Noninterference

Alexander Osherenko[1]

[1] HU Berlin, Institute for Computer Science, Rudower Chaussee 25,
10099 Berlin, Germany
osherenk@informatik.hu-berlin.de

Abstract. In möglichen Sicherheitsmodellen ist der Noninterference-Ansatz angesiedelt. Der Noninterference-Ansatz basiert im Unterschied zu den anderen Methoden auf dem Prinzip der Kontrolle des Informationsflusses und nicht der des Zugriffsschutzes. Dabei werden einzelne Domänen im System genau beschrieben und auch definiert, wie Aktionen im System von den Domänen ausgeführt werden.

Einführung

In dieser Ausarbeitung möchte ich das Noninterference-Sicherheitsmodell beschreiben, dem die Arbeit von John Rushby zugrunde liegt ([1]). Dieses Model ermöglicht es, Sicherheit im Computersystem durch den Informationsfluss und nicht durch den Zugriffsschutz zu definieren. Mit den anderen Worten der Noninterference-Ansatz gibt einzelne Aktionen und entsprechende Ausgaben im System an. Dadurch wird festgelegt, welche Aktionen im System ablaufen können und welche Auswirkungen auf einzelne Komponenten im System die jeweilige Aktion hat.

Um eine Sicherheitspolitik mit dem Noninterference-Ansatz zu definieren, müssen die Systemaktionen und die Bereiche im System bestimmt werden und jeder Aktion muss ein solcher Bereich zugeordnet sein. Durch die Angabe einer Noninterference-Relation wird definiert, welche Bereiche von anderen nicht beeinflusst werden dürfen.

Noninterference deckt beide Gebiete für Security ab (Vertraulichkeit und Integrität). Dementsprechend lassen sich in Noninterference-Systemen sowohl Forderungen an die Vertraulichkeit als auch an die Integrität ausdrücken. Konkret heißt das für Bereiche *d* und *d'*, dass für Bereich d die Aktionen von d' nicht beobachtbar (Vertraulichkeit) und andererseits nicht beeinflussbar sind (Integrität).

Die Unmöglichkeit einer Beeinflussung wird dadurch ausgedrückt, dass für Bereiche *d* und *d'*; der Bereich d nicht unterscheiden kann, ob d' eine Aktion unternommen hat oder nicht. Wird kein Bereich von anderen Bereichen unerlaubt beeinflusst, so ist ein System sicher.

Im weiteren werde ich ein Beispiel bringen, welches das zu behandelnde Problem veranschaulicht, und anschließend einige Definitionen, die als Basis für die weitere Diskussion dient. Die vorgestellten Definitionen werden dann benutzt, um das Unwinding-Theorem einzuführen, welches die Beweise der Nichtbeeinflüssung zur Betrachtung der Zustandsübergänge führt. Abschließend stelle ich eine Frage zur Diskussion, die zur Einordnung des Modells in die Hierarchie der Sicherheitsmodelle dient ([3]).

Das einführende Beispiel

Sicherheitsaspekte werden in einem System untersucht, das in Abbildung 1 dargestellt wird. Es sind 4 Komponenten des Systems aufgezeichnet – Red, Bypass, Black, Crypto, wo Red und Black – Sender bzw. Empfänger einer Nachricht sind, Bypass-Komponente alle eingehenden Daten durchlässt, ohne sie zu verändern, und Crypto-Komponente Informationen verschlüsselt.

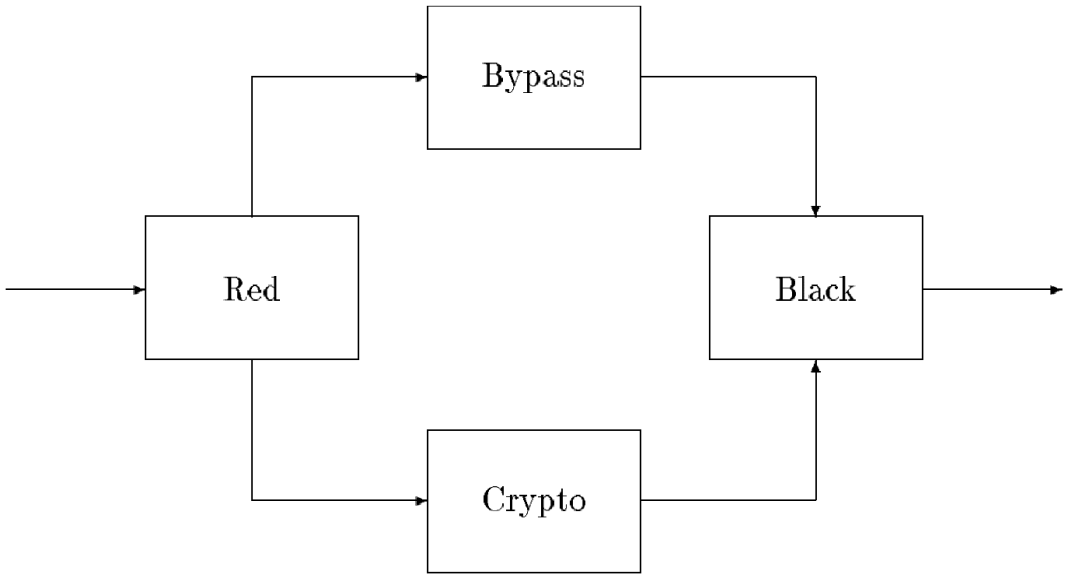

Abbildung 1. Das einführende Beispiel.

Im System wird eine Nachricht von der Red- zur Black-Komponente übertragen. Die Nachricht muss verschlüsselt werden, um in diesem System die Sicherheit der Daten zu gewährleisten.

Offensichtlich kann die Nachricht mit der Crypto-Komponente verschlüsselt werden, was gleichzeitig einige Probleme mit sich bringt – der Nachrichtenserver muss Nachrichtenkopf in unveränderter Form bekommen, um sie Nachricht zum Nachrichtenempfänger weiter schicken zu können.

Das heißt, dass die Nachricht in zwei Teile aufgeteilt werden muss, den Nachrichtenkopf und den Nachrichteninhalt, wo der Nachrichtenkopf durch die Bypass-Komponente unverändert zur Black-Komponente verschickt wird und der Nachrichteninhalt mit der Crypto-Komponente verschlüsselt wird.

Im weiteren wird aufgezeigt, welche Lösung für dieses Problem der Noninterference-Ansatz liefert.

Definition der Noninterference

Der Noninterference-Begriff wird aus dem Begriff domain und dem Begriff Nichtbeeinflüssung-Relation zusammengesetzt.

Bereich (engl. domain)

Für die Definition einer Sicherheitspolitik ist es nötig, die Komponenten eines Systems, wie z. B. Daten, Prozesse oder Schnittstellen, und seiner Umgebung, wie z. B. Benutzer oder Benutzergruppen, zu identifizieren. Um von den konkreten Komponenten zu abstrahieren, wird der Begriff Bereich (engl. domain) benutzt. Ein Bereich kann z. B. ein einzelner Benutzer, eine Datei, aber auch eine Menge von Benutzern und Dateien sein. Der von Goguen und Meseguer für die Security Gemeinschaft geprägte Begriff Interference (engl. für Eingreifen) steht für die Beeinflussung eines Bereichs durch einen anderen.

Die Noninterference-Relation (Nichtbeeinflüssung)

Durch die Angabe einer Noninterference-Relation $\rightsquigarrow$; wird definiert, welche Bereiche von anderen nicht beeinflusst werden dürfen. Basierend auf diesem zur Interference komplementären Begriff, lassen sich Sicherheitspolitiken definieren, die sowohl Vertraulichkeit als auch Integrität umfassen.

Abgrenzung von vorgestellten Modellen

Im Seminar wurden mehrere Sicherheitsmodelle vorgestellt z.B. Bell/La Padula-Model. Das Gemeinsame an diesen Modellen ist die Orientierung an die Zugriffskontrolle. Sicherheitsmodelle, die auf einer Zugriffskontrolle basieren, erlauben die Definition von Sicherheitspolitiken, durch die der direkte Zugriff von aktiven Subjekten, wie z. B. Personen, auf passive Objekte, wie z. B. Dateien, geregelt wird. Das Ziel einer solchen Sicherheitspolitik ist es, den Inhalt der Objekte vor unerlaubten Zugriffen zu schützen, ohne dabei den Subjekten vertrauen zu müssen.

Das Nonintereference-Modell unterscheidet sich von diesen Modellen, indem es das Informationsfluss-Konzept wählt. Der Informationsfluss beschreibt die Routen im System, über welche die Informationen im System fließen dürfen.

Idee
Die Idee von der Noninterference-Relation ist nahliegend. Sie kann indem ausgedrückt werden, dass der Security-Bereich *u* beeinflusst nicht den Security-Bereich *v* wenn keine Aktion, die vom Bereich *u* ausgeführt wird, verändert die Ausgaben, die der Bereich *v* sehen kann.

Im weiteren werden Definitionen gemacht, die das Noninterference-Modell beschreiben.

Definitionen

System
Unter dem System im Noninterference-Ansatz wird ein endlicher Automat verstanden, der eine Menge *S* von Zuständen, Menge *A* von Aktionen, eine Menge *O* von Ausgaben besitzt. Dazu kommt noch eine Menge von Bereichen *D*.

Kleinere Definitionen
Funktion *step* – die übliche für deterministische Automaten Zustandsübergangsfunktion – $S \times A \rightarrow S$

Funktion *output* – Ausgabefunktion definiert Ausgaben, die bei Ausführung von einzelnen Aktionen im System erfolgen.

Funktion *run* – Die Ausführungsfunktion beschreibt die Ausführung von Aktionenfolgen – $S \times A^* \rightarrow S$ und ist eine Erweiterung der *step*-Funktion auf mehrere Aktionen.

Funktion *dom* gibt die Zuordnung von Aktionen im System zu jeweiligen Bereichen im System – $A \rightarrow D$.

Funktion *purge*
Funktion purge beschreibt Aktionen im System, die ein Bereich beeinflussen können. Sie ist folgendermaßen definiert:

Sei $d \in D$ ein Bereich und $\alpha \in A^*$ eine Aktionenfolge, dann wird $purge(\alpha, d)$ durch die folgenden Gleichungen definiert:

$purge(\varepsilon, d) = \varepsilon$
$purge(a \circ \alpha, d) = a \circ purge(\alpha, d)$, falls $dom(a) \rightsquigarrow d$
$purge(a \circ \alpha, d) = purge(\alpha, d)$, falls $dom(a) \not\rightsquigarrow d$

Dementsprechend enthält ausschließlich Aktionen, die den Bereich *d* beeinflussen können.

Sicherer Zustand
Das Noninterference-Modell definiert einen sicheren Zustand im System folgendermaßen:

Ein System mit Anfangszustand $s_0 \in S$ ist sicher für $\not\rightsquigarrow$, wenn für alle $a \in A$ und $\alpha \in A^*$ folgende Gleichung erfüllt ist:

$output(run(s_0, \alpha), a) = output(run(s_0, purge(\alpha, dom(a)), a)$

Diese Definition bedeutet, dass die Ausgabe des Systems auch nach Bearbeitung von Aktionen in der *purge*-Funktion gleich bleibt.

Ausgabekonsistente Sichtenpartionierung
Als ausgabekonsistente Sichtenpartionierung definiert das Noninterference-Modell folgerndermaßen:
Eine Klasse $\sim = (\overset{u}{\sim})_{u \in D}$ von Äquivalenzrelationen auf *S* heißt Sichtenpartitionierung für eine Menge *D* von Bereichen. $\sim$ ist ausgabekonsistent wenn für alle $s, t \in$ S und alle $a \in A$ gilt:

$s \overset{dom(a)}{\sim} t \Rightarrow output(s,a) = output(t,a)$

Wenn zwei Zustände bezüglich einer ausgabekonsistenten Sichtenpartitionierung in Relation $\sim$ stehen, dann sind die Ausgaben aller Aktionen auf diesen Zuständen identisch.

M respektiert $\not\rightsquigarrow$ lokal
System M respektiert eine Noninterference-Relation $\not\rightsquigarrow$ lokal, wenn für alle $a \in A$ und alle $u \in D$ und alle $s \in S$ gilt

$$dom(a) \not\rightsquigarrow u \quad \Rightarrow \quad s \overset{u}{\sim} step(s,a),$$

wo $\sim$ eine Sichtenpartionierung ist.

Schrittkonsistent
System M ist schrittkonsistent, wenn für alle $a \in A$ und alle $u \in D$ und alle $s, t \in S$ gilt

$$s \overset{u}{\sim} t \quad \Rightarrow \quad step(s,a) \overset{u}{\sim} step(t,a),$$

wo $\sim$ eine Schichtenpartionierung ist.

Wenn das System ausgabenkonsistent ist, dann sind die ausgeführten Aktionen identisch.

Sicheres System
Sei $\alpha \in A^*$ und $a \in A$. Ein System ist sicher für die Policy $\rightsquigarrow$ wenn

$$test(\alpha, a) = test(purge(\alpha, dom(a)), a)$$

wo die *test*-Funktion ist definiert als:

test(α, a)=output(do(α), a)
do(α)=run(s_0, a)

Zusammenfassung der Definitionen

S	Zustände	
s_0	Anfangszustand	$s_0 \in S$
O	Ausgaben	
A	Aktionen	z. B. $\{hin, hout, lin, lout\}$
D	Bereiche	z. B. $\{high, low\}$
dom	Zuordnung von Bereichen	$dom : A \rightarrow D$
$\not\rightsquigarrow$	Noninterference-Relation	$\not\rightsquigarrow \subseteq D \times D$
$output$	Ausgabefunktion	$output : S \times A \rightarrow O$
$step$	Zustandsübergangsfunktion	$step : S \times A \rightarrow S$
run	Ausführung von Aktionenfolgen	$run : S \times A^* \rightarrow S$
$purge$	purge	$purge : A^* \times D \rightarrow A^*$

Beispiel
Sei *D = {high; low}*, *A = {hin; hout; lin; lout}*, *dom(hin) = high = dom(hout)*, *dom(lin) = low = dom(lout)* und *high* $\not\rightsquigarrow$ *low*. Somit gibt es vier Aktionen, die zwei Sicherheitsstufen zugeordnet sind, wobei die niedrige nicht durch die hohe Sicherheitsstufe beeinflusst werden darf. Für die Aktionenfolge α = [*hin; lin; hout; lout*] gelten die folgenden Gleichungen:

purge(α; low) = [lin; lout]
purge(α; high) = α

Das Unwinding-Theorem

Um die Sicherheit eines Systems gemäß der oben angegebenen Definition des sicheren Zustandes zu beweisen, müssen alle möglichen Aktionenfolgen – und somit unendlich viele – betrachtet werden. Um sich beim Beweis auf einzelne Zustandsübergänge zu beschränken, sind Unwinding-Theoreme hilfreich.

Sei $\not\rightsquigarrow$ eine Noninterference-Relation, *M* ein System und $\rightsquigarrow$ eine Sichtenpartitionierung von *M*. *M* ist sicher für $\not\rightsquigarrow$ wenn

1. $\rightsquigarrow$ ausgabekonsistent ist,
2. *M* schrittkonsistent ist und
3. M lokal $\not\rightsquigarrow$ respektiert.

Der Beweis erfolgt durch Induktion über die Variable α im Ausdruck

$$s \overset{u}{\sim} t \supset run(s, \alpha) \overset{u}{\sim} run(t, purge(\alpha, u))$$

Der genaue Beweis ist in [1] nachzulesen.

Grenzen des Ansatzes

Das Noninterferenz-Modell hat zwei bekannte Defizite – Probleme mit der Transitivität und Einschränkungen in Bezug auf nichtdeterministische Systeme.

Die gezielte Einschränkung von Informationsfluss zwischen Sicherheitsbereichen ist die Grundlage für den Noninterference-Ansatz. Die beschriebene Variante von Noninterference ist auf transitive Interference-Relationen beschränkt. Die Transitivität verhindert, dass Informationen, die nicht direkt von einem Bereich *d1* zu einem Bereich *d2* fließen dürfen, auch nicht über einen Umweg von *d1* nach *d2* gelangen. Für manche Anwendungen wird jedoch eine Herabstufung von vertraulichen Informationen benötigt, die mit einer transitiven Interference-Relation nicht formalisiert werden kann. Ist z. B. ein direkter Informationsfluss von *d1* nach *d2* unzulässig, d. h. *d1* $\not\rightsquigarrow$ *d2*, ein Informationsfluss über den Umweg *d3* jedoch erlaubt, d. h. *d1* $\rightsquigarrow$ *d3*, *d3* $\rightsquigarrow$ *d2*, so ergibt sich eine intransitive Interference-Relation. Nicht-transitive Interference-Relationen werden ebenfalls benötigt, wenn Kryptokomponenten eingesetzt werden, so dass vertrauliche Daten nach einer Verschlüsselung über offen zugängliche Netze versandt werden.

Der Noninterference-Ansatz betrachtet Systeme als deterministische Automaten. Einige Systeme weisen ein nichtdeterministische Verhalten auf. Es liegt nahe, im Noninterference-Ansatz den Nichtdeterminismus zuzulassen. In [1] ist aufgezeigt, welche Probleme dabei auftreten, beispielsweise dass das verfeinerte System unsicher sein kann, wo das sichere Verhalten des ursprünglichen Systems bewiesen worden ist.

Diskussion

Die folgenden Tabellen mit Klassifikation der Sicherheitsmodelle sind [3] entnommen. Die Frage für die Diskussion lautet – Welche Einordnung hat der Noninterference-Ansatz?

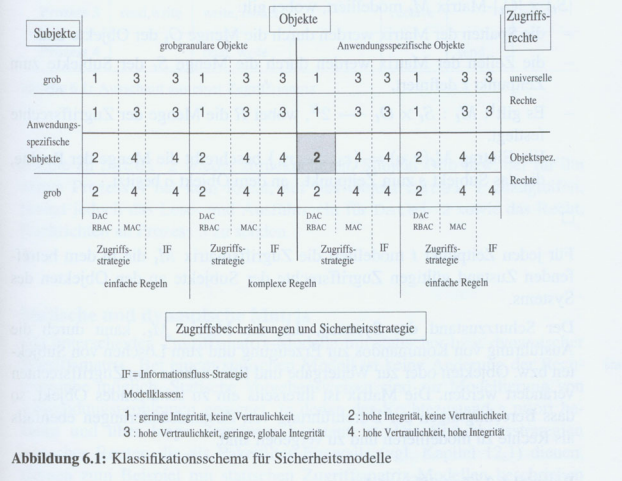

Abbildung 6.1: Klassifikationsschema für Sicherheitsmodelle

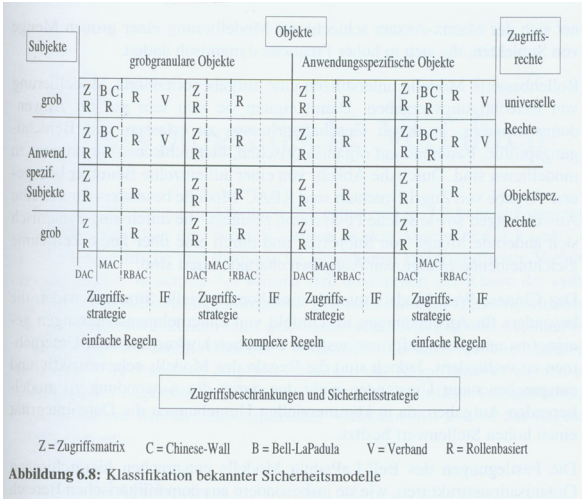

Abbildung 6.8: Klassifikation bekannter Sicherheitsmodelle

Literaturquellen

1. John Rushby – „Noninterference, Transitivity, and Channel-Control Security Policies“. URL: http://www.csl.sri.com/papers/csl-92-2/.
2. Heiko Mantel, Werner Stephan, Markus Ullmann, Roland Vogt – „Leitfaden für die Erstellung und Prüfung formaler Sicherheitsmodelle im Rahmen von ITSEC und Common Criteria”
3. Claudia Eckert – „IT-Sicherheit”