SIRIOS
A Framework for CERTs

Thomas Klingmüller
Federal Office for Information Security (BSI)
section CERT-Bund

May 15, 2005

Abstract

With the project SIRIOS 'CERT-Bund' [1] developed an open source framework for tools and workflows specifically in use within CERT-Bund. But this flexibility gives CERT-Bund the ability to implement its internal workflows in the framework so that they can be edited, logged and optimised.

The system and its databases can be implemented as classic client/server architecture in a closed environment (Intranet). Alternatively it can be set up to as a decentralised open framework with distributed databases and systems working together.
SIRIOS simplifies exchanging incident or vulnerability information between CERTs easy: SIRIOS's internal data structures got derived from international acknowledged data formats such as 'IODEF' [2] for incident information and 'EISPP' [3]/'DAF' [4] for advisory/vulnerability information. As a result, the formats for exporting these objects are well defined and can be used by any CERT regardless of the usage of SIRIOS.

SIRIOS got developed whilst CERT-Bund was still setting up. In early 2002, when CERT-Bund became operational, a trouble ticket system to structure and to log CERT-Bund's workflows was missing. An analysis of tools and workflows in other CERT environments revealed that many CERTs used tools developed on their own. Such toolboxes consisted of Office components for writing advisories and several task-specific tools. As a consequence out of missing standards for CERT specific information and tools that implemented these standards, information sharing was reduced to individual CERTs defining their information interfaces and methods to import/export these information.

With SIRIOS CERT-Bund implemented a framework to control internal workflows and to support the exchange of CERT specific information objects. This Paper explains SIRIOS's functions and the database structures in use. To illustrate these abstract formats, we will use CERT-Bund's role based model and its workflows as an example. Furthermore, we will discuss shortly the possibility of international cooperation between CERTs using decentralised implementations of SIRIOS.
The development of SIRIOS already started during the set-up phase of CERT-Bund.

1 Introduction and Background

SIRIOS is designed as CERT-Bund's internal framework for processing and managing all CSIRT related information and data. The main benefits for CSIRTs using this framework are:

- Processing incoming (security related) information,
- Open source tool, with possibility of customisation and
- The possibility to exchange essential security related data with other CSIRTs according to common standards.

SIRIOS combines the management of security incidents, sources of information and contact details /address data. The processing of this data can be fully adapted to every team's own needs including procedures for quality of service through the possibility to define workflows individually. With the built-in role-based workflow management you can define who can access the data and what actions may be taken upon it. Furthermore, you have the ability to assign roles to users.

The architecture of the framework is modular, similar to the Apache web server. The basic component is the kernel which is enhanced by a set of modules. The framework comes with a set of modules but can be accustomed to use individual modules. Each module enhances the kernel with new functionality.

2 The Scope of the Project

Resulting from the need to optimize our internal procedures and to assure a certain level of quality, we at CERT-Bund realised that we were in need of a centralised, computer based system to keep track of open requests and incidents. While structuring and defining the project several additional options of such a system became quite clear.

The project's primal goal is the implementation/optimisation of iternal workflows and the aggregation of tools relevant for CSIRTs in one framework. Through its architecture with a kernel and the possibility to add enhancing modules - similar to the Apache web server - each team will be able to adapt the SIRIOS system according to their own requirements. Most CERT-specific tasks are already implemented in the modules that are provided with the SIRIOS-project. Let's have a look at the system's main features:

- The system's internal representation of all information is a 'ticket'. Each ticket has a set of attibutes, among which e.g. is its history. Most of the system's functions rely on these attributes.
- A complex role model enables the implementation of a fine-grained user rights management policy for the different information classes within SIRIOS.
- Tickets are collected in queues, which can be structured three levels deep. This is the base of the workflow-implementation.
- SIROS is mostly platform independent and uses e-mail for data exchange between other systems. Clients may access the system via e-mail or the web based interface.

- Each modification of a ticket is logged in it's own history data attribute. This information can only be accessed by autorised roles. The system keeps track of outgoing mail as well and is able to assign incoming mails to information/tickets send by the system itself (if specific mail information didn't change).
- Within the system several accepted standards such as EISPP/DAF for advisory-exchange between CSIRTs or IODEF for exchanging incident information were implemented.

By distributing the SIRIOS framework we hope to encourage german and international teams to use the tool and to expand the information sharing process between CSIRTs.

3 Flexible independent Framework for CSIRTs

SIRIOS with its web-based interface provides a central user rights management for access control. Each instance such as a user or a role can get a fine grained assignment of functionality and resources. This and the flexible definition of workflows allows to map all CSIRT procedures in the framework.

Each work step is implemented as a separate queue while the sequential execution of several steps represents a workflow. Passing through the workflow tickets are sent from queue to queue. Terminating a work step the editor moves the ticket in one of the pre-defined following queues. These transactions pass allowance checks by the system kernel that controls all of the system's activities. A list of all the workflows accessible to an editor can be accessed through a web site which shows the tickets in use by the editor.

Each event is mapped internally to a ticket. While executing the workflow a ticket's content may be changed or enhanced, attributes may get changed or other data appended to that ticket. Every ticket access is watched by the role based access control. Up until a ticket is closed all allowed users/editors can access and/or modify a ticket (according to their rights). Afterwards it gets marked as 'closed' in the database and only an advanced search can reopen it again. Deletion of tickets is solely reserved for the system administrator.

As SIRIOS is based upon the trouble ticket system beneath, we have another important feature is available: the ticket's escalation. This escalation is triggered by the ticket handling of the kernel and can be configured by several definable criteria to urge the user to work on the ticket. While a ticket is in escalation mode, no other ticket may be accessed while the escalation endures. Even system functions could be invoked in escalation mode to perfom any desired action. This functionality increases our service level and guarantees a quality of all processes.

SIRIOS is based on the trouble ticket system 'OTRS' [6] which controls all the ticket handling. Additional functionality is provided through separate modules. These can be accessed through the kernel all the time, while the user's rightsmanagment decides if he/she may or may not use them. Modules can be directly accessed by a user, such as an additional functionality like the vulnerability database (accessible by an icon on the workplace) or integrated in such a way that the user does not realize he just invoked a modules funktion.

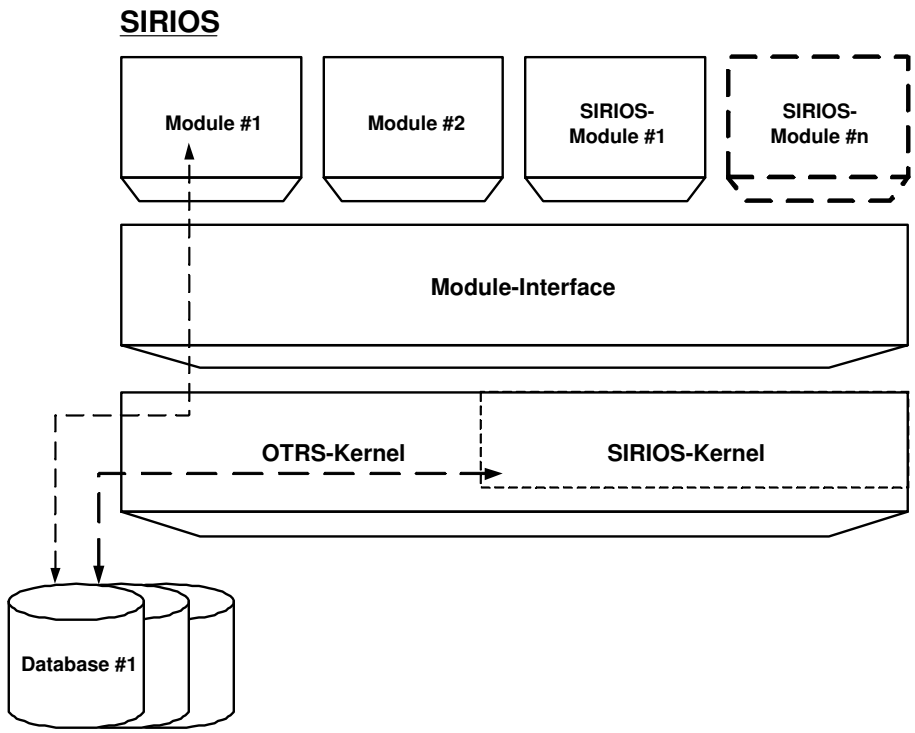

Figure 1: SIRIOS - Layer

By the time of this writing modules for the following tasks are ready for use:

- Incident tracking
- Writing of advisories
- Import and export of information using supported standards
- Checking signatures, encryption, decryption
- Vulnerability database
- Artefact database
- Contact database
- Monitoring of web sites
- Graphical user interface for the administrator
- Multilanguage support
- Paket-manager

Inter-modular communication is controlled through the system's kernel.

4 Incident Tracking

One of the most complex modules is the module for incident tracking. To demonstrate this, let's take a look at the following workflow: The hotliner receives the incoming information and opens a ticket in the framework. This ticket gets delivered to the coordinator who decides who's going to get the ticket next, in our case the incident handler. The incident handler receives the ticket and opens the incident tracking module to create a new incident object in the incident database. Then the incident handler fills in the information from the ticket into the incident object. The form used to fill in the information can be configured to contain fields with pre-defined lists of values (e.g. 'DoS', 'DDoS', 'Worm' or 'TCP-Scan') which makes standardisation much easier.
Incident object input is structured into thematic blocks with several lists of predefined values (according to the IODEF standard). This way we can easily export the data into a XML file, which we can use to exchange incident data objects with other CSIRTs. After filling in the information, the incident object gets associated with the originating ticket and is ready for further processing.

There are several ways to export incident objects:

- anonymising the object: all identifying information gets deleted, or
- 'pseudonymising 'the object: substituting each unique identifying information in the hole object with a variable representing that information

The system proposes the questionable fields which might need a change, separating data of the attacker and data of the victim(s) by highlighting them in different colours. Additional functionality includes combining incident objects with other data objects (e.g. artefacts, vulnerabilities), incident statistics and a search function.

Importing IODEF is fully supported. Before importing the IODEF object into the system, the incident handler can check the data. If this check is passed, the object gets imported into the database. Objects from trusted sources can be configured to be imported automatically.

5 Advisory Handling

This central module is in its appearance very similar to the incident handling module. Input is grouped by thematic categories and expected to meet the EISPP/DAF standard. Advisories can be exported in these formats to exchange them with other CSIRTs.
The export function is implemented in a separate module, which will export the chosen object in the appropriate form (e.g. according to EISPP/DAF , incident objects according to IODEF). Furthermore, the module generates a special format for publication e.g. web portal and email.
The advisory module can be configured with templates for different types of advisories. CERT-Bund publishes three kinds of advisories (depending on constituency and criticality): short summaries on vulnerabilities (called 'Kurzinfo ') which meet the need for prompt warning, detailed advisories (called 'Advisories') and malware/virus information (called 'Virinfo '). Even administrative messages or any other kind of messages can be generated by using this module.

The system scans each incoming information for different types of content: in extent to try to check any digital signature for accuracy, the system tries to recognize any implemented standard (e.g. EISPP/DAF or IODEF) in the incoming message. Import/Export functionality is defined as an exchange of mail attached XML-files. If the system finds such data in the incoming mail, an privileged user has to authorize the import process. Certainly this imported data can be used to generate new tickets/messages/objects.

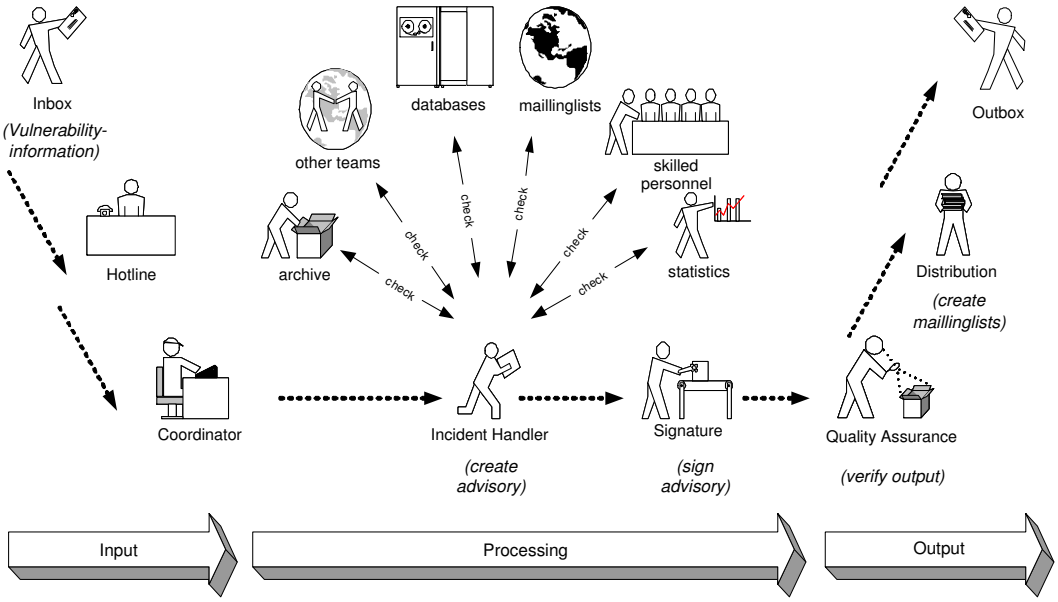

Figure 2: Workflow: Advisory Handling

As usual, search functionality, linking of objects (such as vulnerabilities to advisories or advisories to superseding advisories) and statistic functionality are implemented as well.

6 Cryptography

Several workflows are in need of cryptographic functionality:

- the advisory module for encryption / decryption and signing the messages/objects
- the kernel for checking signatures automatically
- the artefact database for generating/checking MD5-checksums

This functionality is provided by a cryptogrphic module which supports at the time of this writing the PGP/GnuPG and S/MIME standards.
To increase usability and to simplify the cryptographic functionality, the system hides many cryptographic tasks from the user - only asking for information when needed. If possible (corresponding key known to the system) SIRIOS checks signatures or decrypt messages automatically and writes the result into a ticket's attributes. This way an editor can see the cryptographic status of a ticket/data.
Even internal tickets get signed e.g. for administrative tickets that manage an advisory's publication (these tickets are signed for authentication and authorisation purposes). Besides this automatic functionality, the cryptographic functions can be invoked by the editor to sign or check, encrypt or decrypt cryptographic data.

7 Vulnerability information and artefacts

SIRIOS uses several databases for contact information, advisories and for vulnerability information and possible artifacts. To automate the gathering of vulnerability information two import filters were implemented, which can retrieve information from the 'CVE' [5] database and from the Open Source Vulnerability Data Base ('OSVDB' [7]). Artifacts, such as proof-of-concept exploits, are stored in a separate database. Additionally to the data itself, each artefact is associated with a cryptographic checksum (MD5) to identify its integrity.
The system only allows the download of the file onto the client's system. As in any module, each vulnerability information or artefact can be associated with any other object in the system (e.g. advisories, incident objects, contacts or even other artefacts and vulnerabilities). These associations can be found in the attributes of the corresponding object. This allows the editor to get an overview of the object and its associations in the desktop representation.

8 Packetmanagement

SIRIOS is absed upon the "OTRS 2.0" trouble ticket system. Modifications can be applied by editing the SIRIOS-specific configuration file and through the SIRIOS-application itself in the administrator's environment. In this environment can be specify which modules shall be installed by checking the appropriate check-box. These modules could be located locally for installation or loaded from the Online-Repository. Once the source server is chosen from the repository SIRIOS checks the versions of installed modules lists all newer or not yet installed modules that could be chosen for installation. Through this centralised module-management we have the ability to provide an easy way to install and update modules.

Several functions implemented in SIRIOS require a normed naming for operating systems and applications. This is implemented in the so called 'Model for System Information' [8] defined in EISPP/DAF. For this information is quite essential when exchanging information among CSIRT-teams the model is in continual progress, so that all new relevant information is included. To make the process of keeping this model up-to-date within SIRIOS, it can be installed as a seperate module through the packet-manager. Updates to this module will be made available on a regular basis.

9 Multilanguage template based

SIRIOS implements multilanguage features in two aspects:
The system uses a wide range of templates to display information to the user. Depending on the user's browser settings and his choice before logging-in the templates will appear in english or in german.
Language support is important in the way modules get fed with information. E.g. the advisory module supports the creation of multilanguage-advisories: the information can be entered for two supported languages at a time (meanwhile english and german). To be able to specify the desired language for an advisory all fields have to be filled in. When exporting an advisory all supported languages will be included within the XML-object. Though up to now there is only support for german and english, further language-templates can be implemented easily.

10 Summing-Up

The CERT-Bund SIRIOS has the following aims:

- Guarantee internal standardised workflows
- Guarantee quality of service processes in the workflows
- Monitoring and controlling of working steps
- Establishing a platform with the ability to exchange computer security related information within the CSIRT community
- Centralised flow of information and work sequences

11 Workinggroup: SIRIOS

With release 1.0 SIRIOS is free to use for all CSIRTs (regarding the GPL). Planned is to constitute a SIRIOS working group which should have the purpose to:

- assure the improvement of quality and the development of further modules and to
- encourage and establish the exchange of information between teams.

Up to now, round about 10 teams in the european region use and test the SIRIOS framework. Feedback shows a growing interest in the system and its open source status. So in case of widespread use we might see "SIRIOS-based" networks and an extended information sharing process. For this to come true a joint development of the kernel and new modules is required.

CERT-Bund encourages interested teams to test and implement SIRIOS according to their needs and to help evolve SIRIOS in future. In cooperation with these teams we will develop additional modules and enhance the functionality to improve information sharing between CSIRTs.

There's a common goal for all CSIRTs:

'*Having all relevant information regarding an issue as soon as possible.*'

The SIRIOS framework with all released modules can be downloaded from the Internet.

References

[1] Federal Office for Information Security, Bonn Germany- *CERT-Bund*
www.bsi.de

[2] Incident Object Description and Exchange Format - IODEF
http://www.ietf.org/html.charters/inch-charter.html

[3] Common Advisory Format Description, v2.0. - EISPP
http://www.eispp.org

[4] Advisory Format Description - DAF
http://www.cert-verbund.de/daf/daf_description.html

[5] Common Vulnerabilities and Exposures - CVE
http://cve.mitre.org/cve/

[6] Open source Ticket Request System - OTRS
http://otrs.org/

[7] Open Source Vulnerability Data Base - OSVDB
http://www.osvdb.org/

[8] Bernd Grobauer, Siemens CERT: Towards a Common Model of System Information
www.cert-verbund.de/daf