Data Mining in Vulnerability Databases

M. Schumacher, C. Haul, M. Hurler, A. Buchmann
Department of Computer Science
Darmstadt University of Technology
schumacher@ito.tu-darmstadt.de
{haul,hurler,buchmann}@dvs1.informatik.tu-darmstadt.de

22nd March 2000

1 Abstract

We will still need quite some time to reach a security standard of IT systems alike the standard already usual in other fields. The reason for this is that IT systems are highly complex and errors are likely to sneak in all phases of their life cycle, e.g. programming or configuration errors. As written in [And93], we may only perceive an improvement of this situation through systematic examination of the vulnerabilities' characteristics. With this document we present a promising concept based on Vulnerability Databases (VDB). The insights gained through appropriate data mining procedures will help to render both, existing and new systems more secure.

2 Introduction

A system component shows a security breach or vulnerability if protected in an insufficient way against abuse. Once advantage is taken of this vulnerability, the security provided for the system in question is jeopardized. Note that it is irrelevant whether such a violation against the security guidelines is done on purpose or by accident.

These days no single day passes by without a security breach being detected in an IT system or in a new IT component, respectively. Despite the fact that there exist some guidelines for the design of secure IT systems, it does not seem that we sight a noticeable improvement. This is the reason why the research on the reasons and properties of vulnerabilities still ranks on the top.

At present there is already a multitude of various VDBs containing manifold information that can serve as basis for scientific studies. Theses databases are driven publicly or privately by

various organizations. Since information on vulnerabilities may be a competition advantage, little effort is made to strive for standardized data and provider models which in turn leads to a high data redundancy and an aggravated information search.

So far scientific studies being available to the public do only take place in the USA. Since any kind of research in the field of IT security does inevitably stimulate the interest of American governmental authorities, such as the NSA, only few insights are passed on to research community outside the United States. Due to the significance of the subject we consider it necessary to be in a position to study SDBs in an irrespective way. It might be that some approaches being excluded in principle in the USA are possible in Europe on strength of other legal regulations. Since, for instance, there are almost no export restrictions in Germany, an international orientation is much easier. With this document we first of all show the recent years' works being fundamental for the scientific studies on vulnerabilities. Basing on those we shall describe the following steps we consider necessary. Parts of these reflections did already quit the stage of ideas and will be realized within the scope of a research program at Darmstadt University of Technology.

3 The present Situation

Important providers of VDBs are the various “Computer Emergency Response Teams” (CERTs). With advisories they warn of vulnerabilities being highly dangerous or concerning a large group of people and in addition, they keep an eye on fields such as computer viruses and Trojan horses. Although research in this field is only taking place in closed communities inside the USA [CC00], CERTs are found in various countries, as e.g. also in Germany [Deu00] or Australia [Aus00].

Moreover, there exists a large number of collections of information on vulnerabilities focusing on various subjects, for instance on different operating systems. Parts of these information are made available as traditional databases, others do, however, exist as mailing lists, newsletters or newsgroups. Here we find reliable sources such as private organizations, companies and national authorities as well as hacker groups. Without making claim to be exhaustive, we cite in this place e.g. [Fir00], [ISS00], [CIA00], [Bug00], [Phr00], [Roo00], [Sho00], [Cap00], [Sec00a], [L0p00], [Det00] or [Sec00b].

INFILSEC [Sec00a], for instance, calls itself a “Vulnerability Engine” serving as tool for manufacturers, system administrators, security consultants and analysts and aims at building up and operating a central repository for vulnerabilities of operating systems, applications and protocols. Moreover, information on how to face these vulnerabilities is stored. INFILSEC wants to extract the results of mailing lists such as Bugtraq and to make these available via its search engine. For this purpose INFILSEC places an online update system at disposal which offers the possibility to feed information into the system.

Another resource is *CIAC* [CIA00], the Computer Incident Advisory Capability which is a utility of the American Ministry of Energy supporting on request all of its utilities at the occa-

sion of incidents concerning IT security. A similar role, for the American Federal Government, only, however, plays FedCIRC [Cap00], the Federal Computer Incident Response Capability.

L0phT [L0p00] on the other hand is an IT security company resulting from a group of hackers and issuing regular advisories on IT security problems. Although the reputation of L0phT is quite ambiguous since it does not clearly mark off illegal actions, those information are considered to be reliable.

Beside these public resources as to software vulnerabilities, there presumably exists quite a large number of VDBs not available to the public, the existence of which might even not be known. A database the existence of which is known is "Vulda" of IBM which was exclusively created for internal use at IBM which does, however, figure as example for a non-specialized VDB [AD99]. The number of public resources used by IBM for regular updates of VulDa already gives an impression on the effort to be done: more than 30 newsgroups, 60 mailing lists, mirrors of more than 45 FTP servers and copies of some dozens of "hacker pages". Presumably they also use publicly available VDBs as resources. In March 1999 VulDa was comprising approximately 3.5 Gbyte of compressed data.

In addition to these general collections of information, a lot of software manufacturers operate their own, highly specialized databases in which the publicly known vulnerabilities of their own products are documented. We suppose the cases to be rather rare in which manufacturers point to errors not known to the public so far. One can, however, expect such errors to be stored in internal, enhanced versions of the official VDBs.

Other organizations that could act as important operators of further non-public VDBs might be, for instance, Secret Services, who might possibly like to use their VDBs not only as a means for the mere defence of attacks on their IT systems but for other purposes (e.g. counter attacks), too and who are not at all interested in making theses information available to the public.

The main focus of scientific research in the field of VDBs is, as described in the introduction, situated in the USA. Thereby two "invitational" workshops served as forums for the researchers. The first workshop bearing the title "Workshop on Computer Vulnerability Data Sharing" took place in June 1996. The succeeding event in January 1999 was presented with the title "2nd Workshop on Research with Security Vulnerability Databases". The second workshop did not only treat technical questions but motivating aspects and possible consequences in running VDBs. Another important part of the workshop considered the question of benefits and drawbacks of various models of implementation for VDBs (see [MS99]).

The "Center for Education and Research in Information Assurance and Security" (CERIAS) at Purdue University plays a central role in research in the field of VDBs. The dissertation of Ivan Victor Krsul treating the subject "Software Vulnerability Analysis" [Krs98] that he issued at the COAST laboratories (part of CERIAS in the meantime) is a basic work on vulnerabilities. Building up on former studies, Krsul tries to put up a taxonomy for software vulnerabilities. In fact Krusl establishes two taxonomies: The first one allows an a posteriori classification of vulnerabilities and is very well suited for the classification of vulnerabilities actually occurred. The second taxonomy is an a priori classification of vulnerabilities and is suited for

a better understanding and avoidance of the same. In the scope of his dissertation Krsul also implemented a VDB. This one appears to be in use but it is, however, only meant for internal use.

Further works took also place at MITRE, a private, state-aided research institution. There, they are working on a “Common Vulnerability Enumeration” (CVE) [MC99] which guarantees the inputs of various VDBs to be exchangeable by means of a common identifier.

In addition, research in the field of VDBs is being pushed by various state organizations, large industry enterprises and enterprises being specialized in the field of IT security. The American government being involved, one cannot assume the results of the latest research as to the conception of VDBs to be made available to the public.

4 Required Research Projects

The superordinated aim is to create an IT environment one can have confidence in. Security in digital business or with rendering electronic services does not only concern communication to be bugproof and unaffected in transfer but concerns also aspects of trusting these services. Hence, an important building block must be that the systems used for rendering these services are in fact controlled through the owner and that they are not compromised. For the support of these aspects the frame project TRUSTED[1] was initiated in Darmstadt, the creation of a VDB is part of TRUSTED.

Starting from a quite substantial collection of data of compromised systems conclusions are drawn later. These status reports should show all interactions with other software components. In addition, background information on individual components is needed, parts of which may be taken directly from the components, others, however, need completion through experts. Some of those are, for example, version numbers, operating system, origin, time of origin, libraries used, library functions, relationships for the reuse of code as well as common partial functionalities. TRUSTED pursues the following three aims with such a VDB:

1. **Assessment** of the system’s hazard: Through information on comparable compromised systems vulnerabilities can be indicated and counter-measures can be recommended. For completion the force of expression of the assessment can be improved by providing test procedures for individual vulnerabilities.

2. **Prognosis** on how likely it is that vulnerabilities occur and on the category of vulnerability to be expected for new software components not yet registered.

3. **Avoidance** of known faulty design patterns with future software projects: Through analysing the vulnerabilities found the faulty design patterns behind are identified. Building up on this, the corrected design pattern can be developed and made available.

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

Especially the prognosis and the identification of faulty design patterns are a challenge. They are only possible through cooperation of experts and machine based analysis procedures. The potential amount of data is too large to be judged by human experts only (see above).

Before choosing mining-methods there is the question of what kind the information found shall be. In general, data mining suits two purposes: Discovering existing patterns in data sets and predicting which partial group a given new case might belong to. As an additional deciding dimension only a part of the methods can provide a set of rules that can be understood and verified by human beings. As an example neural networks assess the links between network nodes in order to code the “learned” knowledge about the properties of the identified classes. “Decision Trees”, however, generate a hierarchy of questions gradually making the circle of class affiliations smaller. Such question catalogues may also be evaluated without computer support and can be used for the purpose of classification.

Often analysis methods of data mining require a training. In such a training phase the group affiliation of concrete cases is learned and a general description of the concept classes is abstracted. With the aid of existing analyses of vulnerabilities mining methods can be trained. In the following, a probability for new cases can be given as to the affiliation of the classes learnt before by the help of mining methods.

Thereby we are facing four fundamental problems in the context of VDBs:

1. Enough training instances of each class have to be available in order to train the mining methods.
2. The known classifications, e.g. [Krs98], are extensive but, by nature, they cannot be exhaustive.
3. The database does not show any description of systems not bearing vulnerabilities; the concept “free from vulnerabilities” cannot be learnt through the methods.
4. The descriptions of the vulnerabilities are neither identical as to their format nor as to the notions used. Thus especially a machine based interpretation is difficult. Attempts to standardize these descriptions did fail so far on strength of the immense efforts [MC99].

For the training two possibly disjunct sets of training instances are needed one for the actual training and one for a control group . The control group suits to make sure that the mining method does not learn the known allocation of cases of training sets to vulnerability classes but that an abstraction process does indeed take place. If the quality of the results on both lots disperses too much after finishing the training, a so-called “over-fitting” has started. In this case, the learnt parameter set of the mining-method cannot be applied in a useful way on unknown data and the training needs to be repeated: either with a differently composed training or with less iterations.

In the initial phase, however, the identification of vulnerabilities ranks on top; not yet the prediction of vulnerabilities. Hence, there are methods used initialised with “unsupervised

learning”. Thus the training set is identical with the whole VDB and the control group can be dropped. Nevertheless, in relation to the total of entries in the SDB, “enough” instances of each class of vulnerabilities are needed to be in fact identified as individual class by the mining method.

Unsupervised learning refers to the initial training stage of the method and covers the fact that there does not exist a classification for the training instances or such classification is unknown. Thus classification is not supervised. Hence, such methods are especially suited for problem areas where no fixed classification scheme exists. Their results are clusters of “similar” instances. But since this grouping is not predefined it can give new insights into the data’s nature.

With the assessment of unknown software only indirect conclusions can be drawn as to the presumable absence of vulnerabilities. Only the inversion, that the probability to belong to one of the other concept classes is small enough, points at the absence of vulnerabilities. Such a prediction would, however, not necessarily be more reliable when including the entries as to software being “free from vulnerabilities”; to be more correct the classification should read “free from *known* vulnerabilities” and would pretend false security.

The non-standardized vulnerability descriptions seem to be a major problem. At present TRUSTED is trying to face this problem by using dynamic ontologies of important catchwords out of the vulnerability descriptions. Thus the characteristics of catchwords are catalogued with the help of logic-based description language (see e.g. [BS85]). Thereby mining methods can build on a standardized vocabulary. Nevertheless the problem of homonyms, identical words with different meaning, still exists.

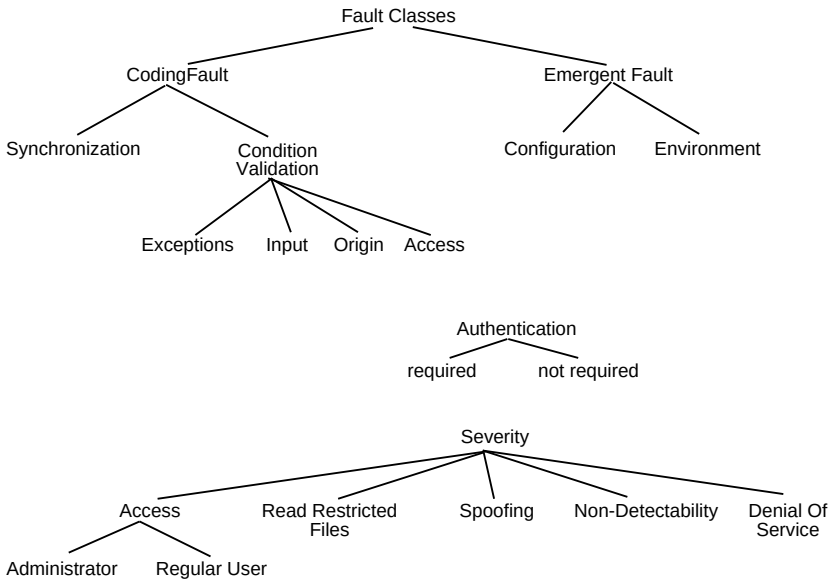

Figure 1: Imaginable Part of a Network of Catchwords

Catchwords are put into relation in a similar way as shown in figure 1. In addition, these terms can have defining attributes. Various kinds of relations are possible: Subsumption, one term sums up others or one term describes a defined characteristic.

Suppose a vulnerability could be described with attributes *fault, severity, authentication, tactic, consequence* like in [Kni00] and those attributes refer to the categories shown in figure 1, then vulnerabilities could be classified according to those relationships, e.g. vulnerabilities that do not need authentication and are caused by an error with “condition validation”.

Obviously, onthologies are not restricted to this example and could be extended to cover all relevant catchwords.

As a consequence, mining methods could build up on a larger and more detailed training basis despite a non-uniform language.

5 Latest Research on VDBs at TRUSTED

The basis for the setup and operation of a VDB is to design a business model which fulfills several main conditions: one of those is the economic operation of the SDB; it must be possible to cover the arising costs. Moreover, it is necessary that the greatest possible number of persons supports the VDB through their entries. In addition, it is necessary to get the support from companies who possibly doubt the benefit of a common VDB. Who is allowed to access the information? On which basis is this access effected? Is the access on the data anonymous or personalized? How will people concerned who put the information on a vulnerability at disposal be protected against a potential attacker? How is the VDB organized? Today’s state with a lot of independent VDBs, also called *balkanised* model, is quite an inefficient organization form. The following organization forms being tested at present are conceivable in the scope of the current research work:

- The *centralized* organization model starts from the rather unlikely constellation that there is exactly one central VDB which is exclusively used by everyone. Thereby it is absolutely possible that a protection for private or confidential data, respectively is granted through the central database. By means of such a central VDB the operator can have a maximum of control and can thus create the VDB according to his ideas.

- The *federated* organization model starts from the point that there are arrangements or contracts as to the data model and the operation of the SDB between all participating VDB operators. In the best possible case the databases participating in the federation hold their data in the sense of an horizontal partitioning so that there is, for instance, one federation member as to the subject “Windows NT” and another as to the subject “Solaris”.

  Less desirable peculiarities could be that federation databases define their contents via replication or in the worst case via redundancies. In this case the model can degrade in direction of the balkanised model described below.

Further the federated model offers to the users the possibility of locally storing data only relevant for themselves via private instances of the database scheme and nevertheless being in a position to hand on data being interesting for everyone to a corresponding public federation member. Here in particular the interests of private organizations are ensured without excluding those from the participation and the use of the VDB.

A special form of the federated model is the variant with only one central public VDB and some private VDB-instances. This corresponds to the centralized model taking into account confidential private instances.

The federated model also bears the advantage that it inherits, despite the operation of several, eventually specialized databases, a high degree of coordination and thereby queries to several databases of the federation can be made without problems. Thus depending on the “license model” of the federation a standard quality of the contents can be reached.

- The *open-source* principle is almost exclusively used with the software development nowadays. Based on this concept an open-source model is also imaginable for the operation of a VDB. Thereby anyone can access all data and can even copy the database as a whole and can, for instance, further develop according to the principles of the GPL. Through the guidance of a project leader (Benevolent Dictator) an extreme splitting up of the VDB-instances and thus a degeneration of the standardized data scheme can hopefully be prevented. Thereby, the open-source approach, based on data and not on program code, is a new variant, an *open-data* approach. Furthermore it is imaginable that, by the help of the open-data aspect a higher motivation exists with many users to participate in the work and the development. The *“GNU Free Documentation License (FDL)”* might serve as a starting point for such a license.

- The *balkanised* organization shows the present state. There is no coordination nor control at all among the VDBs and the number of existing VDBs is high. Cross-referencing among the different VDBs is difficult, if not impossible.

Figure 2 shows a rough classification of the operation models according to the control aspect and the number of database instances to be expected.

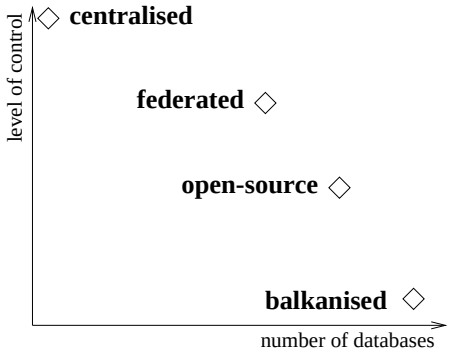

Figure 2: Classification of the organization Models

Another important aspect is the “openness” of the model. This means to understand whether a VDB has got any likely, possibly anonymous circle of participants the VDB operators don’t know anything about or whether a, no matter what kind of, access control takes place which in turn possibly increases the quality or can even be used for financing.

As concerns the access on the database it must be thought over how “confidence” in the entries of individual participants can be created. For that purpose e.g. a central quality-assurance instance could be thought of or a web-of trust in the style of Pretty-Good-Privacy. Depending on the operation model chosen it is also imaginable to introduce a bonus system for good entries to the database contents by which eventual cost contributions to the DB-management could be reduced or equalized. In the scope of the DB-management one also has to face the question to what extent a quality increasing moderation of the VDB contributions would make sense or be desirable.

A further important aspect is the question as to an adequate “publication policy”. This means that a vulnerability which is just reported and not yet known to the broad mass of the public, is possibly not published immediately. Instead of that the manufacturer of the software, if known, is given a corresponding message concerning the vulnerability and a certain period (grace period) is granted to him, in which he can work out a solution for the problem before the vulnerability is published via the VDB. When choosing the publication policy and the period given to the producer for the workout of the problem solution it does, however, need to be taken into account that, as experience shows, a specialized attacker already knows about such vulnerabilities. Presuming that the circle of persons concerned is quite large, a long grace period causes the contrary of the originally good intentions: It would open a larger time window to the attackers to exploit the corresponding vulnerability without giving any possibility to the potential victims to try to face possible attacks with intermediate measures before a final solution is available.

On strength of the special contents of a VDB, this one appears to be a specially attractive aim for an attack. Beside primitive attacks such as a denial-of-service attack, especially attacks on the integrity of the data and the identity of the user when planning the security measures of the VDB have to be considered.

Thus a rather open operation model can easily lead to the information being falsified which in turn decreases the quality. In the worst case the information is, by intention, manipulated that way that users of the VDB’s information introduce more vulnerabilities as intended by the application of a pretended patch which can for example include a Trojan Horse. Both cases can be faced with appropriate quality-saving measures (see above).

If appropriate measures to anonymize the user cannot be taken, the attacker could create personalized profiles through an analysis of the reports on attacks (Incident Report). Thereby conclusions could be drawn on the systems of the users concerned which in turn can facilitate more specific attacks. A tool which can create the information on a concrete attack and anonymize in an appropriate way is at present being developed at TRUSTED.

Beside an extensive risk analysis another study investigates the legal aspects that have to be considered such as e.g. how to answer the question of the liability of the operator in case of

attacks based on information from the VDB. In this context laws crossing various countries have to be taken into account when establishing the VDB on an international basis. It also needs to be cleared how to treat e.g. the copyrights of the authors and the producers.

Once the data and operation model are chosen one has to carry out an initial filling of the VDB which can possibly be quite an immense work to do. The continuation of the contents of the data can be done through the evaluation of mailing lists. In the scope of the current research the possibilities of an interactive relevance filter are examined. Furthermore we look at modules for converting different data structures in order to do a possibly automatized filling. Through increased usage of specimens for registration forms future entries could be recorded in a way being better structured and more compatible with the data model of the VDB.

Without anticipating the decision for one of the outlined business models it should be underlined again that the “TRUSTED VDB”, once being established, is not only available for some internal purposes but for a possibly large circle of persons for various purposes, in particular for research purposes. In order to determine the most acceptable operational properties of our VDB we carry out a survey which is accessible via WWW [TRU00].

As written in [Gra00], we want to contribute to the free and complete access to security related information through the “fast and unbureaucratic publication” of vulnerabilities and a “database-based quality control [...] that is not performed by non disclosed authorities”.

Bibliography

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

[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. 6

[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

[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

[Gra00] P. Graetzel von Graetz. Ein Paradigmenwechsel in der Wissenschaftspublizistik. *http://www.ix.de/tp/deutsch/inhalt/co/5726/1.html*, 2000. 10

[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. 3, 5

[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, 5

[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. 10