Data-Mining von Schwachstellendatenbanken

M. Schumacher, C. Haul, M. Hurler, A. Buchmann
Fachbereich Informatik
Technische Universität Darmstadt
schumacher@ito.tu-darmstadt.de
{haul,hurler,buchmann}@dvs1.informatik.tu-darmstadt.de

21. März 2000

1 Zusammenfassung

Wir sind immer noch weit davon entfernt, die Sicherheit von IT-Systemen in den Maßstäben zu gewährleisten, wie sie in anderen Branchen längst üblich sind. Dies liegt u.a. daran, daß IT-Systeme sehr komplex sind und sich in allen Phasen des Lebenszyklus Fehler einschleichen können. Als Beispiele seien an dieser Stelle Programmier- oder Konfigurationsfehler genannt. Wie in [And93] beschrieben, ist eine Verbesserung nur dann zu erwarten, wenn die Eigenschaften der Schwachstellen in IT-Systemen systematisch untersucht werden. In diesem Dokument stellen wir ein vielversprechendes Konzept vor, das auf Schwachstellendatenbanken (SDB, Vulnerability Databases) basiert. Die durch geeignete Data-Mining Verfahren gewonnenen Erkenntnisse helfen dabei, bestehende und neue Systeme sicherer zu machen.

2 Einleitung

Eine Systemkomponente weist eine Sicherheitslücke oder Schwachstelle auf, wenn sie nur in unzureichendem Maß gegen Mißbrauch geschützt ist. Wird diese Schwachstelle ausgenutzt, so ist die vorgesehene Sicherheit des betroffenen IT- Systems gefährdet. Dabei ist es unerheblich, ob hinter einem solchen Verstoß gegen die Sicherheitsrichtlinien Absicht oder Zufall steckt.

Heute vergeht kaum ein Tag, an dem nicht eine Sicherheitslücke in einem IT-System bzw. einer IT-Komponente aufgedeckt wird. Obwohl es einige Richtlinien für den sicheren Entwurf von IT-Systemen gibt, hat es nicht den Anschein, daß eine spürbare Besserung in Sicht ist. Die Forschung über die Ursachen und die Eigenschaften von sicherheitsbedrohenden Schwachstellen ist daher auf dem Gebiet der IT Sicherheit nach wie vor von hoher Relevanz.

Es gibt heute bereits eine Vielzahl unterschiedlichster SDBs, die vielfältige Informationen enthalten, die als Basis für wissenschaftliche Untersuchungen dienen können. Diese Datenbanken werden öffentlich und privat von den unterschiedlichsten Organisationen betrieben. Da Informationen über Schwachstellen einen Wettbewerbsvorteil darstellen können, werden allerdings wenig Anstrengungen unternommen, einheitliche Daten- und Betreibermodelle anzustreben, was u.a. zu einer hohen Datenredundanz und einer erschwerten Informationssuche führt.

Bisher finden wissenschaftliche, öffentlich zugängliche Arbeiten zu dem Thema ausschließlich in den USA statt. Da jede Art der Forschung auf dem Gebiet der IT Sicherheit unweigerlich das Interesse von amerikanischen Regierungsstellen, wie z.B. der NSA, weckt, werden praktisch keine Erkenntnisse an Forscher außerhalb der Vereinigten Staaten weitergegeben. Wir halten es aufgrund der Bedeutung des Themas für notwendig, unvoreingenommen an SDBs arbeiten zu können. Möglicherweise sind in Europa, z.B. aufgrund anderer gesetzlicher Bestimmungen, einige Ansätze möglich, die in den USA grundsätzlich ausgeschlossen sind. Da es beispielsweise in Deutschland nahezu keine Exportrestriktionen gibt, ist eine internationale Ausrichtung wesentlich einfacher. In diesem Dokument stellen wir zunächst die Arbeiten vor, die in den vergangenen Jahren die Fundamente für die wissenschaftliche Untersuchung von Schwachstellen gelegt haben. Darauf aufbauend werden die von uns als notwendig erachteten nächsten Schritte beschrieben. Teile dieser Überlegungen haben bereits das Ideenstadium verlassen und werden im Rahmen eines Forschungsprogramms an der TU Darmstadt realisiert.

3 Die gegenwärtige Situation

Bedeutende Betreiber von SDBs sind die verschiedenen „Computer Emergency Response Teams“ (CERTs). Sie warnen mit „Advisories“ i.A. nur vor extrem gefährlichen oder einen besonders breiten Personenkreis betreffenden Schwachstellen und beobachten darüber hinaus Gebiete wie etwa Computerviren oder Trojanische Pferde. Wenn auch die Forschung auf diesem Gebiet z.Z. überwiegend in geschlossenen Gemeinschaften innerhalb der USA [CC00] geschieht, so finden sich CERTs in verschiedenen Staaten wie z.B. auch Deutschland [Deu00] oder Australien [Aus00].

Weiterhin existiert eine große Anzahl von Informationssammlungen über Schwachstellen, die inhaltlich unterschiedlich ausgerichtet sind, beispielsweise auf verschiedene Betriebssysteme. Diese Informationen werden teilweise als klassische Datenbanken zur Verfügung gestellt, existieren jedoch auch in Form von Mailinglisten, Newsletters oder Newsgroups. Darunter befinden sich sowohl seriöse Quellen, wie private Organisationen, Unternehmen und staatliche Stellen, als auch „Hacker-Gruppen“. Ohne Anspruch auf Vollständigkeit erheben zu wollen, seien hier z.B. [Fir00], [ISS00], [CIA00], [Bug00], [Phr00], [Roo00], [Sho00], [Cap00], [Sec00a], [L0p00], [Det00] oder [Sec00b] genannt.

„*INFILSEC*“ [Sec00a] bezeichnet sich beispielsweise als eine „Vulnerability Engine“, die als Werkzeug für Hersteller, Systemadministratoren, Sicherheitsberater und Analysten dienen soll und verfolgt damit den Aufbau und Betrieb eines zentralen Repositorys für Schwachstellen von Betriebssystemen, Anwendungen und Protokollen. Außerdem werden Informationen darüber,

wie diesen Schwachstellen begegnet werden kann, gespeichert. INFILSEC will die Ergebnisse von Mailing-Listen wie Bugtraq extrahieren und über seine Suchmaschine zur Verfügung stellen. Dazu stellt INFILSEC ein Online-Update-System zur Verfügung, das die Möglichkeit bietet, Informationen in das System einzuspeisen.

Eine weitere Ressource stellt „*CIAC*“ [CIA00], die Computer Incident Advisory Capability, dar, die eine Einrichtung des amerikanischen Energieministeriums ist und allen Einrichtungen des Energieministeriums auf Anfrage bei Vorfällen, die die IT-Sicherheit betreffen unterstützt. Eine ähnliche Rolle, nur für die amerikanische Bundesregierung und ihre Einrichtungen spielt „*FedCIRC*“ [Cap00], die Federal Computer Incident Response Capability.

„*L0phT*“ [L0p00] wiederum ist ein IT-Sicherheits-Unternehmen, das aus einer Gruppe von Hackern entstand und regelmäßige Advisories zu IT-Sicherheitsproblemen herausgibt. Während der Ruf von L0phT sehr zwiespältig ist, da keine deutliche Abgrenzung zu illegalen Aktionen vorgenommen wird, so gelten die Informationen doch als zuverlässig.

Neben diesen öffentlichen Quellen zu Softwareschwachstellen, existieren vermutlich noch eine große Anzahl nicht öffentlich zugänglicher SDBs, deren Existenz teilweise noch nicht einmal bekannt sein dürfte. Eine Datenbank, deren Existenz bekannt ist, ist „*VulDa*“ von IBM, die ausschließlich für den internen Gebrauch bei IBM erstellt wurde, die jedoch eine nicht spezialisierte SDB darstellt [AD99]. Allein die von IBM für die Aktualisierung von VulDa genutzten öffentlichen Quellen vermitteln einen Eindruck über den zu betreibenden Aufwand: mehr als 30 Newsgroups, 60 Mailinglisten, Spiegelungen von über 45 FTP-Servern und Kopien von mehreren Dutzend „Hacker-Seiten“. Vermutlich werden auch öffentlich zugängliche SDBs als Quellen genutzt. Im März 1999 umfaßte VulDa circa 3,5 GByte komprimierte Daten.

Zusätzlich zu diesen allgemeinen Informationssammlungen betreiben viele Softwarehersteller eigene, stark spezialisierte Datenbanken, in denen die öffentlich bekannten Schwächen der eigenen Produkte dokumentiert werden. Wir nehmen an, daß Hersteller nur in seltenen Fällen auf Fehler hinweisen, die bisher nicht an die Öffentlichkeit gelangt sind. Es ist jedoch davon auszugehen, daß solche Fehler in internen, erweiterten Versionen der öffentlichen SDBs gespeichert sind.

Als wichtige Betreiber weiterer nicht öffentlicher SDBs kommen beispielsweise Geheimdienste in Frage, die ihre SDBs möglicherweise auch zu anderen Zwecken als der bloßen Abwehr von Angriffen auf ihre IT-Systeme nutzen möchten (z.B. Gegenangriffe) und kein Interesse daran haben, diese Informationen allgemein zugänglich zu machen.

Der Schwerpunkt der wissenschaftlichen Forschung auf dem Gebiet der SDBs liegt, wie in der Einleitung ausgeführt, in den USA. Dabei dienten den Forschern zwei „invitational“ Workshops als Foren. Der erste Workshop fand im Juni 1996 unter dem Titel „Workshop on Computer Vulnerability Data Sharing“ statt. Die Nachfolgeveranstaltung im Januar 1999 wurde unter dem Titel „2nd Workshop on Research with Security Vulnerability Databases“ durchgeführt. Der zweite Workshop behandelte neben technischen Fragen, motivierende Aspekte und mögliche Folgen der Unterhaltung von SDBs. Ein weiterer wichtiger Teil des Workshops beschäftigte sich mit Vor- und Nachteilen von unterschiedlichen Realisierungsmodellen für SDBs (siehe [MS99]).

Eine zentrale Rolle bei der Forschung auf dem Gebiet der SDBs spielt das „Center for Education and Research in Information Assurance and Security“ (CERIAS) an der Purdue University. Eine grundlegende Arbeit über Schwachstellen stellt die Dissertation von Ivan Victor Krsul mit dem Thema „Software Vulnerability Analysis“ [Krs98] dar, die er bei den COAST-Laboratories (inzwischen Teil von CERIAS) anfertigte. Krsul versucht, aufbauend auf früheren Arbeiten, eine Taxonomie für Software Schwachstellen aufzustellen. Tatsächlich stellt Krsul dabei zwei Taxonomien auf: Die erste ermöglicht eine a posteriori Klassifikation von Schwachstellen, die sich sehr gut für die Einordnung von tatsächlich aufgetretenen Schwachstellen eignet. Die zweite Taxonomie stellt eine a priori Klassifikation von Schwachstellen dar und ist für das bessere Verständnis und die Vermeidung derselben geeignet. Im Rahmen seiner Dissertation implementierte Krsul auch eine SDB. Diese ist im CERIAS anscheinend auch im Einsatz, sie ist jedoch nur für die interne Nutzung bestimmt.

Weitere Arbeiten fanden auch bei MITRE, einer privaten, staatlich geförderten Forschungsanstalt, statt. Dort arbeitet man an einer „Common Vulnerability Enumeration“ (CVE) [MC99], die eine Austauschbarkeit der Einträge verschiedener SDBs über eine gemeinsame Bezeichnung gewährleisten soll.

Außerdem wird die Forschung auf dem Gebiet der SDBs von verschiedenen staatlichen Organisationen, großen Industrieunternehmen und Unternehmen, die auf das Gebiet der IT Sicherheit spezialisiert sind, vorangetrieben. Aufgrund der Beteiligung der amerikanischen Regierung kann leider nicht immer davon ausgegangen werden, daß alle Forschungsergebnisse bzgl. der Konzeption von SDBs öffentlich zugänglich gemacht werden.

4 Erforderliche Forschungsarbeiten

Das übergeordnete Ziel ist es, eine IT-Umgebung zu schaffen, der Vertrauen entgegengebracht werden kann. Sicherheit bei elektronischen Geschäften oder bei der Erbringung von elektronischen Dienstleistungen erstreckt sich nicht nur auf eine abhörsichere und beim Transport unverfälschbare Kommunikation, sondern betrifft auch Aspekte der Verläßlichkeit von Diensten. Ein wichtiger Baustein muß daher sein, daß die Systeme, die für die Erbringung dieser Dienste benutzt werden, tatsächlich durch den Eigentümer kontrolliert werden und nicht kompromittiert sind. Zur Förderung dieser Aspekte wurde in Darmstadt das Rahmenprojekt TRUSTED[1] initiiert, der Aufbau einer SDB ist ein Teil von TRUSTED.

Ausgehend von einer möglichst umfassenden Sammlung der Daten über kompromittierte Systeme werden später Schlußfolgerungen gezogen. Diese Zustandsberichte sollten möglichst alle Wechselwirkungen mit anderen Softwarekomponenten aufzeigen können. Zusätzlich wird Hintergrundinformation über einzelne Komponenten benötigt, die teilweise direkt den einzelnen Komponenten zu entnehmen ist, teilweise aber durch Experten ergänzt werden muß. Dazu zählen beispielsweise Versionsnummern, Betriebssystem, Herkunft, Entstehungszeit, verwendete Bibliotheken und Bibliotheksfunktionen, Verwandtschaftsgrade im Sinne der Wiederver-

[1] *T*estbed for *R*eliable, *U*biquitous, *S*ecure, *T*ransactional, *E*vent-driven and *D*istributed Systems

wendung von Code sowie gleiche Teilfunktionalitäten. Mit einer solchen SDB werden bei TRUSTED vorrangig drei Ziele verfolgt:

1. **Bewertung** der Gefährdung eines Systems: Durch Informationen über vergleichbare, kompromittierte Systeme können Schwachstellen aufgezeigt und Gegenmaßnahmen empfohlen werden. Ergänzend kann die Aussagekraft der Bewertung durch die Bereitstellung von Testverfahren für einzelne Schwachstellen verbessert werden.

2. **Prognose** der Wahrscheinlichkeit für das Vorhandensein von Schwachstellen und der zu erwartenden Schwachstellenkategorie für neue, noch nicht verzeichnete Softwarekomponenten.

3. **Vermeidung** von bekannt fehlerhaften Entwurfsmustern bei zukünftigen Softwareprojekten: Durch Analyse der gefundenen Schwachstellen werden die zugrundeliegenden, fehlerhaften Entwurfsmuster identifiziert. Darauf aufbauend können korrigierte Entwurfsmuster entwickelt und bereitgestellt werden.

Besonders die Prognose und die Identifikation von fehlerhaften Entwurfsmustern stellen eine Herausforderung dar. Sie wird nur durch die Zusammenarbeit von Experten und maschinellen Analyseverfahren möglich. Die potentielle Datenmenge ist zu groß, um allein durch menschliche Experten beurteilt zu werden (s.o).

Vor der Auswahl der Mining-Methoden steht die Frage, welcher Art die ermittelte Information sein soll. Data-Mining dient i.d.R. zwei Zielen: Dem Entdecken vorhandener Muster in den Daten und der Vorhersage, zu welcher identifizierten Teilgruppe ein gegebener, neuer Fall wahrscheinlich gehört. Als zusätzliche Entscheidungsdimension liefert nur ein Teil der Methoden einen Satz von für Menschen verständlichen und überprüfbaren Regeln. Beispielsweise kodieren Neuronale Netze das „gelernte“ Wissen über die Eigenschaften der gefundenen Klassen durch die Gewichte der Knotenverbindungen. „Decision Trees“ hingegen generieren eine Hierarchie von Fragen, die sukzessive den Kreis der potentiellen Klassenzugehörigkeiten verkleinert. Solche Fragenkataloge können auch ohne Rechnerunterstützung ausgewertet und zur Klassifikation herangezogen werden.

Oft setzen Analysemethoden des Data-Mining ein „Training“ voraus. In einer solchen Trainingsphase wird die Gruppenzugehörigkeit von konkreten Fällen „gelernt“ und eine allgemeinere Beschreibung von Konzeptklassen abstrahiert. Mit Hilfe vorhandener Analysen von Schwachstellen können die Mining-Methoden trainiert werden. Anschließend kann mit Hilfe der Mining-Methoden für neue Fälle eine Wahrscheinlichkeit für die Zugehörigkeit zu den zuvor gelernten Klassen ermittelt werden.

Dabei treten im Zusammenhang mit SDBs vier grundlegende Probleme auf:

1. Es müssen genügend Trainingsinstanzen jeder Klasse verfügbar sein, um die Mining-Methoden zu trainieren.

2. Die bekannten Klassifizierungen, z.B. [Krs98], sind zwar umfangreich, können naturgemäß jedoch nicht erschöpfend sein.

3. Es finden sich keine Beschreibungen von nicht mit Schwachstellen behafteten Systemen in der Datenbank; das Konzept „frei von Schwachstellen“ kann von den Methoden nicht gelernt werden.

4. Die Beschreibungen der Schwachstellen sind weder vom Format noch von den verwendeten Begriffen identisch, so daß insbesondere eine maschinelle Interpretation schwierig ist. Versuche, diese Beschreibungen zu vereinheitlichen, sind bisher aufgrund des imensen Aufwands gescheitert [MC99].

Für das Training werden zwei nach Möglichkeit disjunkte Mengen von Trainingsinstanzen für das eigentliche Trainung und für eine Kontrollgruppe benötigt. Die Kontrollgruppe dient dazu, sicherzustellen, daß die Mining-Methoden nicht nur die bekannte Zuordnung von Fällen der Trainingssets zu Schwachstellenklassen lernt, sondern tatsächlich ein Abstraktionsprozeß stattfindet. Klafft die Qualität der Ergebnisse auf den beiden Mengen nach Beendigung des Trainings stark auseinander, so hat sogenanntes „over fitting“ eingesetzt. In diesem Fall kann der erlernte Parametersatz der Mining-Methode nicht sinnvoll auf unbekannten Daten angewendet werden und das Training muß wiederholt werden: entweder mit einem anders zusammengesetzten Trainingsset oder mit weniger Iterationen.

In der Anfangsphase steht jedoch die Identifikation von Schwachstellenklassen und damit das Verständnis von Schwachstellen und deren Entstehung im Vordergrund; noch nicht die Vorhersage von Schwachstellen. Daher werden Methoden eingesetzt, die mit „unsupervised learning“ initialisiert werden. Damit ist das Trainingset identisch mit der gesamten SDB und die Kontrollgruppe entfällt. Dennoch werden im Verhältnis zur Gesamtmenge der Einträge in der SDB „genügend“ Instanzen einer jeden Klasse von Schwachstellen benötigt, um von den Mining-Methoden tatsächlich als eigene Klasse identifiziert zu werden.

Unsupervised Learning bezieht sich auf die Trainigsphase der Methode und bedeutet, daß noch keine Klassifizierung der präsentierten Instanzen bekannt oder vorgegeben ist. Somit wird die Klassifizierung von Instanzen durch die Methode nicht bewertet oder überwacht. Damit eignen sich diese Methoden besonders für Gebiete, bei denen es noch keine vorgegebenen Klasseneinteilungen gibt. Als Ergebnis erhält man nun Gruppen von „ähnlichen“ Instanzen. Da diese Klassifikation nicht vorgegeben ist, kann sie neuen Einsichten über die Zusammenhänge in den Daten dienen.

Bei der Bewertung unbekannter Software kann nur indirekt auf die vermutliche Abwesenheit von Schwachstellen geschlossen werden: Nur durch Umkehrung, daß die Wahrscheinlichkeit, zu einer der anderen Konzeptklassen zu gehören, klein genug ist, deutet auf die Abwesenheit von Schwachstellen hin. Allerdings wäre selbst bei der Einbeziehung von Einträgen zu Software, die „frei von Schwachstellen“ ist, eine solche Vorhersage nicht notwendigerweise zuverlässiger; korrekter müßte die Klassifikation „frei von *bekannten* Schwachstellen“ lauten und würde falsche Sicherheit vortäuschen.

Das größte Problem erscheinen die unstandardisierten Schwachstellenbeschreibungen zu sein. Derzeit wird bei TRUSTED versucht, dieses Problem durch die Verwendung von dynamischen Ontologien wichtiger Schlagwörter aus den Schwachstellenbeschreibungen anzugehen. Dabei werden mit Hilfe einer logikbasierten Beschreibungssprache (siehe z.B. [BS85]) die Eigenschaften von Schlagwörtern katalogisiert. Dadurch können Mining-Methoden auf einem standardisierten Vokabular aufsetzen. Allerdings besteht weiterhin das Problem der Homonyme, also gleichlautenden Worten mit unterschiedlicher Bedeutung.

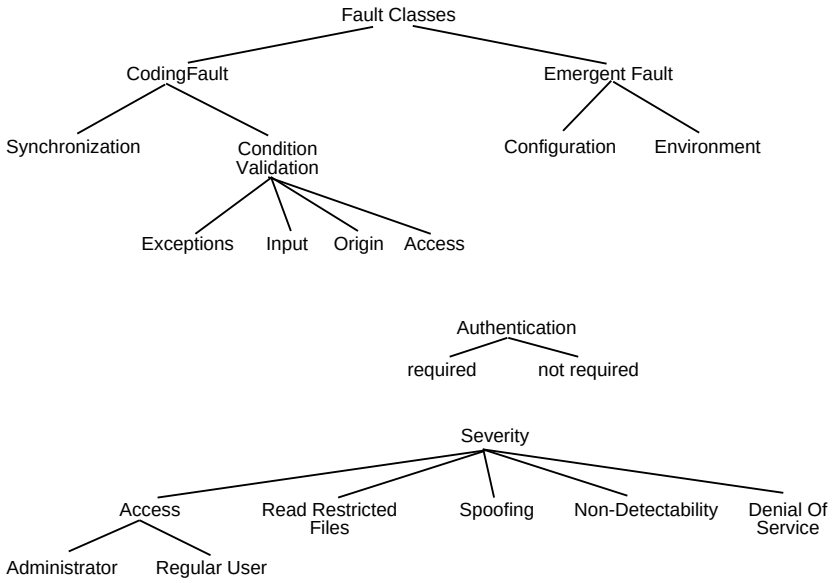

Abbildung 1: Ausschnitt aus einem möglichen Begriffsnetzwerk in Analogie zu [AKS, Kni00]

Schlagworte werden ähnlich der Abbildung 1 in Beziehung gesetzt. Zusätzlich können diese Begriffe definierende Attribute besitzen. Verschiedene Arten von Beziehungen sind möglich: Subsumtion, ein Begriff faßt andere zusammen, oder ein Begriff stellt eine definierende Eigenschaft dar. Angenommen, eine Schwachstelle ließe sich in Anlehnung an [Kni00] durch die Attribute *Fault, Severity, Authentication, Tactic, Consequence* beschreiben und für „Fault, Severity, Authentication“ sind die in Abbildung 1 angegebenen Kategorien definiert, so ließen sich später die Schwachstellen entsprechend kategorisieren, beispielsweise Schwachstellen, die keine Authentisierung vorausetzen und auf einem Fehler bei einer „Condition Validation“ beruhen.

Die Anwendung der Onthologien ist nicht auf diese Anwendungen beschränkt, sondern kann analog auf alle relevanten Begriffe angewandt werden.

Als Konsequenz können Mining-Methoden trotz uneinheitlicher Sprache der Schwachstellenbeschreibungen auf eine größere und detailliertere Trainingsbasis zurückgreifen.

5 Aktuelle Forschung zu SDBs bei TRUSTED

Grundlage für den Aufbau und Betrieb einer SDB ist es, ein Geschäftsmodell zu entwerfen, das mehrere Grundvoraussetzungen erfüllt: Zu den wichtigsten zählt dabei der wirtschaftliche Betrieb der SDB; es muß möglich sein, die entstehenden Kosten zu decken. Desweiteren ist es notwendig, daß eine größtmögliche Anzahl von Personen die SDB durch Beiträge unterstützt. Es ist außerdem notwendig, die Unterstützung von Unternehmen zu erhalten, die möglicherweise den Nutzen einer gemeinsamen SDB bezweifeln. Dazu kommen Fragen bezüglich des Zugangs zu den Informationen der SDB. Wer darf auf die Informationen zugreifen? Auf welcher Grundlage erfolgt dieser Zugriff? Wird anonym oder personalisiert auf die Daten zugegriffen? Wie werden Betroffene, die Informationen über eine Schwachstelle in ihrem System zur Verfügung stellen, vor einem potentiellen Angreifer geschützt? Wie wird die SDB organisiert? Der heutige Zustand mit vielen unabhängigen SDBs, der auch als balkanisiertes Modell bezeichnet wird, stellt eine sehr ineffiziente Organisationsform dar. Denkbar sind die folgenden Organisationsformen, die derzeit im Rahmen der laufenden Forschung untersucht werden:

- Das *zentralisierte* Organisationsmodell geht von der eher unwahrscheinlichen Konstellation aus, daß es nur genau eine zentrale SDB gibt, die von jedem ausschließlich genutzt wird. Dabei ist es durchaus möglich, daß für private bzw. vertrauliche Daten ein Schutz durch die zentrale Datenbank gewährt wird. Über diese zentrale SDB kann der Betreiber ein Maximum an Kontrolle ausüben und die SDB so nach seinen Vorstellungen gestalten.

- Das *föderierte* Organisationsmodell geht davon aus, daß es zwischen allen beteiligten SDB-Betreibern Absprachen oder Verträge hinsichtlich des Datenmodells und des Betriebs der SDB gibt. Die an der Föderation teilnehmenden Datenbanken halten im günstigsten Fall ihre Daten im Sinne einer horizontalen Partitionierung, so daß es beispielsweise ein Föderationsmitglied zum Thema „Windows NT“ und eine weiteres zum Thema „Solaris“ gibt.

  In eher weniger wünschenswerten Ausprägungen könnten die Föderationsdatenbanken ihren Inhalt via Replikation definieren oder im ungünstigsten Fall über Redundanzen. In diesem Fall kann das Modell in Richtung des unten beschriebenen balkanisierten Modells degradieren.

  Zusätzlich bietet das föderierte Modell den Teilnehmern die Möglichkeit, über private Instanzen des Datenbankschemas nur für sie relevante Daten lokal zu speichern und trotzdem für die Allgemeinheit interessante Daten an ein entsprechendes öffentliches Föderationsmitglied weiterzuleiten. Hier werden insbesondere die Interessen von privaten Organisationen gewahrt, ohne diese von der Beteiligung an und der Nutzung der SDB abzuhalten.

  Eine Sonderform des föderierten Modells stellt die Variante mit nur einer zentralen öffentlichen SDB und den privaten SDB-Instanzen dar. Dies entspricht dem zentralisierten Modell unter Einbeziehung vertraulicher, privater Instanzen. Das föderierte Modell

bietet auch den Vorteil, daß ihm trotz des Betriebs mehrerer, eventuell spezialisierter Datenbanken ein hohes Maß an Koordination innewohnt und so auch Recherchen über mehrere Datenbanken der Föderation unproblematisch durchgeführt werden können. Je nach „Lizenzmodell“ der Föderation kann so eine einheitliche Qualität des Inhalts erreicht werden.

- Das *Open-Source* Prinzip wird heute fast ausschließlich bei der Softwareentwicklung angewandt. Frei nach diesem Konzept ist auch ein Open-Source Modell für den Betrieb einer SDB vorstellbar. Dabei kann jeder auf alle Daten zugreifen und sogar die Datenbank in ihrer Gesamtheit kopieren und beispielsweise nach den Grundsätzen der GPL weiterentwickeln. Durch die Führung eines Projektleiters (Benevolent Dictator) kann eine zu starke Aufsplittung der SDB-Instanzen und eine damit verbundenen Degeneration des einheitlichen Datenschemas hoffentlich verhindert werden. Dabei stellt der Open-Source-Ansatz, der auf Daten und nicht auf Programmcode basiert, eine neue Variante dar, sozusagen einen *Open-Data-Ansatz*. Darüber hinaus ist es denkbar, daß durch den Open-Data-Aspekt bei vielen Nutzern eine höhere Motivation vorhanden ist, sich an der Arbeit und Erweiterung zu beteiligen. Als Ausgangspunkt könnte die *GNU Free Documentation Licence (FDL)* [Fou00] dienen.

- Das *balkanisierte* Organisationsmodell stellt den Ist-Zustand dar. Es gibt keinerlei Koordination oder Kontrolle zwischen den SDBs und die Anzahl der vorhandenen SDBs ist hoch. Cross-Referenzierung zwischen verschiedenen SDBs ist schwierig, wenn nicht unmöglich.

Abbildung 2 zeigt eine Einordnung der Betreibermodelle nach dem Kontrollaspekt und der Anzahl der zu erwartenden Datenbankinstanzen.

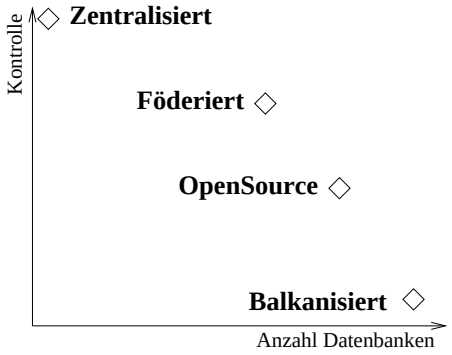

Abbildung 2: Einordnung der Organisationsmodelle

Ein weiterer wichtiger Aspekt ist die *Offenheit* des Modells. Darunter ist zu verstehen, ob eine SDB einen beliebigen, möglicherweise anonymen Teilnehmerkreis hat, über den die SDB-Betreiber nichts wissen oder ob eine, wie auch immer geartete, Zugangskontrolle stattfindet, die sich möglicherweise qualitätssteigernd auswirkt oder aber auch zur Finanzierung eingesetzt werden kann.

Bezüglich des Zugriffes auf die Datenbank ist zu überlegen, wie „Vertrauen“ in die Beiträge einzelner Teilnehmer geschaffen werden kann. Dazu wäre z.B. eine zentrale, qualitätssichernde Instanz denkbar oder aber ein Web-of-trust im Stile von Pretty-Good-Privacy. Je nach gewähltem Betreibermodell ist es auch denkbar, ein Bonussystem für gute Beiträge zum Datenbankinhalt einzuführen, mit dem eventuelle Kostenbeiträge zum DB-Management gemindert oder abgegolten werden könnten. Im Rahmen des DB-Managements wird man auch der Frage nachgehen müssen, inwiefern eine qualitätssteigernde Moderation der SDB-Beiträge sinnvoll oder wünschenswert wäre.

Ein weiterer wichtiger Aspekt ist die Frage nach einer geeigneten „Veröffentlichungspolitik“. Darunter ist zu verstehen, daß eine Schwachstelle, die neu gemeldet wird und noch nicht der breiten Öffentlichkeit bekannt ist, u.U. *nicht sofort* veröffentlicht wird. Statt dessen wird dem Hersteller der Software, soweit bekannt, eine entsprechenden Mitteilung bezüglich der Schwachstelle gemacht und ihm ein gewisser Zeitraum („grace period“) eingeräumt, in dem er eine Problemlösung erarbeiten kann, bevor die Schwachstelle über die SDB öffentlich bekannt gegeben wird. Bei der Wahl der Veröffentlichungspolitik und des Zeitraums, den der Hersteller zur Erarbeitung einer Problemlösung erhält, ist jedoch zu beachten, daß ein spezialisierter Angreifer erfahrungsgemäß schon von solchen Schwachstellen weiß. Unter der Vermutung, daß der entsprechende Personenkreis relativ groß ist, bewirkt eine lange grace period das Gegenteil der ursprünglich guten Absichten: Sie würde dann Angreifern ein größeres Zeitfenster für die Ausnutzung der entsprechenden Schwachstelle eröffnen, ohne daß die potentiellen Opfer auch nur versuchen könnten, möglichen Attacken mit „notdürftigen“ Maßnahmen entgegenzutreten, bevor eine endgültige Lösung zur Verfügung steht.

Aufgrund der besonderen Inhalte einer SDB, stellt diese ein besonders reizvolles Ziel für einen Angriff dar. Neben primitiven Attacken, wie etwa einem Denial-of-Service Angriff, sind insbesondere Angriffe auf die Integrität der Daten und die Identität der Benutzer bei der Planung der Schutzmaßnahmen des SDB Systems zu berücksichtigen.

So kann es bei einem eher offenen Betreibermodell leicht zu einer Verfälschung der Informationen kommen, die zu einer Verminderung der Qualität führen. Im schlimmsten Fall werden Informationen bewußt so manipuliert, daß bei den Nutzern der Informationen mehr Sicherheitslücken entstehen, als durch das vermeintliche Einspielen eines Patches, der z.B. ein Trojanisches Pferd enthält, beabsichtigt war. Beiden Fällen kann mit geeigneten qualitätssichernden Maßnahmen begegnet werden (s.o.).

Falls nicht geeignete Maßnahmen zur Anonymisierung der Nutzer getroffen werden, könnte ein Angreifer durch eine Analyse der Meldungen über Angriffe (Incident Report) personalisierte Profile erstellen. Daraus ließen sich Rückschlüsse auf die Systeme der Betroffenen ziehen, die gezieltere Angriffe ermöglichen können. Ein Werkzeug, daß die Informationen über einen konkreten Angriff erzeugen und in geeigneter Weise anonymisieren kann, wird z.Z. bei TRUSTED entwickelt.

Neben einer umfassenden Risiko-Analyse wird in einer weiteren Studie untersucht, welche rechtlichen Aspekte berücksichtigt werden müssen, wie z.B. die Frage nach der Haftung des Betreibers im Falle von Angriffen auf Basis von Informationen, die aus der SDB stammen, beantwortet werden. Interessant sind in diesem Zusammenhang auch länderübergreifende Ge-

setze, die bei einer europäischen bzw. internationalen Ausrichtung der SDB unbedingt beachtet werden müssen. Geklärt werden muß auch, wie z.B. mit den Urheberrechten der Autoren und der Hersteller umgegangen wird.

Wenn Daten- und Betreibermodell feststehen, muß eine initiale Füllung der SDB vorgenommen werden, was sich u.U. als sehr aufwendig erweisen kann. Die Fortschreibung des Datenbankinhalts kann unter anderem durch die Auswertung von Mailinglisten geschehen. Im Rahmen der laufenden Forschung werden die Möglichkeiten eines interaktiven Relevanzfilters untersucht. Außerdem betrachten wir Module zur Konvertierung von unterschiedlichen Datenstrukturen, um eine möglichst automatisierte Füllung vornehmen zu können. Durch die verstärkte Vorgabe von Meldeformularen könnten zukünftige Beiträge besser strukturiert und kompatibel mit dem Datenmodell der SDB erfaßt werden.

Ohne eine Entscheidung für eines der beschriebenen Geschäftsmodelle vorwegnehmen zu wollen, soll abschließend noch einmal betont werden, daß die „TRUSTED SDB“, wenn sie erst einmal eingerichtet ist, nicht nur für einige wenige, sondern für einen möglichst großen Personenkreis für unterschiedlichste Zwecke, insbesondere aber Forschungszwecke, zur Verfügung steht. Um eine möglichst hohe Akzeptanz zu erzielen, wird gerade zu diesem Thema im Internet eine Befragung durchgeführt [TRU00].

Wie auch in [Gra00] dargestellt, möchten wir durch die „schnelle und unbürokratische Veröffentlichung“ von Schwachstellen und einer „datenbankgestützten Qualitätskontrolle, die [...] nicht durch geheimgehaltene fachliche Autoritäten vorgenommen wird“, zu dem freien und vollständigen Zugang zu sicherheitsrelevanten Informationen beitragen.

Literaturverzeichnis

[AD99] D. Alessandri and M. Dacier. Vulda: A Vulnerability Database. Technical report, IBM Zurich, 1999. 3

[AKS] T. Aslam, I. Krsul, and E. Spafford. 7

[And93] Ross Anderson. Why Cryptosystems Fail. In *1st ACM Conference on Computer and Communications Security*, pages 215–227. ACM Press, 1993. 1

[Aus00] CERT Australia. CERT Australia. *http://www.auscert.org.au*, 2000. 2

[BS85] R. J. Brachmann and J. G. Schmolze. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, 9(2):171 – 216, 1985. 7

[Bug00] NT Bugtraq. NT Bugtraq. *http://www.ntbugtraq.com/*, 2000. 2

[Cap00] Federal Computer Incident Response Capability. FedCIRC. *http://www.fedcirc.gov/*, 2000. 2, 3

[CC00] CERT CC. CERT Coordination Center. *http://www.cert.org/advisories/index.html*, 2000. 2

[CIA00] CIAC. Computer Incident Advisory Capability. *http://ciac.llnl.gov/*, 2000. 2, 3

[Det00] Matthew Deter. Matt’s Unix Security Page. *http://www.deter.com/unix/index.html*, 2000. 2

[Deu00] CERT Deutschland. CERT Deutschland. *http://www.cert.dfn.de/*, 2000. 2

[Fir00] First.org. What is FIRST? *http://www.first.org/about*, 2000. 2

[Fou00] Free Software Foundation. GNU Free Documentation Licence. *http://www.gnu.org/copyleft/fdl.html*, 2000. 9

[Gra00] P. Graetzel von Graetz. Ein Paradigmenwechsel in der Wissenschaftspublizistik. *http://www.ix.de/tp/deutsch/inhalt/co/5726/1.html*, 2000. 11

[ISS00] ISS. X-Force Database. *http://xforce.iss.net/*, 2000. 2

[Kni00] Eric Knight. Computer vulnerabilities. Technical report, Security Paradigm, 2000. Draft, http://www.securityparadigm.com/compvuln_draft.pdf. 7

[Krs98] Ivan Victor Krsul. *Software Vulnerability Analysis*. PhD thesis, Purdue University, 1998. 4, 6

[L0p00] L0phT. L0pht Heavy Industries. *http://www.l0pht.com/*, 2000. 2, 3

[MC99] David E. Mann and Steven M. Christey. Towards a Common Enumeration of Vulnerabilities. *The MITRE Corporation*, 1999. 4, 6

[MS99] Pascal C. Meunier and Eugene H. Spafford. Final Report of the 2nd Workshop on Research with Security Vulnerability Databases. Technical report, CERIAS Purdue University, 1999. 3

[Phr00] Phrack. Phrack. *http://www.phrack.com/*, 2000. 2

[Roo00] Rootshell. Rootshell. *http://www.rootshell.com/*, 2000. 2

[Sec00a] INFILSEC Systems Security. INFILSEC. *http://www.infilsec.com/vulnerabilities/*, 2000. 2

[Sec00b] Securityfocus. Bugtraq. *http://www.securityfocus.com/forums/bugtraq/faq.html*, 2000. 2

[Sho00] NT Shop. NT Shop. *http://www.ntshop.net/*, 2000. 2

[TRU00] TRUSTED. Survey on Vulnerability Databases. *http://www.ito.tu-darmstadt.de/survey.html*, 2000. 11