https://doi.org/10.2352/ISSN.2470-1173.2017.6.MOBMU-309

Concept for software-based configuration of the organizational and technical security of a company of arbitrary size

***Thomas Möller** [1], **Knut Bellin** [2], **and Reiner Creutzburg** [2]*

[1] ***Assecor GmbH, Storkower Str. 207, D-10369 Berlin, Germany***
[2] ***Technische Hochschule Brandenburg, Department of Informatics and Media, IT- and Media Forensics Lab, P.O.Box 2132, D-14737 Brandenburg, Germany***

Email: thomas.moeller@assecor.de, bellin@th-brandenburg.de, creutzburg@th-brandenburg.de

Abstract

The aim of this paper is to show the recent progress in the design and prototypical development of a software suite Sunlight Security Systems (former Copra-Breeder) [1] *for semi-automatic generation of test methodologies and security checklists for IT vulnerability assessment of a company of arbitrary size.*

Introduction

Nowadays, companies, particularly small- and medium-sized enterprises (SME), have to deal more and more with the issue of IT security. Due to the ever-growing popularity of mobile devices, but also by the general acceptance of IT technology in everyday life, new security threats to corporate data occur every day [1-13].

In addition to the technical challenges, companies must also face new legal and organizational requirements, such as the introduction of ISO 27001.

Due to different terms and conditions within a company, there is no single security solution to counter all different threats.

In addition, dependent on the experience of the administrators, devices and services may be misconfigured and thus open security vulnerabilities.

Companies can protect themselves against such risks by assessing using penetration testing to get an accurate analysis of the threats and develop individual security concepts. However, there are two major challenges. How can companies be aware of the importance of security inspections? How can a check be offered so inexpensive that even in the face of SMEs regular checks are made possible?

One solution is to completely automate the vulnerability and penetration tests and to reduce the necessary oral audits to an essential minimum. With this approach, security audits could be carried out efficiently and with reduced effort and businesses are encouraged to perform these important checks regularly.

Typical Scenarios in a Security Assessment

In the analysis of companies, there are typical study areas that are everywhere carried out in the same way. A small selection include the following test methods:

1. Testing network security - Typically an organization's network is checked for its security. Here one has to focus, for example, on correct configuration of the firewall, protection of data transfer and accessibility and security of services. In addition, network spoofing attacks should be carried out to mitigate possible man-in-the-middle attacks.
2. Survey of important processes - In every company there are security-critical processes. These can be simple processes such as regular maintenance and upgrading of systems and analysis of log data. But often not IT-specific and complex processes can have a major impact on safety and security, such as the storage and processing of sensitive and mission-critical data that is printed. Also, the delivery of the first password is of great importance.
3. Testing the system security - In addition to the network the systems must be checked for security. Test direction here is whether the current patches are installed correctly, rights are reserved understandable and a virus protection and firewall concept is implemented.
4. Verification of system and service configurations - Besides the base system, also individual programs (in particular services that are reachable on the network) must to be tested in detail. These are one of the most frequent intrusion points in a company. Therefore, the configuration and timeliness must be checked in particular. In addition, accesses must be controlled, monitored and analyzed for malicious behavior and possible protection actions.
5. Testing the system responses to attacks - Despite all the tests of the individual security, the correct behavior can be confirmed only by attacks on practice tests. Therefore, for a full security check it is always recommended to perform so-called penetration tests. If possible, in addition to the standard attack vectors, individual industry or company-specific tests should be used in the tests.

In addition to these technical scenarios, typical requirements arise from legal data protection or other organizational requirements, eg from ISO 27001.

1. How can central password protection be ensured? Today's norms and laws generally require that passwords be distributed only centrally, for example, through an Active Directory, thereby ensuring that password guidelines, such as

[1] This work is supported by the German Bundesministerium für Wirtschaft und Technologie within the framework of ZIM ("Zentrales Innovationsprogramm Mittelstand") under project no. KF2565104ED4.

the character length, are adhered to. Exceptions and systems that do not support this need to be documented. How can this be proven?

2. Wie kann der zentrale Passwortschutz sichergestellt werden Heute Normen und Gesetze fordern in der Regel, dass Passwörter nur zentral, beispielsweise durch ein Active Directory verteilt werden dürfen. Ausnahmen und Systeme, die das nicht unterstützen, müssen dokumentiert sein. Wie kann das bewiesen werden?

In today's practice a wide variety of testing tools and questionnaires for these typical test problems are available.

To a large extent, the questionnaires to be prepared and processed are often very expensive to handle. The problem with the questionnaires is that repetitions occur by similar types of questions and very detailed answers are required in many areas, often without discussing company-specific idiosyncrasies.

The problem with the application of test tools and suites is that they do not cover all areas. If a lot of individual testing tools is used for an inspection, a more detailed analysis can usually be done, however, a high level of knowledge about these tools is required. If a composite test suite is used often it can not cover all necessary details.

At the end of the tests and audits carried out one is facing the challenge of how these various results from tests and audits are combined. There must be clear and efficient rules, how to deal with conflicts between tests and interviews, and there must be an analysis of whether multistage attacks (because of the structure) would be successfully.

But what alternative or overall solution can be designed to facilitate the work?

Architecture of Copra Breeder

In cooperation between Assecor GmbH and the Brandenburg University of Applied Sciences a plan was developed to overcome these problems of security checks. The project is entitled under the name of "Copra Breeder".

The software suite Copra Breeder (CB) (see Figure 1) will be developed to solve the necessary tasks of security investigations in several individual program modules. The results of all individual program modules are stored centrally in order to enable a comprehensive analysis.

In the following the components and functionalities are described in more detail.

Copra Breeder Checker

The Copra Breeder Checker (CB Checker) (see Figure 2) is a component that provides various questionnaires. Some questionnaires include for example:

- protection requirements analysis of a company,
- creation of an IT security policy for a company,
- generation of process structure questions,
- supplementary questions about penetration testing.

These questionnaires (some of which are freely available from public authorities) shall be transferred to a general structure and stored in a standard database. With this general structure it is possible to supplement central information of a company and thus directly to note weaknesses in an interview or audit. At the same time, the questions can be structured so that questions about technologies will not be repeated unnecessarily.

The CB checker is implemented as a web interface to allow access to the questions using various operating systems.

Copra Breeder Investigator

The Copra Breeder Investigator is the central control unit for all automated tests.

The Copra Breeder Investigator (CB Investigator) (see Figure 3) has sensors to perform various tests on the systems. Some of these sensors are installed directly on the test system, others can be on any other system that the investigator can access.

These sensors are nothing more than addressable interfaces that existing programs "wrap" or make other data or interfaces available. Thus, these sensors may provide, for example, the current network traffic of a test system for analysis or connect existing systems for external testing.

The data from the sensors are evaluated by the investigator and shall provide information on the test network and test systems, together with the found vulnerabilities in the CB Central for storage.

For the test procedures of the CB Investigator various workflows are available. These include the plan for individual test sequences. The workflows are determined through defined actions which tasks are to be met in order to obtain the data, or to identify a vulnerability. These audit workflows decide in advance whether the use of a specific workflow is possible. New workflows can be added anytime for company-specific requirements.

This allows that several different workflows can be combined to form a whole and so discovering of new vulnerabilities across the network or even entire test series to be developed. This means that the previous workflows represent sub-workflows under the currently developed workflows.

The CB Investigator will provide a web interface through which the sensors for selected test scenarios are configured and the current actions of the CB Investigator may be inspected. In addition, the interface can provide an overview of the currently tested network and the already found vulnerabilities.

Copra Breeder Central

The component Copra Breeder Central (CB Central) is the central repository of all network security knowledge of a company. It takes control of communication among all components and provides the basic security architecture.

The storage of data is carried out in a schema-less database. Thus, no complicated schema changes in the extension and modification of test tools and analysis algorithms must be made. At the same time the database structure remains simple and clear.

The security structure of CB is realized fundamentally about certificates. Exclusively self-signed certificates are used, which are independent of the particular corporate structure. This ensures that any malpractice with the root certificate from the company does not cause weakness in the architecture of the CB. During the transmission of all data, regardless of their degree of sensitivity, they are multiple encrypted with different algorithms.

Copra Breeder Analyzer

The Copra Breeder Analyzer (CB Analyzer) (see Figure 4) is responsible for the analysis of all data and the creation of a fi-

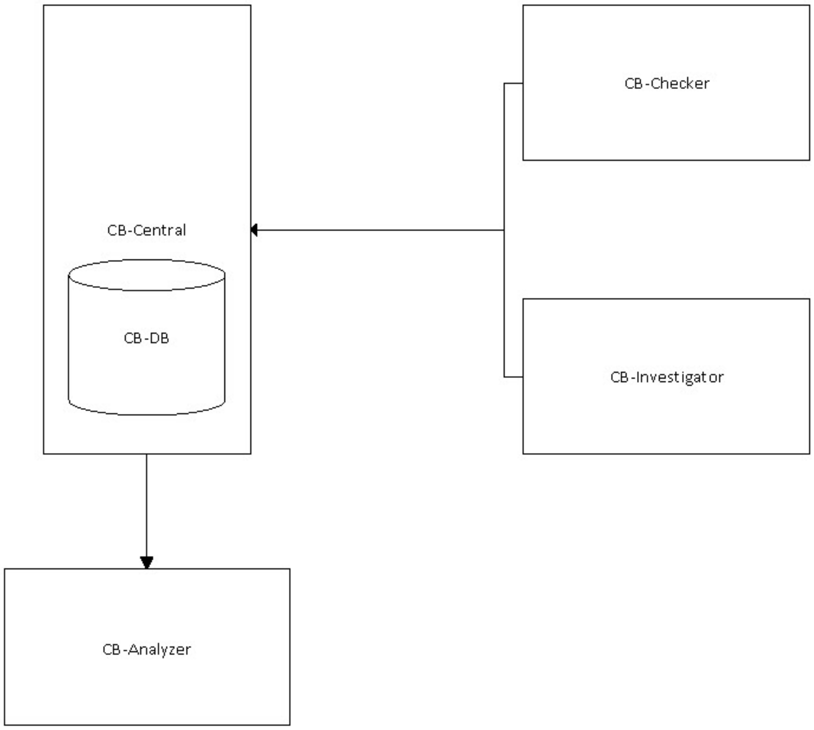

Figure 1. *Overall structure of Copra Breeder.*

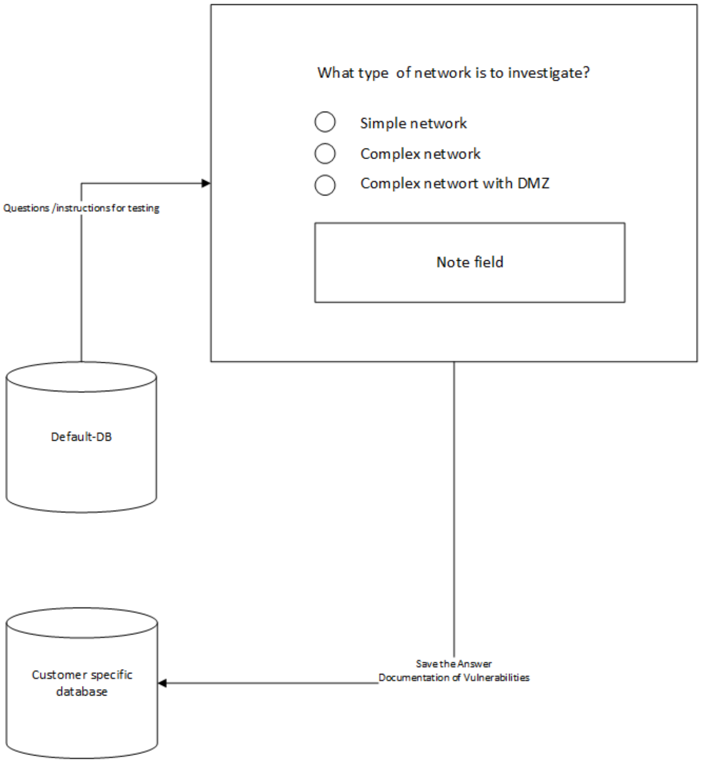
What type of network is to investigate?
○ Simple network
○ Complex network
○ Complex networt with DMZ
Note field
Questions /instructions for testing
Default-DB
Save the Answer
Documentation of Vulnerabilities
Customer specific database

Figure 2. *Overall structure of Copra Breeder Checker.*

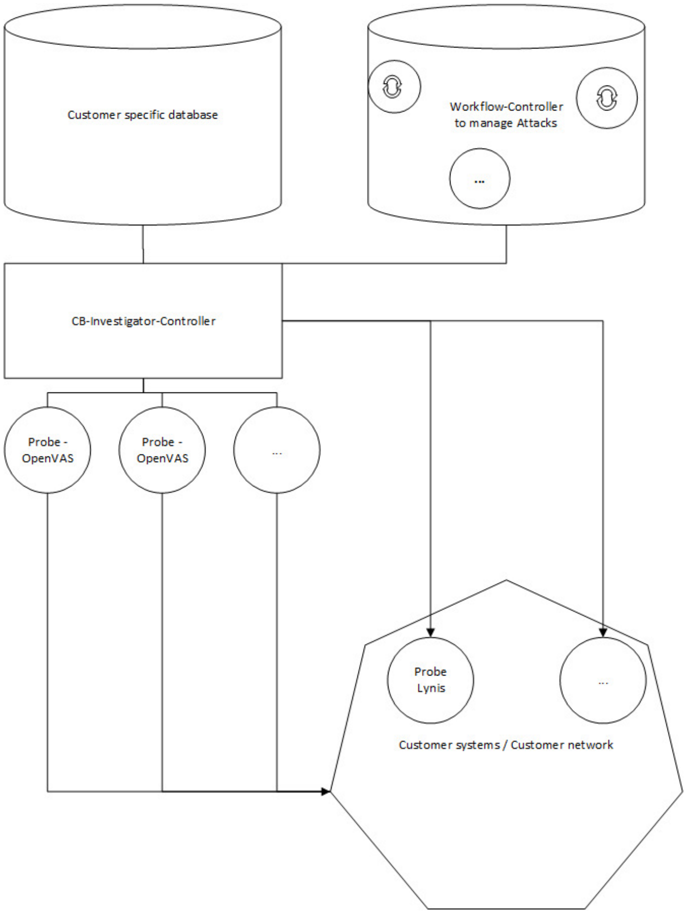
Customer specific database
Workflow-Controller to manage Attacks
...
CB-Investigator-Controller
Probe - OpenVAS
Probe - OpenVAS
...
Probe Lynis
...
Customer systems / Customer network

Figure 3. *Overall structure of Copra Breeder Investigator.*

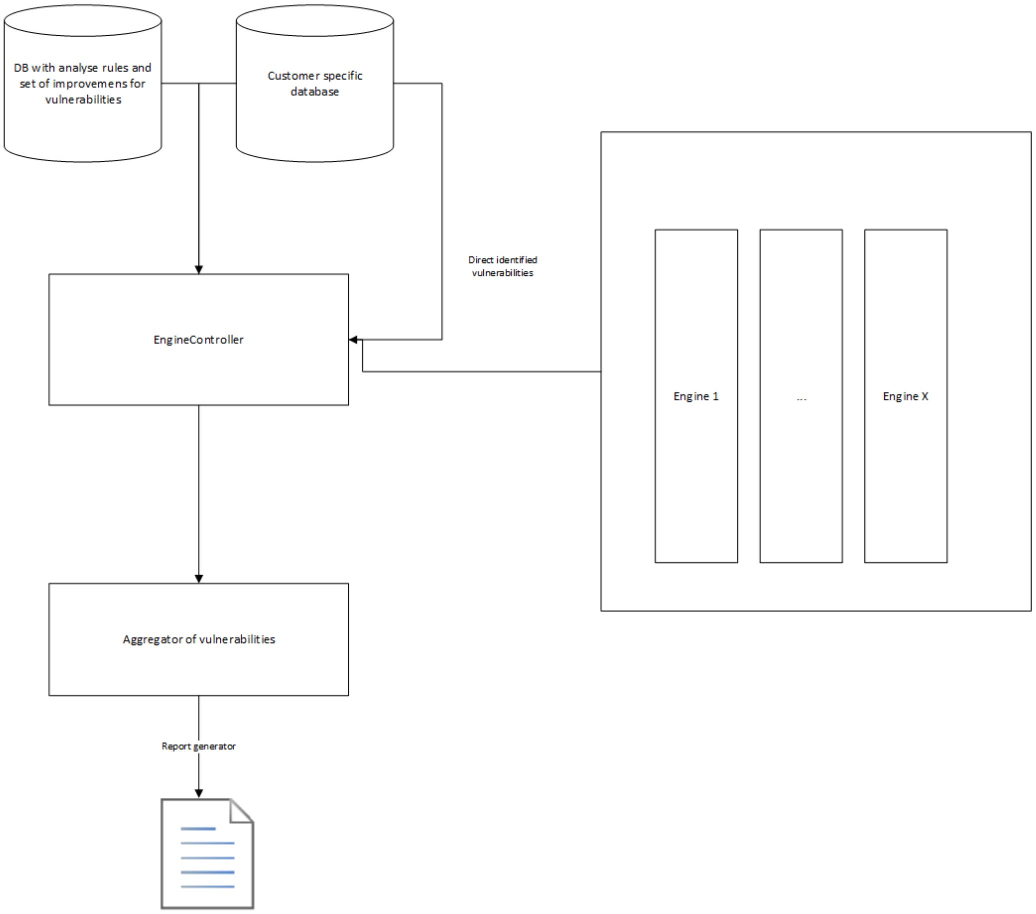
DB with analyse rules and set of improvemens for vulnerabilities
Customer specific database
Direct identified vulnerabilities
EngineController
Engine 1
...
Engine X
Aggregator of vulnerabilities
Report generator

Figure 4. *Overall structure of Copra Breeder Analyzer.*

nal security report. The CB analyzer will have several analysis engines. In these engines different analysis algorithms are implemented. When reporting all available data is distributed in these engines and analyzed. Some of these engines are:

- pattern-matching analysis,
- heuristic analysis,
- logical Analysis.

As a result, thereby further vulnerabilities are found and can be included in the report.

Application Example of Copra Breeder

In the following some application scenarios of Copra Breeder are presented.

Example of network penetration testing

Networks are one of the most common weaknesses in the company. The the following it will be presented how the Copra Breeder would act during the study of the most common network attacks, the ARP spoofing.

Fig. 5 shows a variant of an experimental setup. In a subnet, there are 3 computers which are connected via a sensor with the Copra Breeder Investigator. For this example, the following assumptions were made: Sensor 1 simulates the attacker and sensor 2 and 3 are installed on the systems that are attacked. Then the sensor 1 will initialize an ARP attack on the systems of sensors 2 and 3.

After a few seconds the ARP cache is then transmitted and checked whether manipulation of the MAC address exists. In addition, a defined message from sensor 2 is sent to sensor 3. Sensor 1, in the event of a successful ARP attack, would be able to read the message in its traffic.

If successfully exploited the vulnerability is introduced to the corporate network security report.

If the attack is successful, questions will be deposited under the corresponding implementation ID. These could be, for example:

- Could the attack be detected by an intrusion detection system (IDS) or intrusion prevention system (IPS)?
- Have other systems been affected by this attack?

If one of the sensors, such as sensor 2, discovered ARP spoofing defense programs such as ARP Watch, more questions will be added:

- Did the installed ARP Watch program respond to the attack?
- Is ARP Watch configured in such a way that qualified personnel is informed of the attempted attack?

Example of system tests

In reviewing the individual companies the various systems must be tested for vulnerabilities. Mostly, due to the large amount of systems (such as client PCs), only a certain range of systems is checked. In this case, however, some differently configured systems can be overlooked.

The CB Investigator allows a central roll-out of programs (the distribution of the test sensors) in order to achieve a scan of all existing systems. At the start up the sensors automatically login at the CB Investigator.

The CB Investigator will provide various tools for the examination. For Unix / Linux systems, for example, the scan tool Lynis can be used.

Lynis checks for typical configuration errors and provides suggestions for improving and hardening of the system ready. Typical tests are, for example:

- correct configuration of the authentication,
- settings of certificates,
- settings of databases,
- assignments of passwords and password policies.

Example of vulnerability tests

The most common attacks on companies are done via the network and via outdated or poorly configured networks. These attacks are caused by incorrect or improper maintenance.

To cover such weaknesses, so-called vulnerability scanners are developed that include various detection routines so that well-known and frequently-occurring vulnerabilities can be detected. Some vulnerability scanners even provide a solution description of the problem found.

One of the most popular open-source vulnerability scanner is OpenVAS. For example, OpenVAS is integrated over a Copra Breeder sensor and can be used easily.

OpenVAS allows the vulnerability analysis of XSAD known vulnerabilities. Due to the structure of OpenVAS these tests count as vulnerability testing. Because of these and other tools Copra Breeder is able to perform vulnerability scans without the typical negative effects of penetration tests, such as reducing the availability and risk of system failures.

Typical vulnerability tests in this context are:

- detection of weak passwords,
- Testing of known problems in configurations
- detection of outdated versions,
- detection of open ports,
- detection of unused or unnecessary services,
- free access to shares.

These vulnerability scans are often enough for many small- and medium-sized enterprises to perform a preliminary security analysis and to obtain meaningful information about the IT security status of the company.

Results of the analysis with Copra Breeder

The components CB Investigator and CB Checker will collect only data about a company and report only obvious vulnerabilities to CB Central, such as an outdated version. Thereafter, by means of CB Analyzer a comprehensive analysis of all data can be carried out and a complete report can be generated.

The report evaluation is carried out in several stages. In the first stage, all data is pushed in all the existing recognition engines. These recognition engines recognize patterns or heuristics due to possible weaknesses and across the systems of customers.

The recognition engines can recognize this fact and then give the previously detected vulnerabilities a higher priority.

Depending on the implementation of the engines they have different success rates. While patterns are usually more reliable

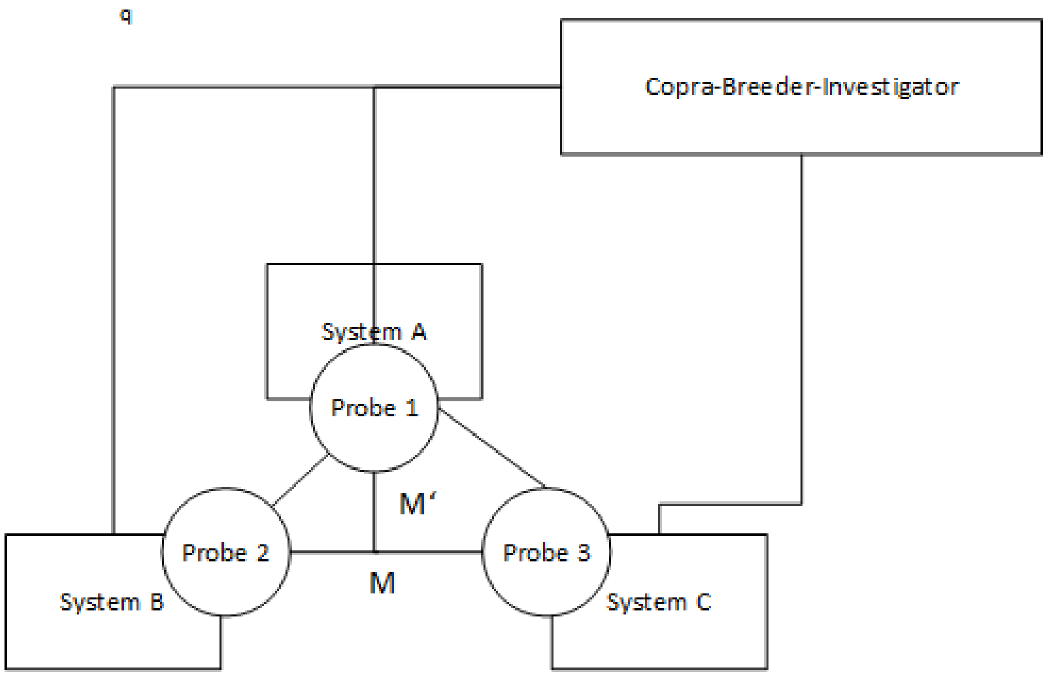

Figure 5. Representation of network penetration testing.

Weakness / Vulnerability	Details
Sendmail version has a remote buffer overflow	service port: smtp (25/tcp)
Sendmail-Version Zertifikat-Schwachstelle	service port: smtp (25/tcp)
TFTP directory traversal	service port: tftp (69/udp)

Table 1: Example table for examination result

than heuristics they are valued at a higher “reliability” in recognition. In the case of overlaying of two detections of various engines to a possible vulnerability, this vulnerability is assigned a higher reliability. This matching corresponds to level 2.

In stage 3, the results of the automated tests are compared with the entries of the user. Then, it will be tried to identify inconsistencies within the security statements and the Copra Breeder user will be informed about it. The user can then plan a way forward with the customer.

In the end it is up to the testers and project managers to decide whether the weaknesses in the final report be noted. In stage 4, all deficiencies are evaluated manually and included in the final report. Pre-sorting is possible over a threshold level of reliability.

Then, a final report can be generated from the identified vulnerabilities and weaknesses. The report contains:

- template for executive summary,
- template for summary of investigations,
- details of the investigations,
- list of vulnerabilities per server system,
- details of all vulnerabilities along with recommended solution

The final report is generated in a custom-ready format. As a Word-based format was used, for example, executive summary can be added with little effort.

An example of the presentation of the server vulnerabilities are shown in Table 1. For each system a new item and a new table is added:

1.1.1.1 Server - beispiel.assecor.de (192.168.123.123)
level of danger: high
test date: 07/07/2014

Finally, “tickets” are added for all vulnerabilities. These are structured as shown in Table 2.

They always include a title of vulnerability, a description, a prioritization and a proposed solution. More details as affected server and place of vulnerabilities can be added depending on the type of vulnerability. The texts can be displayed in any language.

Challenge in the Development and Limits of Copra Breeder

The Copra Breeder has some limitations that can lead to a manipulation of the result in its future development. The Copra breeder is dependent on correct inputs. Attempts at deception are difficult to identify and can be difficult to compensate. Even prior security measures, such as “security through obscurity” may confuse the Copra Breeder. The change of version numbers and welcome messages of programs can lead to false identification of a version. These false messages can lead to erroneous prediction of vulnerabilities. In particular, when only a vulnerability assessment without subsequent penetration tests is performed, this can lead to serious errors.

Also it is not possible to fully analyze own developments and

TFTP directory traversal
Level: High **CVE:** CVE-1999-0498, CVE-1999-0183 **Description:** TFTP configured in such a way that downloads can be initiated without authentication.
Proposed solution: Configure the system correctly. For this, the configuration file must be adapted to the service in general.
Affected server: 192.168.123.123 (beispiel.assecor.de)

Table 2: Example table for vulnerability description

unknown software.

By means of some sensors source code analyzes can be performed, for example, by means of spiders systematic studies on typical static or vulnerabilities can be made. Using spider and source code analysis, for example, the following weaknesses are recognized:

- SQL injection,
- cross-site scripting,
- OS injection,
- buffer overflow,
- manipulation of the update mechanism,

This makes it possible to identify in advance a lot of vulnerabilities that leads an individual analysis to a more substantial reduction of the expenditure.

Summary and Discussion of Results

Copra Breeder is planned as a software, which significantly reduces the cost of the security investigation. This Copra Breeder improves with each use, as the developed workflows can be reused.

Thus, it can reduce more and more the effort for further analysis.. Due to the almost complete automation capabilities of Copra Breeder this can be used in different scenarios, for example as:

- durable checking tool,
- one-time verification of companies,
- examination on request.

Copra Breeder as a permanent testing tool is capable to analyze a network in configured intervals in fully automated mode. It is possible to determine whether changes to the network lead to harmful behaviors or new vulnerabilities. The regular updates and improvements of Copra Breeder will improve the results over time. The Copra Breeder runs it in save-mode, so that only tests are run that do not lead to failure of the network.

Of course one can also start Copra Breeder at the request of the company, the first variant. The test level can be set individually so that even more intensive tests are run. By gaining more knowledge about the company and one will get more detailed information about vulnerabilities by Copra Breeder.

A one-time scan with Copra Breeder is performed after consultation with the company by an expert locally or remotely and is controlled. By adapting the workflow, a company can also be checked on all individual characteristics in order to create a comprehensive test result.

The Copra Breeder is being tested and designed from the beginning to own security. Data are obtained by Copra Breeder is safely and reliably transmitted and protected with the highest level of security against unauthorized access.

Acknowledgments

The authors wish to thank ....

References

[1] Microsoft Corporation: Windows Phone 8 Security Overview. Okt. 2012.

[2] Lucas Davi et.al: Privilege Escalation Attacks on Android. Boca Raton, Florida, Okt. 2010.

[3] Himanshu Dwivedi, Chris Clark und David Thiel: Mobile Application Security. McGraw-Hill Osborne Media, Jan. 2010. ISBN: 9780071633567.

[4] Pete Herzog: OSSTMM 3 – The Open Source Security Testing Methodology Manual. ISECOM, Dez. 2010.

[5] NIST and Emmanuel Aroms: NIST Special Publication 800-115 Technical Guide to Information Security Testing and Assessment. CreateSpace Independent Publishing Platform, Feb. 2012. ISBN: 9781470140427.

[6] TJ O'Connor: Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers. 1. Aufl. Syngress, Nov. 2012. ISBN: 9781597499576.

[7] Ryan Russell: Hack Proofing Your Network (Syngress). Syngress, Jan. 2000. ISBN: 9781928994152.

[8] Asaf Shabtai et. al: Google Android: A State-of-the-Art Review of Security Mechanisms. Department of Information Systems Engineering, Ben-Gurion University, Israel. Department of Computer Science, Ben-Gurion University, Israel. Deutsche Telekom Laboratories at Ben-Gurion University, Israel., Dez. 2009.

[9] Bundesamt für Sicherheit in der Informationstechnik: Durchführungskonzept für Penetrationstests. 1. Aufl. Bundesamt für Sicherheit in der Informationstechnik, Nov. 2003.

[10] Jeff Six: Application Security for the Android Platform: Processes, Permissions, and Other Safeguards. Oreilly & Associates Inc, Dez. 2011. ISBN: 9781449315078.

[11] Rick Hayes: PTES Technical Guidelines. `http://www.pentest standard.org/index.php/PTES-Technical-Guidelines`

[12] Google Inc. Android Security Overview. `http://source.android.com/tech/security/`

[13] Dave Kennedy and Iftach Ian Amit: High Level Organization of the Standard. `http://www.pentest-standard.org/index.php/Main-Page`