IT-Sicherheitsmanagement Seminar: Group-Centric Models for Secure Information Sharing (g-SIS)

Uwe Kühn, Tobias Pöppke

1 Einleitung

TODO: Subscription Service, Meeting room Metaphern. [KSNW09] [SKNW10]

2 Grundlagen

Policy-, Enforcement-Layer.

2.1 Linear Temporal Logic

2.2 Lattice-Based Access Control

Eine wichtige Klasse von Zugriffskontrollmodellen sind *Lattice-Based Access Control (LBAC)* Modelle. Die Grundlagen von LBAC wurden in den 1970ern von Bell, LaPadula, Biba und Denning gelegt und entstanden aus den Bedürfnissen des Verteidigungssektors. Eine Übersicht über die verschiedenen Modelle und formale Definitionen, sowie weiterführende Referenzen finden sich bei Sandhu [Sand93].

LBAC Modelle konzentrieren sich auf die Betrachtung des Informationsflusses zwischen Sicherheitsklassen eines Systems. Dazu wird jedem *Objekt* des Systems eine *Sicherheitsklasse* zugewiesen. Ein *Objekt* sei hier informell definiert als ein Container für Informationen. Wenn nun die Informationen eines Objekts x zu einem Objekt y fließen geht damit ein Informationsfluss einher. Dieser Informationsfluss findet zwischen den Sicherheitsklassen der jeweiligen Objekte statt.

Die *Richtlinien* mit denen die Informationsflüsse beschrieben werden, können mithilfe von *Lattices* beschrieben werden. Ein *Lattice* ist in diesem Zusammenhang die partiell geordnete Menge der Sicherheitsklassen. Im Modell von Bell und LaPadula werden beispielsweise die Objekte ihren Sicherheitsklassen zugewiesen. Benutzer könne dann *Subjekte*, wie zum Beispiel einen Prozess, erstellen, die eine Sicherheitsklasse besitzen, die von der Sicherheitsklasse des Benutzers im Lattice dominiert wird. Ein Subjekt kann genau dann Objekte lesen, wenn die Sicherheitsklasse des Objekts von der des Subjekts dominiert wird. Auf ein Objekt schreibend zugreifen kann ein Subjekt genau dann, wenn die Sicherheitsklasse des Objekts die Sicherheitsklasse des Subjekts dominiert.

2.3 Domain and Type Enforcement

Die Idee des *Domain and Type Enforcement (DTE)*, von Badger et al. vorgeschlagen [BSSW+95], ist eine Erweiterung der Lattice-Based Access Control um *Domänen* und *Typen*. Eine *Domäne*

enthält Subjekte und Objekte werden zu *Typen* zugeordnet. Um den Informationsfluss zu kontrollieren, wird eine Matrix benutzt, in der die Lese- und Schreibrechte für jede Kombination von Domäne und Typ festgelegt werden. Diese Art der Zugriffskontrolle ist beispielsweise dann anwendbar, wenn es gewünscht ist, vertrauenswürdige Pipelines zu etablieren. Wenn zum Beispiel gewünscht wird, dass Informationen nur über ein Subjekt einer dritten Domäne zum Zielobjekt fließen dürfen, ist dies mit klassischen LBAC-Modellen nicht möglich. Dies liegt an der transitiven Natur der zugrundeliegenden Relation in LBAC.

Da die im Lattice höher angesiedelten Sicherheitsklassen in LBAC alle unter ihnen liegenden Sicherheitsklassen dominieren und diese Relation transitiv ist, kann Information immer von allen dominierten Sicherheitsklassen direkt in die dominierende Sicherheitsklasse fließen. Da bei DTE eine Matrix den zulässigen Informationsfluss definiert, ist diese Relation nicht länger transitiv und die Informationen können nach Wunsch über verschiedene Wege fließen.

2.4 Role-Based Access Control

Role-Based Access Control (RBAC) ist ein Zugriffskontrollmodel, dass von Sandhu et al. und Ferraiolo et al. vorgeschlagen wurde [SCFY96, FSGK$^+$01]. Ein Benutzer sei im Folgenden, ohne Beschränkung der Allgemeinheit, definiert als ein menschlicher Benutzer. Das Hauptmerkmal von RBAC sind *Rollen*. Eine Rolle basiert auf der Idee eine Position innerhalb einer Organisation darzustellen, die mit bestimmten Rechten und Verantwortlichkeiten einhergeht. Daher werden jeder Rolle bestimmte *Zugriffsrechte* auf *Objekte* zugewiesen, wobei ein *Objekt* Informationen bereitstellen oder erhalten kann. Ein *Zugriffsrecht* gewährt einer Rolle die Möglichkeit eine oder mehrere *Operationen* auf einem oder mehreren Objekten auszuführen.

Ein Benutzer kann in RBAC mehreren Rollen zugehören und eine Rolle kann ebenfalls mehreren Benutzern zugeordnet sein. Diese Relation wird als *User Assignment (UA)* bezeichnet. Genau so können auch Zugriffsrechte zu Rollen in einer many-to-many Beziehung zugewiesen werden, die als *Permission Assignment (PA)* bezeichnet wird.

Ein weiteres Konzept das in RBAC benutzt wird, sind *Sessions*. Ein Benutzer kann einer oder mehreren Sessions zugeordnet sein und jede Session ist genau einem Benutzer zugeordnet. In jeder Session kann wiederum eine Untermenge der Rollen, denen der Benutzer angehört, aktiviert sein. Welche Rollen, und damit auch welche Zugriffsrechte, in einer Session aktiviert sind, kann durch die Funktion *session_roles* abgefragt werden. Um zu erfahren, welche Sessions zu einem Benutzer gehören, kann die Funktion *user_sessions* aufgerufen werden.

Die obigen Eigenschaften beschreiben *Core RBAC* beziehungsweise $RBAC_0$. Formal kann damit Core RBAC wie folgt definiert werden.

Definition 1 (Core RBAC)

- USERS, ROLES, OPS und OBS*: Mengen der Benutzer, Rollen, Operationen und Objekten.*
- $UA \subseteq USERS \times ROLES$*: Menge der Zuweisungen von Benutzern zu Rollen.*
- $assigned\_users : (r : ROLES) \rightarrow 2^{USERS}, assigned\_users(r) = \{u \in USERS | (u, r) \in UA\}$*: Zuordnung der Rolle* r *zu der zugehörigen Menge der Benutzer.*
- $PRMS = 2^{(OPS \times OBS)}$*: Menge der Zugriffsrechte.*
- $PA \subseteq PRMS \times ROLES$*: Zuordnung der Zugriffsrechte zu Rollen.*
- $assigned\_permissions(r : ROLES) \rightarrow 2^{PRMS}, assigned\_permissions(r) = \{u \in USERS | (u, r) \in PA\}$*: Zuordnung der Rolle* r *zu der zugehörigen Menge der Zugriffsrechte.*

- $Ob(p : PRMS) \rightarrow \{op \subseteq OBS\}$*: Gibt die Menge der Operationen an, die einem Zugriffsrecht* p *zugeordnet sind.*
- $Ob(p : PRMS) \rightarrow \{ob \subseteq OBS\}$*: Gibt die Menge der Objekte an, die einem Zugriffsrecht* p *zugeordnet sind.*
- SESSIONS*: Menge der Sessions.*
- $user\_sessions(u : Users) \rightarrow 2^{SESSIONS}$*: Zuordnung eines Benutzers* u *zu einer Menge von Sessions.*
- $session\_roles(s : SESSIONS) \rightarrow 2^{ROLES}, session\_roles(s_i) \subseteq \{r \in ROLES | (session\_users(si), r) \in UA\}$*: Zuordnung einer Session* s *zu der zugehörigen Menge von Rollen.*
- $avail\_session\_perms(s : SESSIONS) \rightarrow 2^{PRMS}, \bigcup r \in session\_roles(s) assigned\_permissions(r)$*: Die Zugriffsrechte, die einem Benutzer in einer Session zur Verfügung stehen.*

Das Core RBAC Modell definiert die grundlegenden Eigenschaften jedes RBAC Modells. Es wurden auch weitergehende Modelle beschrieben, wie hierarchisches RBAC und constrained RBAC. Für die Definitionen dieser Modelle sei an dieser Stelle jedoch lediglich auf Ferraiolo et al. [FSGK+01] verwiesen

TODO: Übergang zum nächsten Kapitel

3 G-SIS

3.1 Klassifizierung von g-SIS Modellen

In vielen Fällen ist es notwendig, die Benutzer eines abzusichernden Systems in mehr als eine Gruppe einzuteilen. Diese Gruppen können sowohl nach der Art der Beziehung zwischen den einzelnen Gruppen unterschieden werden, als auch nach der Art der Beziehungen zwischen den Benutzern innerhalb einer Gruppe.

Verschiedene Gruppen in g-SIS könenn *verbunden* oder *isoliert* sein. Sind Gruppen untereinander *isoliert*, hat die Mitgliedschaft eines Benutzers, Subjekts oder Objekts keinen Einfluss auf die Mitgliedschaft eines Benutzers, Subjekts oder Objekts in einer anderen Gruppe.

Im Falle mehrerer *verbundener* Gruppen kann die Mitgliedschaft in einer Gruppe verschiedene Auswirkungen auf die Mitgliedschaft und die möglichen Operationen in einer anderen Gruppe haben.

Die Mitglieder innerhalb einer Gruppe können wiederum alle die gleichen Zugriffsrechte besitzen. Diese Art der Gruppe wird in g-SIS definiert als eine *undifferenzierte* Gruppe. Im Gegensatz dazu können in einer *differenzierten* Gruppe verschiedenen Benutzern auch verschiedene Zugriffsrechte eingeräumt werden.

Die genannten Unterschiede und ihre Kombinationen führen zu vier möglichen g-SIS Modellen. Diese sind das *isoliert undifferenzierte Modell*, das *isoliert differenzierte Modell*, das *verbunden undifferenzierte Modell* sowie das *verbunden differenzierte Modell*.

Das *isoliert undifferenzierte Modell* beschreibt das Modell, in dem die Mitgliedschaft in einer Gruppe keinen Einfluss auf andere Gruppen im System hat. Ebenso gibt es innerhalb der einzelnen Gruppen keinen Unterschied in den Zugriffsrechten der einzelnen Mitglieder, außer den Unterschieden, die durch die g-SIS Core Properties (s.h. Abschnitt **??**) für die Zugriffsrechte festgelegt werden. Das isolierte undifferenzierte Modell bildet damit die Grundlage für die anderen g-SIS Modelle.

Im *verbundenen undifferenzierten Modell* kann die Mitgliedschaft in einer Gruppe verschiedene Auswirkungen auf andere Gruppen haben, die im folgenden Abschnitt genauer betrachtet werden. Desweiteren haben die Mitglieder untereinander, wie im isolierten undifferenzierten Modell, die selben Zugriffsrechte. In Abbildung 1 sind die Beziehungen zwischen den einzelnen g-SIS Modellen grafisch dargestellt. Das isoliert undifferenzierte Modell ist in allen anderen Modellen enthalten. Das isoliert differenzierte Modell und das verbunden undifferenzierte Modell sind in diesem Sinne nicht vergleichbar. Das verbunden differenzierte Modell wiederum enthält alle anderen Modelle.

Im Folgenden soll allerdings lediglich das verbunden undifferenzierte Modell betrachtet werden, da dies ausreicht um Aussagen über die Beziehungen zwischen einzelnen Gruppen machen zu können.

TODO: Grafik einbauen

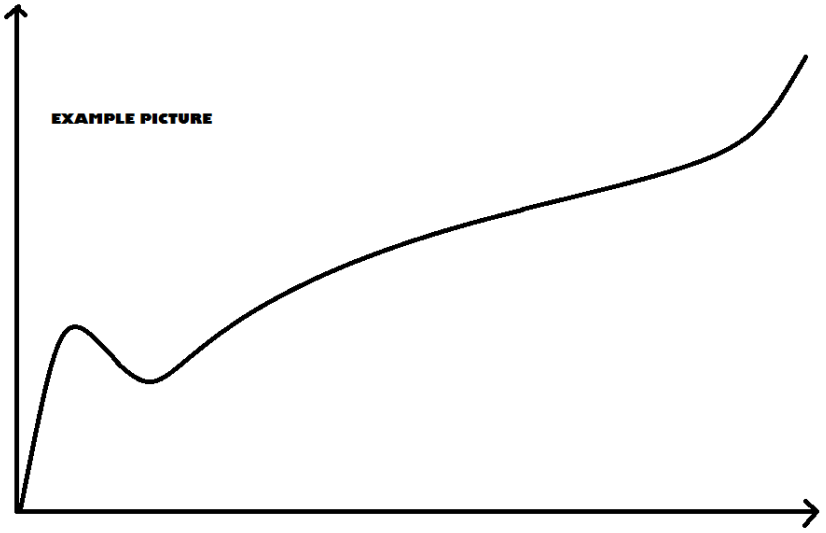

Figure 1: Hier g-SIS Modelle

3.2 Beziehungen zwischen Gruppen

Inter-group Relationship Semantics für connected undifferentiated group models.

4 Andere Policies in g-SIS

LBAC Policies in g-SIS, Domain and Type Enforcement.

RBAC Policies in g-SIS.

5 Fazit

References

[BSSW+95] L. Badger, D.F. Sterne, D.L. Sherman, K.M. Walker und S.A. Haghighat. Practical Domain and Type Enforcement for UNIX. In *Security and Privacy, 1995. Proceedings., 1995 IEEE Symposium on*, May 1995, S. 66–77.

[FSGK+01] David F Ferraiolo, Ravi Sandhu, Serban Gavrila, D Richard Kuhn und Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)* 4(3), 2001, S. 224–274.

[KSNW09] Ram Krishnan, Ravi Sandhu, Jianwei Niu und William H. Winsborough. Foundations for Group-centric Secure Information Sharing Models. In *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies*, SACMAT '09, New York, NY, USA, 2009. ACM, S. 115–124.

[Sand93] R.S. Sandhu. Lattice-based access control models. *Computer* 26(11), Nov 1993, S. 9–19.

[SCFY96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein und Charles E. Youman. Role-Based Access Control Models. *Computer* 29(2), Februar 1996, S. 38–47.

[SKNW10] Ravi Sandhu, Ram Krishnan, Jianwei Niu und WilliamH. Winsborough. Group-Centric Models for Secure and Agile Information Sharing. In Igor Kotenko und Victor Skormin (Hrsg.), *Computer Network Security*, Band 6258 der *Lecture Notes in Computer Science*, S. 55–69. Springer Berlin Heidelberg, 2010.