Security Assessment and Testing Tools Information Repository

Naaliel Mendes[1], João Durães[1], Marco Vieira[1], Henrique Madeira[1]

[1] CISUC, Department of Informatics Engineering – University of Coimbra
3030-290 Coimbra, Portugal

{naaliel,jduraes,mvieira,henrique}@dei.uc.pt

Abstract. *This paper presents a repository for storing and comparing the characteristics of security assessment and testing tools. This repository supports information on the tools from different sources, ranging from specialized web sites and magazines to academic publications. This is in fact a powerful mean to share and disseminate information on security testing tools and to raise security issues awareness. It can also represent a very important information source for researchers and practitioners interested in exploring those tools. A working version of the repository is already online.*

1. Introduction

Security assessment and testing tools such as vulnerability scanners and penetration testing tools have become a key factor to evaluate the security configuration of computer systems and applications. These tools aim to find and to prevent system vulnerabilities that could be discovered and exploited by attackers. In addition, the simulation of many sorts of attacks (such as denial of service, cross-site scripting, SQL injection, and buffer overflow) using these tools is aimed at helping system administrators to improve the security configuration of their systems in order to protect them against known attacks. In practice, the use of these tools ranges from end-users to academia and industry.

One important problem is the selection of the right tool for a given concrete scenario, as there is a variety of tools and their features are not easy to compare. A logical solution is to build an information repository to store relevant information on existing tools and help users on their decision when selecting a testing tool.

A repository is simply an infrastructure aimed at collecting data and offering functionalities to help user on finding, analyzing, comparing, and disseminating the repository data. A repository addressing security tools should store characteristics such as target system, type of addressed attack, and type of addressed vulnerability. Additionally, characteristics not directly related to the tools such as academic publications reporting on the utilization of tools are also relevant and should be stored in the repository.

With the prevalence of internet, repositories based on web-enabled tools have increased significantly the power of dissemination and sharing of repository data. Taking into account this web dissemination ability, many repository initiatives have been proposed on the web in several research topics (examples are given in the next section), also supporting the exploration of the stored data through business intelligence

techniques [1]. However, security assessment and testing tools have not been addressed by these previous initiatives.

In this paper we present the Security Assessment and Testing Tools Information Repository (SATTIR). SATTIR provides functionalities ranging from the submission of information on tools to the analysis of such information using well-known techniques of data exploration such as multidimensional analysis. To the best of our knowledge, this is the first repository in this research topic that addresses all the following characteristics:

- **Indexation of tools information from different sources.** This aims to collect and to index the information about tools, supporting contributions provided by repository users. This solves the problem of user's dependence on search engines and reading of publications to find out the state-of-the-art of security assessment and testing tools.
- **Comparison of tools.** The repository allows a detailed comparison of the tools characteristics and helps solving the problem of having to install and test the tools to assess their characteristics in detail. Today, the existence of multiple information sources on tools available (e.g., conferences, magazines, and announces of web sites) and the variety of the information formats have considerably increased the difficulties in comparing tools. This is a key contribution for end-users and system administrators, as they will be able to compare quickly available tools in order to find out which ones fit exactly their security testing purposes.
- **Support for advanced information analysis.** Technologies for information analysis such as Information Retrieval and On-line Analytical Processing (OLAP) are part of the repository functionalities. Data retrieval technologies explore information represented by text data. OLAP tools support the analysis of multidimensional data (data warehousing), being traditionally used to decision support analysis. The means for multidimensional analysis in SATTIR allow powerful comparisons of the tool characteristics, being possible, for instance, to query how many times a given tool was cited in a given type of publication (e.g., academic papers, magazines). Obviously, the significance of the conclusions taken from multidimensional analysis relies on factors as the diversity and amount of information stored in SATTIR. The information quality (a classical problem in OLAP systems) is not an issue given that we have defined procedures to assure that only real tools are taking part of the repository.
- **Dissemination and information sharing.** The repository is made universally available by web-enabled tools, allowing the sharing of the information on tools to end-users and to the industry and academic communities.

The structure of this paper is as follows. The next section presents related work. Section 3 shows the architecture and the underlying technology of SATTIR. The data storage infrastructure of SATTIR is detailed in Section 4. Section 5 discusses on the potential users and SATTIR exploitation scenarios. Section 6 presents the overview of SATTIR functionalities. Section 7 presents the conclusion and ongoing work.

2. Related Work

The collection and storing of data for future analysis or historical purposes continues being a hot topic in our days. A quick search for repositories on the web shows the existence of hundreds of repositories as well as the existence of software packages for building such repositories. These repositories cover different subject areas, being sponsored by universities, industries, and governments. This section presents some of these initiatives within the context of computer science. Sub-section 2.1 presents widely used repositories platforms that are employed for building new repositories and also shows examples of institutional repositories built on those platforms. Subsections 2.2 and 2.3 present respectively examples of data and software repositories. The subsection 2.3 is the one that presents the repository initiatives related directly to security assessment and testing tools.

2.1. Repositories Platforms

The popularity of repositories has motivated the proposal of platforms to help users in the creation of new repositories. The main idea is to endow community with tools that mitigate the technical difficulties of implementing a new repository (coding of scripts and web site interface, backups scripts, etc), which in fact demand considerable time and hard work from programmers and administrators. These tools typically offer an infrastructure where users may store data in a typical relational database, submit and search data through a web catalog interface.

Repository platforms have been mainly addressed in studies that demand the storage of huge amount of data, such as libraries, museums, and universities (called as institutional repositories). To help administrators in choosing platforms that better fit to theirs institution purposes, the Open Society Institute compared the characteristics of the most relevant open-source institutional repository platforms [2][3]. Here we present only three of the most representative of these platforms.

Eprints is a platform for building repositories, providing an easy and fast way “to set up repositories of research literature, scientific data, student theses, project reports, multimedia artifacts, teaching materials, scholarly collections, digitized records, exhibitions and performances” [4]. This platform runs over the MySQL database and the Apache Web Server and was coded in Perl programming language. One of the advantages of Eprints is that institutions can get it up and running relatively quickly and with a minimum of technical expertise, due to the size of the installed base. The archive section of the Eprints web site presents hundreds of repositories that are using this platform. Examples of repositories using Eprints are the Archive of European Integration (http://aei.pitt.edu/), the Cambridge University Engineering Department Publications Database (http://publications.eng.cam.ac.uk/), and Indian Institute of Science (http://eprints.iisc.ernet.in/).

DSpace is an open source dynamic digital repository solution for accessing, managing and preserving scholarly works [5]. In fact, this platform is also focused on the problem of long-term preservation of deposited research material. This platform runs over the PostgresSQL or Oracle databases and was coded in Java programming language. To search information in text files, DSpace use the Lucene search engine [6]. One of the main strengths of DSpace is the capacity of recognizing and managing a

large number of file format and mime types. Some of the most common formats managed within the DSpace environment are PDF and Word documents, JPEG, MPEG, TIFF files. DSpace argues that over two hundred academic and cultural institutions use their platform. Examples of repositories using DSpace are The Hong Kong University of Science and Technology Library (http://repository.ust.hk/dspace/), The American Museum of Natural History (http://digitallibrary.amnh.org/dspace/), and The Royal Naval Museum (http://www.seayourhistory.org.uk/).

The **Flexible Extensible Digital Object Repository Architecture** (Fedora) is a repository architecture for digital libraries proposed by Cornell University [7]. The implementation of this architecture is currently available through the Fedora software that is promoted by Fedora Commons organization [8]. It is a non-profit organization aimed at providing "sustainable technologies to create, manage, publish, share and preserve digital content as a basis for intellectual, organization, scientific and cultural heritage". This platform runs over MySQL, McKoi, and Oracle database and Tomcat web server, and was coded in Java programming language. Fedora's features cover aspects such as service-oriented architecture (repository access and management via web services), content versioning (repositories can maintain a record of how digital objects have changed over time), and basic search in the repository. Fedora tool package is available in Fedora web site, allowing the contribution of users with the proposal of new Fedora services. Examples of digital libraries repositories using Fedora are Encyclopedia of Chicago (http://www.encyclopedia.chicagohistory.org/), The National Science Digital Library (http://nsdl.org/), and The University of Virginia Library (http://www.lib.virginia.edu/digital/).

Although existing repository platforms are of paramount importance to the field of digital libraries, and even considering other initiatives [9] [10], [11], there is no repository platform addressing needs of the field of security assessment and testing tools. However, the study of the presented repositories was essential to assess the state-of-the-art of the top technologies employed in the construction of those repositories.

2.2. Data Repositories

Data repositories are an excellent resource to store and share information for research purposes. One type of valuable information that can be shared through data repositories is the results from field data studies. This subsection describes some of the fault and failure data repositories currently available.

The **Data & Analysis Center for Software** (DACS) is a Department of Defense Information Center supporting research on software reliability and quality. It serves as centralized source for data related to software metrics. The DACS maintains the Software Life Cycle Experience Database (SLED). This repository is intent to support the improvement of the software development process. The SLED is organized into nine data sets covering all phases and aspects of the software lifecycle [12][13]. Examples of these datasets are the DACS Productivity Dataset [14], the NASA/SEL Dataset, and the Software Reliability Dataset [15].

The **Metrics Data Program** (MDP) Repository is a database maintained by the NASA Independent Verification and Validation facility [16]. The repository is aimed at the dissemination of nonspecific data to the software community and it is made

available to the general public at no cost. All the data available in the repository is sanitized by the project representatives, and all the necessary clearances are provided. Users of the repository are free to analyze the data for their specific research goal. The repository contains data on the software projects that were collected and validated by the MDP program, spanning more than 8 years and including more than 2700 error reports. The information stored in the repository consists of error data and software metrics and error data at the function/method level.

The **Software Reference Fault and Failure Data Project** [17] is maintained by the National Institute of Standards & Technology and is aimed at the development of metrology, taxonomy and repository for reference data for software assurance. The project maintains a repository on software fault data specifically aimed at to help industry protect against releasing software systems with faults and to help assess software system quality by providing statistical methods and tools for analysis of software systems. The repository is available to the public upon request. The access to the information online allows users to view data and execute simple queries. Analytical and statistical use of the data is possible through a program developed within the project and available to the public.

The **Computer Failure Data Repository** (CFDR) is a public repository on computer failure data [18][19] supported by USENIX. The repository is aimed at the acceleration of the research on system reliability with the ultimate goal of reducing or avoiding downtime in computer systems. The CDFR repository is open to both obtaining and contributing data kind of users. The repository comprises nine independent datasets focusing mainly on very large storage systems. Example datasets are: the LANL dataset which covers 22 high performance computer systems encompassing more than 4700 machines during 9 years, and the HPC3 dataset which covers a 256 node cluster with 520 drives. The repository information covers many aspects, including: software failures, hardware failures, operator errors, network failures, and operational environment problems. Each failure is described by the time of start and end, the node affected and the classification of root cause. The raw data is available to the public [9] through a web interface. To-date the project does not offer online capability for analytic and statistical data-processing.

The **AMBER Data Repository** (ADR) has been developed within context of AMBER Coordination Action project, where our research team is integrated and leading the work of development of the ADR [1]. The repository is aimed at collecting raw data results from resilience assessment experiments to promote interaction among research groups. The repository is an infrastructure that integrates data from different sources in such way that enables comparison and cross-exploitation in a meaningful manner.

The AMBER data repository does not target data information related to security assessing and testing tools. In fact, the ADR and our proposal combined endow research community with powerful mechanisms to increase the impact of research. The first collects raw data from resilience assessment experiments. The second collects information about the tools used in these experiments.

2.3. Software and Information Repositories

The importance of software and software information repositories has increased in the last years, given factors such as the users necessity of finding tools to test or to assess the security aspects of their systems. This subsection presents the most relevant of these initiatives we have found.

SourceForge (SF) is the world's largest repository of open source software. It provides "free hosting to projects with a centralized resource for managing project, issues, communications, and code" [20]. This is a powerful example of software repository that offers, among other things, mailing lists, message forums, and version controlling. One interesting aspect of SourceForge is the ability of performing advanced search in its database through a user-friendly web-interface. Although its significant contribution to the overall open source movement, SourceForge is not a repository for finding security assessment and testing tools. It is a generic repository for open-source development software, though containing some projects related to security assessment and testing tools. Examples of software related to security testing in SourceForge are Wapiti (to audit the security of your web applications) and W3AF (a Web Application Attack and Audit Framework).

The **Open Web Application Security Project** (OWASP) is "a worldwide free and open community focused on improving the security of application software" [21]. Besides the coordination of diverse initiatives such as the ranking of the most common web application security vulnerabilities (OWASP Top Ten Project) and the proposal of security testing procedures and checklists (OWASP Testing Guide), OWSAP also recommends the use of security assessment and testing tools, such as WebScarab and WZFuzzer. However, the list of recommendation does not present tools that are not taking part of the OWASP and does not offer means for comparing the security characteristics of the tools.

Sectools.org is perhaps the most important reference in terms of indexing network security tools [22]. This ranks the favorites tools of a security mailing list with thousands of users. The last ranking edition categorized the top tools in classes such as password crackers, sniffers, vulnerabilities scanners tools, web scanners, wireless scanners tools, vulnerabilities exploitation tools, and packet crafters. Each tool ranked in this web site is described by one or more of the following attributes: type of license (commercial or free), operating system (Linux, Windows, Mac OS X, etc.), type of user interaction (command-line or GUI), and source code availability. However, the descriptions of the tools are too generic and the users need to access the site of the vendor to find specific characteristics such as the type of target system that the tool address. One important limitation in this site is the lack of a comparison of the functionalities of the tools.

3. Repository architecture and underlying technology

SATTIR was developed using widely known open-source technologies. Under a model-view-controller architectural pattern (MVC), the repository uses the Java EE and Spring framework as the programming technologies. The selected web server was the Apache Tomcat 6. The data warehousing and OLAP techniques are currently implemented using

the Mondrian and JPivot frameworks (http://mondrian.pentaho.org/). These technologies are distributed in the tree main parts of SATTIR Architecture (Figure 1):

- The **Data Collection Infrastructure (#1)** is aimed at providing mechanisms and interface where repository users can submit information on tools. The infrastructure relies on the user interaction to collect new data through the completion of a proper form. For that reason, in the current architecture mechanisms are not necessary to extract transform and load information from any source. Examples of mechanisms in these infrastructures are the JSP pages coded to build the form where uses will complete information.
- The **Data Storage Infrastructure (#2)** is constituted by the database of the repository. The database management system supporting the repository is the PosgreSQL 8. The relation entity model of the database is described in details in the next section.
- The **Dissemination and Sharing Data Infrastructure (#3)** contains the mechanisms that present the repository contents to the end-user, using the repository web servers. One important aspect here are the mechanisms that answer the analysis questions users requests (data warehousing, statistical analysis, and information retrieval).

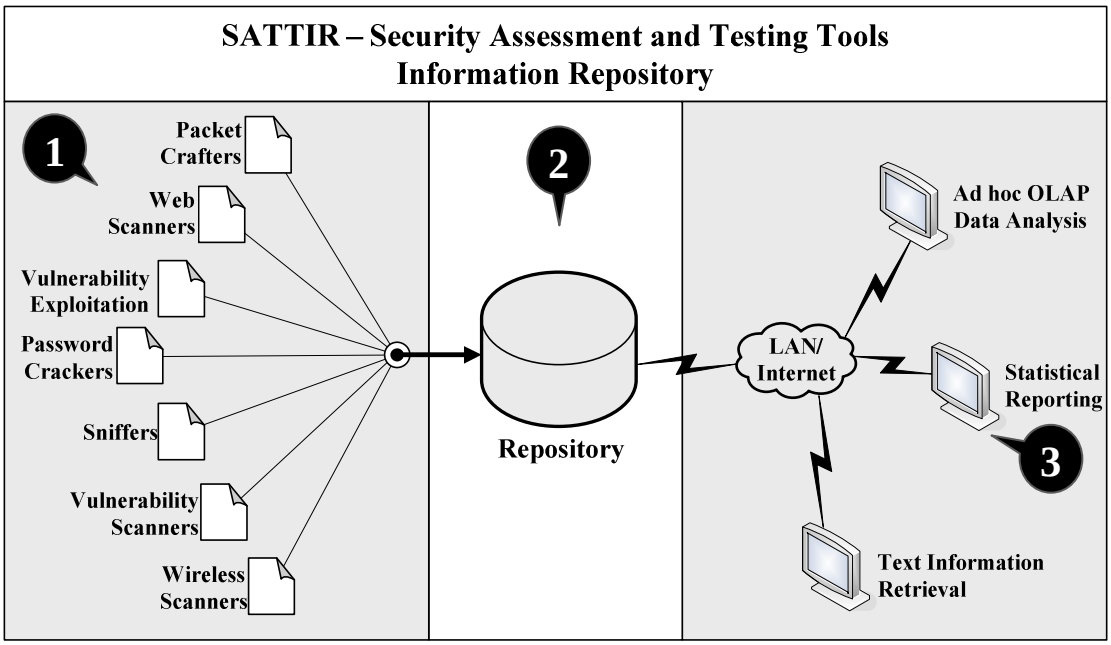

Figure 1. SATTIR Architecture

Business Intelligence (BI) and Information Retrieval technologies are part of the Dissemination and Sharing Data Infrastructure. In our repository, BI refers to multidimensional analysis, data warehousing and On-Line Analytical Processing.

A data warehouse is a global repository that stores large amounts of data that has been extracted and integrated from heterogeneous systems (operational or legacy systems) [23]. OLAP (On-Line Analytical Processing) is the technique of performing complex analysis over the information stored in a data warehouse [23]. The data

warehouse coupled with OLAP enables decision makers to creatively analyze and understand business trends since it transforms operational data into strategic decision making information.

In data warehousing the data is organized according to the multidimensional model (star model), which includes two kinds of data: facts and dimensions. Facts are numeric or factual data that represent a specific business or process activity and each dimension represents a different perspective for the analysis of the facts. Each dimension is described by a set of attributes. Note that the facts are just numerical quantities and only acquire meaning when are referenced to the dimensions. The OLAP analysis over a multidimensional cube consists of dicing and slicing the data in order to compute the desired measures. Figure 2 depicts one example of the data warehousing model within the context of our proposal, showing the fact table Citations in the center of the model. The entities around the fact table represent the model dimensions.

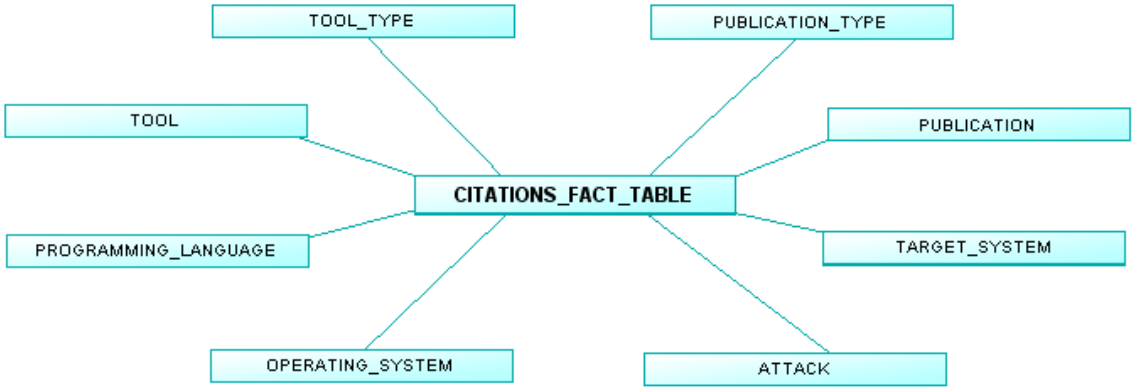

Figure 2. Example of data warehousing model

Information retrieval techniques will be used for raw data in textual formats that is not subject to a multidimensional characterization in facts and dimensions, nor even in a well defined set of attributes such as data used for data mining analysis. Textual data is indexed using traditional IR techniques based on inverted files and ranking algorithms to sort out search results by order of relevance. This way, the user can find specific patterns in the textual raw data, which is normally used to guide specific (and manual) data analysis of interesting cases (e.g., faults that lead to a unexpected behavior, not adequately handled by fault tolerance mechanisms available in the system).

4. Data storage infrastructure

The ability to provide meaningful comparisons of security tools depends essentially on the quality, the diversity, and the amount of information and the existence of a proper infrastructure to store and analyse that information.

Figure 3 presents a snapshot of SATTIR Class Diagram containing the most relevant entities of SATTIR. The main entity of this diagram, and of course of the repository, is the Tool Entity. This entity is the cornerstone of the repository, where all repository functionalities are based on, from storing to exploring information. The remainder entities represented in the class diagram are either related to intrinsic tool characteristics (represented outside the green box of Figure 3) or to extrinsic characteristics such as the academic publications related to a given tool. The intrinsic

tool characteristics are still differentiated by its relation with security aspects (represent by the red box) or not (all entities in the blue box except Tool Entity).

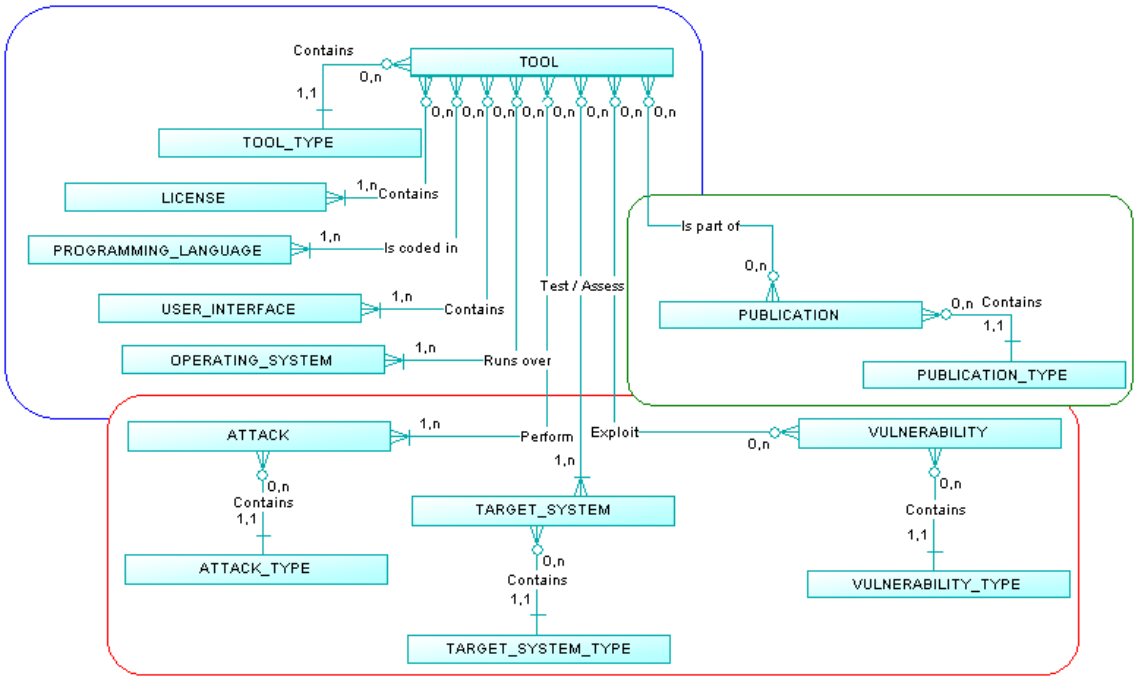

Figure 3. SATTIR Class Diagram

Another important aspect is to understand the type of information that can be stored in the repository entities. To this goal, we detail briefly the entities around the Tool Entity in Table 1 and Table 2. This main entity, therefore, contains attributes such as tool name, tool description, tool owner and other basics information of the tool.

Table 1. Description of SATTIR Entities (1/2)

Entity Scope	Entity Category	Entity Name	Entity Description
Internal	General	Tool Type	This contains values such as packet crafters, password crackers, sniffers, vulnerability exploitation, vulnerability scanner, web scanners, and wireless scanners.
		License	This refers either to commercial licenses or public licenses such as GNU General Public License (GPL), Academic Free License (AFL).
		Programming Language	This refers to the programming languages used in the coding of a given tool, containing values as Java, C++, and Python.
		User Interface	This refers to the means that user may interact with the tool, such as command-line and graphical interface.
		Operating System	This refers to the operating system that supports the installation of a given tool such as Windows, Linux, Unix, and Mac OS X.

With an in-depth focus on tool characterization and the proper design of the repository data storage infrastructure, it is possible to predict questions that may be queried to the repository. Two examples of meaningful questions that are answered by our model and that confirm the relevance of our approach are: How many citations a given type of tool received in a given period? Which are the tools that test or assess a given set of target system?

Table 2. Description of SATTIR Entities (2/2)

Entity Scope	Entity Category	Entity Name	Entity Description
Internal	Security	Target System	This refers to the system under test or evaluation such as web servers, web services, web applications, and wireless and network.
		Target Type	This contains values such as network and web systems.
		Attack Type	This refers to the category of attacks addressed by a given tool, such as Injection, Protocol Manipulation, and Sniffing Attacks [10].
		Attack	This contains values such as Cross-site scripting, SQL Injection, Buffer Overflow.
		Vulnerability Type	This refers to the category of vulnerabilities that a given tool exploits using its attacking functionalities. This contains values such as Access Control Vulnerability, Protocol Errors, and Environmental Vulnerability [10].
		Vulnerability	This contains values such as Insufficient Privileges, User Management Errors, and Empty String Password.
External	Citations	Publication Type	This contains values such as Magazine, Journal, and Proceeding Conferences.
		Publication	This refers to the description of the magazine, journal or conferences where a given tool was cited.

5. Potential users and exploitation scenarios

The main motivation of SATTIR is to be simple and fast, contributing to reduce radically the effort of users in finding, analyzing, and comparing security assessment and testing tools. To this goal, the repository can be used in two typical scenarios:

- As a form of common repository to store and share tool information. The current situation is not satisfactory as tool information is spread by different sources. SATTIR can change drastically this situation, as it uses a common format to share the data with the advantage of using multidimensional model. At the same time, the availability of tool information in SATTIR allows a very efficient dissemination of the tools recently proposed by industry and academic communities. **End-users**, **practitioners** and **researchers** may found in the repository the state-of-the-art of tools to test and evaluate the security of their systems and to employ in their experimental setups.
- To perform fast analysis of tool information in a very efficient way, ranging from cross-exploitation to comparative perspective. **Tool vendors** can analyze the strengths and the weakness of competing tools in order to improve the features of their own tools. **Researchers** will identify the research impact of a given tool through the analysis of the number of citations that it received in a given academic context. Vendors and researchers may still identify the necessity of proposing new tools to cover aspect still not addressed by the existents tools.

In practice there are two types of potential users: readers and writers. Readers access the repository only to explore and analyze tool information. Writers contribute to the repository by making tool information available to the community. Obviously, writers also use the repository to analyze their information, compare it to information provided by other repository contributors, and disseminate the data worldwide.

Henceforth, repository readers will be referred as **SATTIR Readers** and writers as **SATTIR Contributors**.

The simplicity of SATTIR architecture is emphasized through the tree steps needed to use the repository:

1. **User registration and authentication.** Before being able to access the repository, either SATTIR Readers or SATTIR Contributors have to register in the repository web site. Registration is then analysed and approved. Authentication is required every time users access the data repository. The registration is conditioned to the acceptance of the terms and conditions of the repository. The registration is required in order to allow future statistical analysis (e.g. the most compared set of tools in a given category of potential users).
2. **Submission of tool information.** After registration, SATTIR Contributors are able to send their information through the web interface of SATTIR. The availability of the submitted information in the repository relies on the approval of SATTIR Team. While SATTIR Team evaluates the authenticity of the provided tool information, which may take from 1 to 3 business days, the information will be set as pending state. The access to the pending states tools is just available to SATTIR Contributors. This aims to avoid duplicate efforts in order to register a given tool. However, contributors may suggest the improvement of the description of a given tool.
3. **Analysis of the information** using the tools and techniques (OLAP and Information Retrieval) available in the repository. This includes not only the normal cross-exploitation of tool information but also comparative analysis using multidimensional approaches to find out the differences among tool that addresses a given set of target system.

6. Functionalities overview

The preliminary version of SATTIR web site is currently available at http://security.dei.uc.pt/sattir. This web site was developed to provide easy access to the community to the repository functionalities.

The potential users of the repository must become registered users in order to get access to the main functionalities of the repository such as the exploration of third-party data (readers and contributors) and the submission of data (contributors). The relationship of registered users to the repository may be of several types: end-users, practitioners, makers, and researchers.

The interface of SATTIR web site (Figure 4) is constituted of the following sections:

Menu section (#1). The menu section allows the access to the main functionalities of the repository (e.g. tool exploration, tool submission, user registration, documentation).

Direct access section (#2). This section presents the authentication fields that users must fill to obtain access to protected repository functionalities (e.g. tool submission).

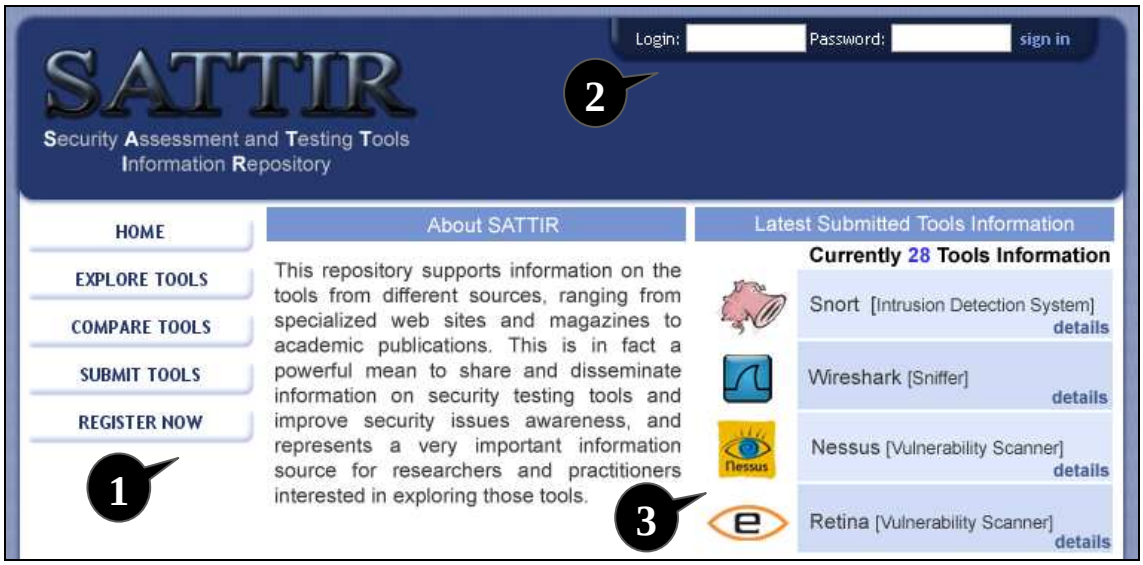

Figure 4. Home Page of SATTIR

Announce section (#3). This section shows the most recent updates in the repository such as the latest submitted tools information.

The main functionalities of repository web site are available through the menu section. The high importance of these functionalities leads us to describe the most relevant of them:

- **Submit tool.** This option allows the user to contribute data about a particular tool. The precondition here is to have an account with privileges of contributor. After the registration, the new repository member is able to send the characteristic of the tools to the repository through the completion of a proper form.
- **Explore tool**. This option provides the access to the most meaningful functionalities of the repository. This option offers to users the list of all tools that are available for analysis, showing a detailed description of those tools. Users also may select several tools and click on the compare button to see the characteristics of the tools side-by-side. Furthermore, if a given class of tools is meaningful for the user, there is an option where the user may check to receive information by email when new information is available.

To help disseminating the data, an up-to-date log of data addition and modification is posted in the web site focusing on the most recent additions or modifications. A mechanism for email notifications is also available. For example, users can request a notification to be sent when a given dataset is updated (data added or modified) or when a new dataset related to a particular research field is created.

7. Conclusion and Ongoing Work

In this paper we presented the **Security Assessment and Testing Tools Information Repository** (SATTIR). This repository provides to the community an infrastructure to collect and explore information about network security assessment and testing tools. It also offers useful functionalities to the community (end-users, practitioners, researchers,

and makers) through the indexation of tools information from different sources supporting the contribution of users, the comparison of tools characteristics using a relational database approach, the support of advanced information analysis using information retrieval and data warehousing, and the dissemination and sharing of tools information using web-enabled tools.

In fact, SATTIR is a potential tool for industry and research community to disseminate their security testing tools and prototypes. An important aspect is that, if successful, SATTIR will contribute to increase tools sales (in the case of commercial tools) and to augment the number of citations to the research papers describing the prototypes (in the case of research prototypes). To contribute to the repository, users just need to register and submit information about tools. The analysis of existing data is open to the entire world. The repository is maintained by an experienced team that will help users using the repository.

Ongoing work includes extending the current functionalities of the repository as well as adding data on more tools. The representativeness of this repository will also rely on the ability of storing and comparing the most meaningful tools. For that reason, we expect vendors, security and research community to provide data from network security assessment tools to the repository.

References

[1] AMBER Data Repository, available from http://amber.dei.uc.pt/repository/; Internet; accessed 29 March 2008.

[2] Crow, R. "A Guide to Institutional Repository Software.", Open Society Institute, 2004.

[3] Wyles, R., et al., "Technical Evaluation of selected Open Source Repository Solutions", Open Access Repositories in New Zealand (OARINZ) project, Christchurch Polytechnic Institute of Technology, v1.3, 13 September 2006.

[4] Eprints for Digital Repositories, available from http://www.eprints.org/; Internet; accessed 29 March 2008.

[5] DSpace, available from http://www.dspace.org; Internet; accessed 30 March 2008.

[6] Lucene Search Engine, available from http://lucene.apache.org/; Internet; accessed 30 March 2008.

[7] Payette, S., Lagoze, C., "Flexible and Extensible Digital Object and Repository Architecture (FEDORA)," Second European Conference on Research and Advanced Technology for Digital Libraries, Heraklion, Crete, Greece, September, 1998.

[8] Fedora Commons, available from http://www.fedora-commons.org; Internet; accessed 29 March 2008.

[9] Repository in Box, available from http://icl.cs.utk.edu/rib/; Internet; accessed 30 March 2008.

[10] Open Repository, available from http://www.openrepository.com/; Internet; accessed 30 March 2008.

[11] Proquest Digital Commons, available from http://www.proquest.com/; Internet; accessed 30 March 2008.

[12] DACS, Software Reliability Dataset, available from https://www.thedacs.com/databases/sled/swrel.php; Internet; accessed 15 March 2008.

[13] Delude, J., Vienneau, R., “Analyzing Quantitative Data Through the Web”, Proc. Of the 6th Annual Dual Use Technologies & Applications Conference, June 1995.

[14] Nelson, R., “Software Data Collection and Analysis”, Rome Air Development Center, Rome, NY, September 1978.

[15] Musa, J., “Software Reliability Data”, Data & Analysis Center for Software, January, 1980.

[16] NASA/WVU IV&V Facility, Metrics Data Program, available from http://mdp.ivv.nasa.gov; Internet; accessed 15 March 2008.

[17] Error, Fault, and Failure Data Collection and Analysis, available from http://hissa.nist.gov/project/eff.html; Internet; accessed 15 March 2008.

[18] Schroeder, B., Gibson, G., “The Computer Failure Data Repository (CFDR)”, Workshop on Reliability Analysis of System Failure Data (RAF'07), MSR Cambridge, UK, March 2007.

[19] The computer failure data repository (CFDR), available from http://cfdr.usenix.org/; Internet, accessed 25 March 2008.

[20] SourceForge, available from http://sourceforge.net/projects/repository/; Internet; accessed 29 March 2008.

[21] Open Web Application Security Project, available from http://www.owasp.org/index.php/Category:OWASP_Project; Internet; accessed 24 March 2008.

[22] Top 100 Network Security Tools, available from http://sectools.org/; Internet; accessed 24 March 2008.

[23] Chauduri, S., Dayal, U., “An overview of data warehousing and OLAP technology”, SIGMOD Record, March 1997.