource, im Beispiel also den Zugang zum Flugzeug.

Die Form von Access Control, auf der hier der Fokus liegen soll, ist technische bzw. logische Access Control. Sie beschäftigt sich mit Zugangskontrolle bei Softwaresystemen. Um Methoden von logischer Access Control zu kategorisieren, müssen drei Fragen beantwortet werden:

- Was ist die kleinste schützbare Einheit? Das wird als Granularität bezeichnet.

- Welche Operationen sollen auf den autorisierten Ressourcen ausgeführt werden können? Meistens

- Wie ist der Zugangsschritt konkret umgesetzt?

Beispiel 1: Ein Anmeldebildschirm gibt dem Benutzer mit Nutzerkennung und Passwort oder PIN (Zugang) Zugriff auf das gesamte Benutzerprofil mit allen eigenen Dateien (Granularität), die vom Benutzer gelesen, verändert, ausgeführt und gelöscht werden können (Operationen).

Beispiel 2: Die Windows Dateifreigabe ermöglicht es, Besitzern von Dateien anderen Nutzern Zugriffsrechte zu erteilen (Zugang & Granularität). Ob die Nutzer die Dateien lesen, verändern oder löschen können, wird vom Besitzer entschieden (Operationen)

2. Logische Access Control Modelle

Jedes logisches Access Control System basiert auf einem von vielen logischen Access Control Modellen. Im Folgenden werden einige der Modelle vorgestellt, um zu zeigen, wie vielfältig logische Access Control sein kann.

a) Mandatory Access Control

Auf Mandatory Access Control basierende Systeme haben Zugriffsrechte, die zentral von einem Administrator gesteuert werden. Dabei werden hierarchisch angeordnete Labels für Benutzer und Dateien verwendet. Somit kann ein Benutzer, mit einer höheren Sicherheitsstufe auf alle Ressourcen auf seiner Stufe und niedrigeren zugreifen. Es wird häufig in Regierungsorganisationen eingesetzt wie zum Beispiel beim System der Geheimhaltungsstufen in Deutschland oder der Classified Information der USA.

b) Role Based Access Control

Ein anderer Ansatz von logischer Access Control, bei der Zugriffsrechte zentral gesteuert werden, ist Role Based Access Control. Der Hauptunterschied zwischen den beiden Modellen besteht darin, dass die den Nutzern zugewiesenen Rollen bei Role Based Access Control nicht hierarchisch angeordnet sind. Jede Rolle kann somit nur auf die für sie vorgesehenen Ressourcen zugreifen.
Häufig werden diese beiden Modelle gleichzeitig in Access Control Systemen eingesetzt. In einem Unternehmen zum Beispiel können die unterschiedlichen Abteilungen durch Role Based Access Control und die hierarchische Struktur (zB Gruppenleitung, Abteilungsleitung, Geschäftsführung usw.) durch Mandatory Access Control abgebildet werden.

c) Discretionary Access Control

Das Modell von logischer Access Control, das den meisten bekannt sein wird, ist Discretionary Access Control. Die meisten Betriebssysteme wie Windows, Linux und macOS benutzen Access Control Methoden, die auf Discretionary Access Control basieren. Häufig wird eine Discretionary Access Control List benutzt, die alle Beziehungen von Benutzern und Dateien speichert, wobei jede Datei einen Eintrag mit den gewährten Zugriffsrechten pro Benutzer hat.

3. Access Control Implementierungen

a) Windows

Das Windows Access Control System besteht aus zwei Elementen, den Access Tokens und den Security Descriptors. Beide Elemente beinhalten relevante Informationen bezüglich Access Control, die Security Identifier, wobei die Access Tokens zu Nutzern und die Security Descriptors zu Objekten gehören. Der Security Descriptor eines Objekts verweist auf die in der Discretionary Access Control List festgelegten Zugangsrechte, um zu bestimmen, welche Operationen ein Nutzer an dem Objekt ausführen kann.

Ein Grund dafür, dass Windows lange für mangelhafte Sicherheit bekannt war, ist die Tatsache, dass Nutzer – und ihre ausgeführten Prozesse – standardmäßig Adminrechte hatten. Somit konnten auch Viren und Malware einfach Änderungen am System vornehmen.

Mit Windows Vista wurde allerdings eine Technologie eingeführt, die User Account Control heißt. Sie sollte dazu führen, dass Windows Benutzer zwar Prozesse mit Adminrechten ausführen können, aber eine zusätzliche Sicherheitsschicht dafür sorgt, dass solche Prozesse keine ungewollten Konsequenzen mit sich ziehen. UAC fragt den Nutzer jedes Mal, wenn Adminrechte gefordert werden, nach einer Bestätigung, teilweise muss dafür erneut das Passwort eingegeben werden. Dafür wird dem Benutzer ein Security Token für den angeforderten Prozess gewährt, mit dem Adminrechte verwendet werden dürfen. Dadurch können bösartige Programme nicht weiterhin ohne Wissen des Benutzers den gleichen Schaden anrichten wie vor UAC.

Aufgrund der Art, wie ältere Windows Programme geschrieben wurden, ist es allerdings schwierig, sie in neueren Systemen mit UAC problemlos auszuführen, da viele Programme mit dem Gedanken im Hinterkopf entwickelt wurden, immer Zugriff auf Adminrechte zu haben. Dafür existiert der Kompatibilitätsmodus und die Möglichkeit, Programme direkt als Administrator auszuführen.

Durch UAC wird die Prozessicherheit von Windows mittlerweile als gleichwertig mit der sudo Funktion aus Linux angesehen, welches es Standardusern jedes Mal nach Zugangsdaten fragt, wenn auf Prozesse, die Adminrechte nutzen, zugegriffen werden soll.

b) Linux

Linux hat eine etwas simplere Herangehensweise an Access Control als Windows. Für jedes Objekt ist ein dreistelliger Oktalwert gespeichert, der alle möglichen Beziehungen von Objekten und Nutzern darstellen kann. Diese Oktalwerte stellen wiederum jeweils eine dreistellige Binärzahl dar, die sich aus den drei Operationen lesen, schreiben und ausführen zusammenstellt. So steht eine 110 für die Berechtigung, eine Datei lesen und verändern, aber nicht ausführen zu dürfen. Die drei Oktalzahlen stellen somit die Zugangsberechtigungen für Besitzer einer Datei (erste Zahl), Nutzer in der Gruppe der Datei (zweite Zahl) und alle anderen (dritte Zahl) dar.

Mit dem Kommandozeilenprogramm chmod ist es möglich, Berechtigungen in Linux zu verändern. Das Schema davon funktioniert folgendermaßen: nach chmod wird zuerst entschieden, für wen sich die Zugangsberechtigungen verändern sollen. Die Zeichenkette ugoa steht in diesem Kontext für Besitzer, Gruppe, andere und alle. Danach wird mit einem +, = oder – entschieden, ob Rechte hinzugefügt, gleichgesetzt oder entfernt werden sollen. Anschließend werden durch r, w oder x die Zugangsart bestimmt.

Ein Beispiel:
akishore@ASEEMVOSTRO: ~ /Test$ ls -l
-rw-rw-rw- 1 akishore akishore 10240 Feb 8 1732 practice
akishore@ASEEMVOSTRO: ~ /Test$ chmod ug+x practice
akishore@ASEEMVOSTRO: ~ /Test$ ls -l
-rwxrwxrw- 1 akishore akishore 10240 Feb 8 1732 practice

Linux wird von manchen dafür kritisiert ein veraltetes Zugangsberechtigungssystem für Dateien zu benutzen. Das liegt daran, dass Access Control Lists zwar von neueren Versionen unterstützt werden, aber nicht wie bei Windows standardmäßig verwendet werden. Dazu kommt, dass die meisten Benutzer, die sich nicht in einem Arbeitsumfeld befinden, die Access Control List Unterstützung nicht ausnutzen.

4. Virtuelle Maschinen und Containerisierung

Sowohl Virtuelle Maschinen als auch Container sind Ansätze, um Anwendung in fremden Umgebungen ausführen zu können. Virtuelle Maschinen benutzen dafür Hypervisor, die auf dem Host-Betriebssystem oder direkt auf der Hardware separaten Gastsystemen Hardware-Ressourcen erteilen. So ist es zum Beispiel möglich, Linux auf Windows auszuführen, ohne dass eines der beiden Betriebssysteme etwas von der Existent des anderen weiß. Container erstellen hingegen kein weiteres System, sondern führen mithilfe der Container Engine direkt Anwendungen aus, die auf dem Betriebssystem nicht lauffähig wären.

Dabei teilen sich die Anwendungen auf Containern, anders als bei virtuellen Maschinen, Daten mit dem Host-Betriebssystem und untereinander. Das führt zu einer leichtgewichtigen Lösung, die jedoch nicht die Sicherheitsvorteile eines separaten Systems beinhaltet. Deswegen wird davon abgeraten, Container als direkte Ersatzlösung von virtuellen Maschinen anzusehen.

Doch auch die Sicherheit von virtuellen Maschinen wird gelegentlich durchbrochen, ein solcher Vorgang wird „Virtual Machine Escape“ genannt. Dabei gelingt es einem Prozess, der auf dem Gastbetriebssystem ausgeführt wird, Zugriff auf das Host-Betriebssystem oder auf andere Gastbetriebssysteme zu bekommen und potenziell Schaden anzurichten. Dies geschieht, indem versucht wird, den Hypervisor zu kontrollieren, um vollständigen Zugriff auf Hardware- und somit auch Software-Ressourcen zu erhalten.

Trotz solcher Bedenken sind virtuelle Maschinen grundsätzlich sehr sicher. Direkt auf den Hypervisor einzugreifen ist nicht einfach, in vielen Fällen sehr viel schwieriger als andere potenziell vorhandenen Sicherheitslücken in einem Computersystem auszunutzen. Ein Beispiel davon ist Admin Escape. Bei Admin Escape wird die Sicherheit von Systemen durchbrochen, die Zugriff auf administrative Tools besitzen. Viele Firmen geben zu vielen Computern Rechte auf diese Tools, was dazu führt, dass ein beeinträchtigtes System schwerwiegende Konsequenzen für das gesamte Netzwerk haben kann. Auf diese Systeme direkt zuzugreifen und mit ihren Privilegien zu arbeiten ist somit für viele weitaus einfacher als Virtual Machine Escape zu versuchen.

5. Zusammenfassung

Es wurde gezeigt, wie vielfältig das Thema Access Control sein kann und wie viele unterschiedliche Grundmodelle, Methoden, und Problemfälle existieren.
Gerade in einer Welt, in der die Verbreitung von vernetzter Technologie durch Themen wie Internet of Things und künstlicher Intelligenz stark zunimmt, bleibt Access Control relevant. Auf der einen Seite gibt es gute Gründe dafür, optimistisch zu sein. Viele Systeme wie neue Versionen von Linux und Windows unterstützten leistungsfähige Sicherheitstools oder benutzten sie sogar standardmäßig. Dazu kommen Angebote wie virtuelle Maschinen, die größtenteils nach wie vor solide Sicherheit anbieten. Auf der anderen Seite wird aber nicht immer verantwortungsvoll genug mit Sicherheitsthemen umgegangen. Zu häufig geben sich Nutzer – auch im professionellen Umfeld – zu viele Privilegien und gefährden damit die Sicherheit von sich selbst und vielen anderen. Unabhängig davon aber, in welche Richtung sich Themen rund um Access Control entwickeln, wird es interessant bleiben.

Quellen

- https://www.searchdatacenter.de/definition/Virtuelle-Maschine-Virtual-Machine-VM
- https://wiki.archlinux.org/index.php/File_permissions_and_attributes
- https://en.wikibooks.org/wiki/Fundamentals_of_Information_Systems_Security/Access_Control_Systems#Technical
- https://docs.microsoft.com/de-de/windows/desktop/SecAuthZ/security-identifiers
- https://docs.microsoft.com/de-de/windows/desktop/SecAuthZ/access-tokens
- https://docs.microsoft.com/en-us/windows/desktop/secauthz/dacls-and-aces
- https://docs.microsoft.com/en-us/windows/desktop/secauthz/access-control
- https://docs.microsoft.com/de-de/windows/desktop/SecAuthZ/security-descriptors
- https://www.tenouk.com/ModuleH2.html
- https://compas.cs.stonybrook.edu/~nhonarmand/courses/fa14/cse506.2/slides/ACLs-Vasu_and_Yaohui.pdf
- https://neuvector.com/container-security/containers-vs-virtual-machines-vms/
- https://www.datenschutzbeauftragter-info.de/authentisierung-authentifizierung-und-autorisierung/
- https://www.geeksforgeeks.org/access-control-listsacl-linux/
- https://www.techotopia.com/index.php/Mandatory,_Discretionary,_Role_and_Rule_Based_Access_Control
- https://entwickler.de/leseproben/containerisierung-der-it-579775782.html
- https://www.ca.com/de/blog-automation/was-ist-containerisierung-und-bedeutet-sie-das-ende-der-virtualisierung.html
- https://wr.informatik.uni-hamburg.de/_media/teaching/wintersemester_2015_2016/nthr-1516-berreis-virtualization_and_containerization-ausarbeitung.pdf
- https://de.wikipedia.org/wiki/Access_Control_List
- https://de.wikipedia.org/wiki/Role_Based_Access_Control

- https://de.wikipedia.org/wiki/Mandatory_Access_Control
- https://de.wikipedia.org/wiki/Discretionary_Access_Control
- https://de.wikipedia.org/wiki/Zugriffskontrolle
- https://de.wikipedia.org/wiki/Hypervisor
- https://slideplayer.com/slide/7028404/
- https://www.heise.de/newsticker/meldung/Hacker-brechen-aus-virtueller-Maschine-aus-3658416.html
- https://blog.docker.com/2016/03/containers-are-not-vms/
- https://www.networkworld.com/article/2230977/comparing-access-control-in-windows-and-linux.html
- https://searchenterprisedesktop.techtarget.com/Why-User-Account-Control-in-Windows-is-necessary
- https://wiki.ubuntuusers.de/sudo/
- https://www.researchgate.net/publication/255791810_virtual_machine_escapes
- https://www.astroarch.com/tvp_strategy/vm-escape-is-not-your-main-worry-39578/