Automated resolving of security incidents as a key mechanism to fight massive infections of malicious software

Jochen Kaiser[1], Alexander Vitzthum[1], Peter Holleczek[1], Falko Dressler[2]

[1] Regionales Rechenzentrum Erlangen, University of Erlangen, Germany
{kaiser,alexander.vitzthum,holleczek}@rrze.uni-erlangen.de
[2] Dept. of Computer Science 7, University of Erlangen, Germany
dressler@informatik.uni-erlangen.de

Abstract – Today, many end systems are infected with malicious software (malware). Often, infections will last for a long time due to missing (automated) detection or insufficient user knowledge. Even large organizations usually do not have the necessary security staff to handle all affected computers. Obviously, automated infections with malicious software cannot be handled by manual repair; new approaches are needed. One way to encounter automatic mass infections is to semi-automate the incident management. Less important security incidents should be handled by the user himself while serious incidents should be forwarded to qualified personal. To enable the end user resolving his own security incidents, both organizational and technical information have to be provided in a comprehensible way. This paper describes PRISM (Portal for Reporting Incidents and Solution Management), which consists of several components addressing the goal: a unit receiving security incidents in the IDMEF format, a component containing the logic for handling security incidents and corresponding remedies, and a component generating dynamic web pages presenting adequate solutions for recorded security incidents. PRISM was verified using case studies for universities, companies and end-user/provider scenarios.

Keywords – Incident Management, Malicious Software, Massive infections

1 Introduction

In today's networks, malicious software is a realistic threat. There are many possible ways for infecting network components. A typical example is the use of mobile clients from nomadic users. These mobile clients can become infected during visits in unprotected networks. Afterwards, such systems are again brought to the protected network and it can infect other hosts. Another way for infecting systems by malware is the use of application layer transport mechanisms like web and email. Users often click on interesting content without checking the origin of the information first. Even Internet protection mechanism like firewalls cannot guarantee that a network is not infected with malicious software. This is due to the fact that in most cases application protocols and spreading mechanisms of malware use the identical Internet TCP/UDP resources. The massive increase of security incidents results in more work for CSIRTs (Computer Security Incident Response Team) and may lead to overstrain them in handling the incidents. Then, the CSIRT is often more involved in administrating low level incidents than in searching causes and circumstances of high level attacks. To compensate these negative side effects, new methods have to be developed so that a CSIRT can again concentrate on the most relevant computer security incidents. Typical systems that are used by CSIRTs to administrate security incidents like RT (Request Tracker) or OTRS (Open source Ticket Request System) are not offering the needed support. In this paper we depict an alternative to manual resolving of security incidents. The developed program PRISM (Portal for Reporting Incidents and Solution Management) is introduced and discussed.

2 Incident Management

To understand the functioning of an incident management system it is necessary to have a deeper insight in the differences to a standard helpdesk system. Such helpdesk systems usually implement a queue based administration of helpdesk cases. On the other hand, an incident management system needs further functionality like a role model and an escalation model. Additional tools enable a more efficient operation.

2.1 Unqualified or coincidently aimed security incidents

Nowadays, many computer systems are attacked and compromised in an automated way. Shortly after a new security threat is discovered, it will be exploited by quickly distributed programs. Mostly, the usage of such programs is not aimed at specific targets. Instead, automated scans try to detect systems that can possibly be infected and then an attack is initiated. Such attacks are more annoying than being a critical security incident. Of course, there are exceptions to this rule when systems are affected that have either a critical role or function, e.g. systems in a medical context.

2.2 Well aimed or qualified security incidents

On the other side, qualified security incidents are well-aimed security attacks to specific systems. They always follow a specific reason and can harm critical damage to a company or infrastructure. The benefit of distinguishing between those attack types is that a CSIRT can assign priorities to different incidents. However, even if it becomes possible to direct the attention to the most critical security incidents, it is not recommended to ignore massive infections as discussed in the introduction. To counteract the huge number of massive infections, the aid of a rule-based management is needed. Such a system should manage security incidents and provide technical help for approaching the incidents. Additionally, they should offer an interface to firewall systems so that the isolation of a compromised system becomes possible.

2.3 Incident management systems, workflows, and tools

The approach is to create tools and workflows to handle the incidents in a way that involves as less human resources as possible. Thus, an analysis of the processes accompanied by an incident lifecycle is needed supporting the following processes: detecting and reporting incidents, finding the responsible persons for an incident, solution finding, resolving, and post-resolving (taking care that an incident is not repeated). The above mentioned processes are the basis for our implementation of an incident management system.

2.4 Basic functions of an incident management

An incident management system should consist of a portal that presents the incidents and offers additional management functions. It should have a generic interface to accept incident reports from different sensors. Finally, a function should be available to review an incident by the user of the infected system. Such a self-resolving terminal must be accessible via WWW and offer functions for the user to learn about the incident, to remove its cause, and to set the state to “closed”.

2.5 Role model

There should be a role model which distinguishes between different persons:

1. The end user, who is the regular user of the PC
2. The computer/network administrator of the organizational unit or department
3. The CSIRT admin, who is highly qualified

2.6 Escalation and delegation model

To maximize the efficiency in managing security incidents a delegation model is needed. It is necessary to allow a CSIRT to apply priority classes to the incidents. An incident starts with a low priority. The status increases when the security incident can not be handled by untrained personnel in a dedicated time.

Security incidents are divided into two classes. The first class comprises the incidents, which are more an *annoyance* for an organization. To go in more detail, these incidents are coming from massive infections with malicious software. The second class includes *higher risk* attacks as discussed previously. The first class consists of three escalation levels describing their security context:

- Class 1 - Level 1: security incidents having a low risk to the organization
- Class 1 - Level 2: the end user was not able to solve the problem himself and now the responsible computer administrator has to clear the problem
- Class 1 - Level 3: the computer administrator cannot fix the level 2 problem, thus a CSIRT administrator must supervise the incident
- Class 2: These incidents have a significant impact on the organization. They will never be in the scope of an end user and must be solved by the CIRT team.

Of course there are always exceptions to the above scheme when a critical system is infected or the impact of the case is categorized larger then “annoying”. Then the entrance level of the incident will be higher in the same class.

2.7 Solution management processes

An incident management system should be able to administrate the security incidents in form of a help desk system. It should have views where security incidents can get a priority and where additional actions to isolate a system may be taken. The security administrator should get support to find often needed information on typical incident so that these incidents very easily can be resolved.

2.8 Workflow

An exemplary incident helps to understand the workflow of the system. It is assumed that a sensor detects a security problem and reports it to the management system which starts to handle the incident.

Workflow for the Security Admin

A CSIRT member logs into the system and find a security incident. He has a brief look on it and selects the priority class of the security incident and additional sanctions like isolating or blocking the system. The CSIRT member then clicks the button to preview the WWW page which is generated for exactly this incident. The admin now sees a text the incident management proposes. The text page contains the following text modules: a classification of the security incident, an explanation, helpful links, links for removal tools/patches, and threat of punishment (i.e. escalation process).

The security admin can accept this proposal, change it, or generate a new one. A newly generated incident prototype can be stored in a database for further use. Additionally to storing the incident and the remedy page in the database, the user is informed vie electronic mail that an incident exists and requested to use the incident management WWW portal.

Workflow for the user

When a user wants to access the Internet, usually a WWW page is requested. Since the system is isolated or blocked, the request is re-routed to the incident management system user information terminal. The incident management system detects the source IP address of the user and shows the security incident(s) relevant for this system. The user sees a specifically generated page delivering information on the incident, the cure, and following steps. The user can now use the information and clean the PC.

If the attempt to remove the cause for the security incident was successful, the user can again access the incident management system and set the security incident to the resolved state. The incident management system then releases the block or isolation of the PC and the user has full Internet access again.

Escalation methods

If the removal of the cause for the incident was not successful, the affected system will again trigger the security sensors and generate a security incident. The incident management system will assign the new security incident to the former existing one and propose to increase the escalation level.

This will activate the above mentioned processes but this time the user is not the one to resolve this security incident. When the user connects to the incident management system there will be an information page which tells him that the escalation level has been increased and that the responsible administrator of the department has to login to resolve the security issue. The administrator will receive an electronic mail that informs about a problem with one of the systems in his area of competence.

Role of the regional administrator

The Administrator of the affected PC now can login to the management system and finds a WWW page showing all security incidents in his area of responsibility. He can access the security incident and gets the same page as presented to users in the previous level one. So he gets the same hints to solve the problem as the user. After he has solved the problem, he can set this incident to the state “solved”.

2.9 Additional Tools

The following additional tools are needed to administrate incidents:

- Host isolation /quarantine networks: A mechanism is needed to isolate infected hosts. This may be done by manipulating firewalls/routers or using quarantine networks which are dynamically configured in the access layer of a campus network.
- Knowledge about the administrative situation: To reduce work for the CSIRT team, it is necessary to know exactly which administrator is responsible for the affected part of the network. For this task, profound knowledge of the administrative background of the network is important. Often the responsibility for a host is not clear. In the worst case, a security incident is delegated to a system administrator who is not responsible for the affected host and, therefore, refuses to cooperate. Therefore, tool support must be provided to track changing responsibilities.
- Update Networks: When there is a mechanism to isolate/block systems which have a security incident it should be easy for the system administrator to get the relevant antiviral computer tools or security patches to clean the system and remove any malicious programs and additionally to actualize the system with the missing patches.

2.10 Related work

Most CSIRT teams use helpdesk systems to track their incidents. Sometimes, the helpdesk systems are modified to better meet the needs of a security incident handling team. Well known help desk systems are the Request Tracker (RT) [BP1] and the Open Source Ticket Request System (OTRS) [OT1]. There is a RT extension available called Request Tracker for Incident Response (RTIR) [BP2], which mainly concentrates on incident handling and tracking. There are no strategies included which may help scoring an incident. More interesting is the SIRIOS [SI1] extension to OTRS. The extension includes features like a customer database, which provides correct contact information. Additionally, there are several modules available which allow the import of data from vulnerability databases or categorizing IT-products.

3 PRISM

In this section, we describe our own incident management system PRISM (Portal for Reporting Incidents and Solution Management). Starting with an overview to the architecture and available sensors, the working principles are presented and discussed.

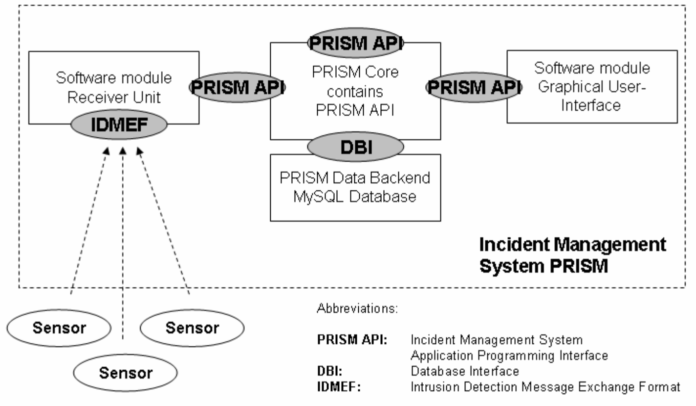


Fig. 1. Overview of the PRISM architecture

3.1 Architecture

The architecture of PRISM is shown in Fig. 1. The system was implemented in form of a WWW-portal using the PERL scripting language. An overview of the complex structure and the different programs are depicted in Fig. 2. PRISM consists of four different modules:

- The PRISM core organizes the internal workflow and the solution management.
- The receiver gets the sensor information in IDMEF (Intrusion Detection Message Exchange Format). It is connected to the PRISM Core via the PRISM API.
- The graphical user interface generates dynamic web pages for the incident administrator and the user web pages.
- The backend is realized by a MySQL database using a DBI interface.

3.2 Implemented sensors

The following sensors have been implemented. The sensors send their incident information via HTTP to a receiver unit of the PRISM system encoded in IDMEF:

- *Direct input* - It is possible to insert new security incidents via a WWW-interface. The user has to authenticate, so the origin of the incident message is known.
- *No DNS* - The traffic accounting may be correlated with the information in the DNS system. So it is possible to detect system not officially registered. In this way DNS registration policies can be easily achieved.
- *IDS input* - Another sensor is a snort IDS (intrusion detection system) that monitors all the traffic from and to the network. Since the snort IDS generates immense numbers of incidents, the snort IDS does not send IDMEF messages directly to the PRISM incident management. Instead, an aggregation process collecting all security incidents in a database and generating IDMEF events after having evaluated them was implemented. The aggregation configuration can be administrated via a WWW-Interface from the PRISM system.
- *Virus information input* - The antivirus client software is configured to send information to a central mail address when a virus is detected. A process retrieves this information and opens an incident case in the incident management system.

3.3 Solution management

The solution management is realized in the context of the procedures to deliver generated WWW pages for each incident. The incident administrator can assemble a page, which is later presented to a user accessing the self-resolving terminal WWW pages. There are two ways in which the PRISM system supports the CSIRT:

- *Prototype remedies* are offered containing information about previously seen incidents. The incident manager has full access to the remedies and can edit the categories and single remedies belonging to a category.
- *Virus pattern help* is a tool that tries to find accurate information about a malware infection on external pages of an antiviral software manufacturer. Therefore, the tool scans the WWW pages of the manufacturer and delivers basic information about the found malware like TCP/UDP resources and the malware name. The tool scans the results that are delivered to the search process and extracts descriptions and URLs which are then proposed to the incident administrator. If the incident administrator approves the information, they are included in the user's WWW portal page.

3.4 Operation of the PRISM tool

Fig. 3 shows the main page for the CSIRT admin. On this page, an overview of all incidents of the network is provided. Each line depicts a single incident: the first part is information about the incident and the sensor; the second one presents status information about the incident and allows interaction of the admin.

1. The first row lets the administrator decide whether a security incident may be self resolving or not. If it is activated, a user can access the incident management portal and manage the security incidents belonging to the scope. On deactivation, a CSIRT or subnet administrator must resolve the issue.
2. The second row is a status indicator and shows if the incident is already solved. The indicator is either set by the user when the incident is resolved via the incident self-resolving portal or by the security administrator,
3. The third row indicates the reactive security option. If set, the system sends a block host command to the FAUST system [FP06], which generates adequate firewall rules.
4. The fourth row is a mail symbol linking to the mail address of the responsible system or network administrator. The address is collected from an information system.
5. The last row offers to create notes about the incident. If saved, a note cannot be removed.

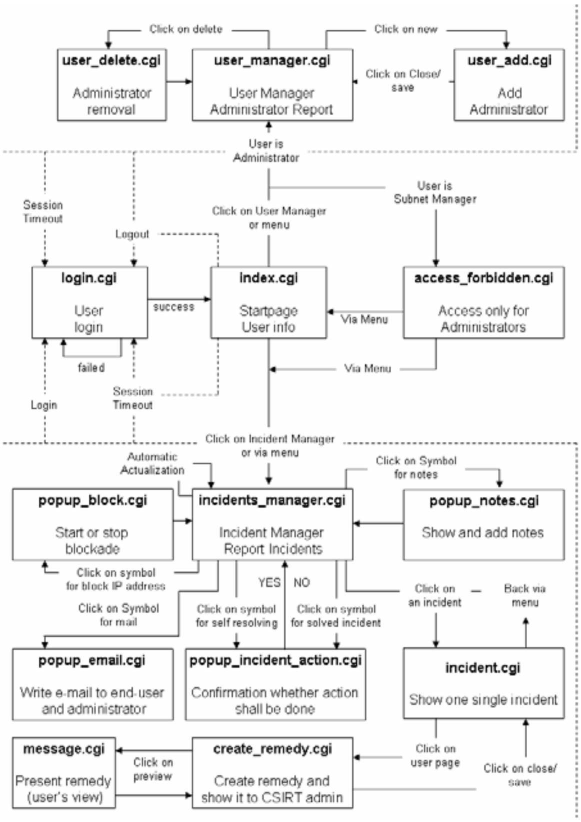
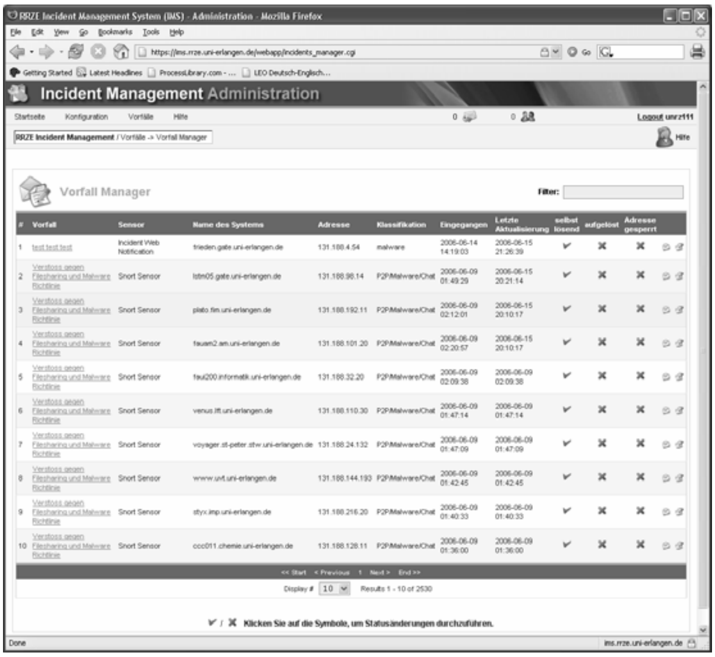


Fig. 2. Overview of the PRISM CGI scripts



RRZE Incident Management System (IMS) - Administration - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://ims.rrze.uni-erlangen.de/webapp/incidents_manager.cgi
Getting Started Latest Headlines ProcessLibrary.com - ... LEO Deutsch-Englisch...
Incident Management Administration
Startseite Konfiguration Vorfälle Hilfe
Logout unrz111
Hilfe
Done
ims.rrze.uni-erlangen.de

RRZE Incident Management / Vorfälle -> Vorfall Manager

Vorfall Manager

Filter:

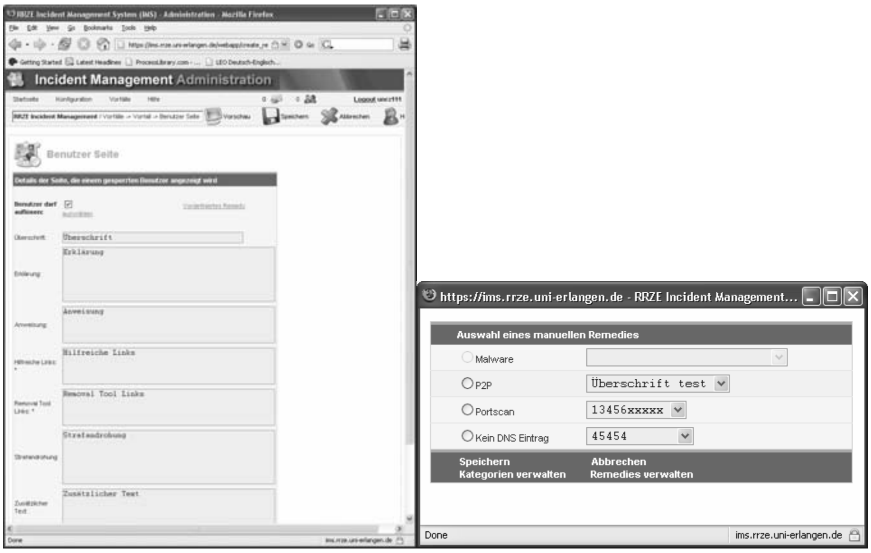
#	Vorfall	Sensor	Name des Systems	Adresse	Klassifikation	Eingegangen	Letzte Aktualisierung	selbst lösend	aufgelöst	Adresse gesperrt
1	test test test	Incident Web Notification	frieden.gate.uni-erlangen.de	131.188.4.54	malware	2006-06-14 14:19:03	2006-06-15 21:26:39	✓	✗	✗
2	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	lstm05.gate.uni-erlangen.de	131.188.98.14	P2P/Malware/Chat	2006-06-09 01:49:29	2006-06-15 20:21:14	✓	✗	✗
3	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	plato.fim.uni-erlangen.de	131.188.192.11	P2P/Malware/Chat	2006-06-09 02:12:01	2006-06-15 20:10:17	✓	✗	✗
4	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	fauam2.am.uni-erlangen.de	131.188.101.20	P2P/Malware/Chat	2006-06-09 02:20:57	2006-06-15 20:10:17	✓	✗	✗
5	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	faui200.informatik.uni-erlangen.de	131.188.32.20	P2P/Malware/Chat	2006-06-09 02:09:38	2006-06-09 02:09:38	✓	✗	✗
6	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	venus.lft.uni-erlangen.de	131.188.110.30	P2P/Malware/Chat	2006-06-09 01:47:14	2006-06-09 01:47:14	✓	✗	✗
7	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	voyager.st-peter.stw.uni-erlangen.de	131.188.24.132	P2P/Malware/Chat	2006-06-09 01:47:09	2006-06-09 01:47:09	✓	✗	✗
8	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	www.uvt.uni-erlangen.de	131.188.144.193	P2P/Malware/Chat	2006-06-09 01:42:45	2006-06-09 01:42:45	✓	✗	✗
9	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	styx.imp.uni-erlangen.de	131.188.216.20	P2P/Malware/Chat	2006-06-09 01:40:33	2006-06-09 01:40:33	✓	✗	✗
10	Verstoss gegen Filesharing und Malware Richtlinie	Snort Sensor	ccc011.chemie.uni-erlangen.de	131.188.128.11	P2P/Malware/Chat	2006-06-09 01:36:00	2006-06-09 01:36:00	✓	✗	✗

<< Start < Previous 1 Next > End >>

Display # 10 Results 1 - 10 of 2530

✓ / ✗ Klicken Sie auf die Symbole, um Statusänderungen durchzuführen.

Fig. 3. Administrator view of all incidents



RRZE Incident Management System (IMS) - Administration - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://ims.rrze.uni-erlangen.de/webapp/create_re
Getting Started Latest Headlines ProcessLibrary.com - ... LEO Deutsch-Englisch...
Incident Management Administration
Startseite Konfiguration Vorfälle Hilfe
Logout unrz111
Vorschau
Speichern
Abbrechen
Done
ims.rrze.uni-erlangen.de
https://ims.rrze.uni-erlangen.de - RRZE Incident Management...
Done
ims.rrze.uni-erlangen.de

RRZE Incident Management / Vorfälle -> Vorfall -> Benutzer Seite

Benutzer Seite

Details der Seite, die einem gesperrten Benutzer angezeigt wird

Benutzer darf aufheben: ☑

Überschrift: Überschrift

Erklärung: Erklärung

Anweisung: Anweisung

Hilfreiche Links: * Hilfreiche Links

Removal Tool Links: * Removal Tool Links

Strafandrohung: Strafandrohung

Zusätzlicher Text: Zusätzlicher Text

Auswahl eines manuellen Remedies

○ Malware	
○ P2P	Überschrift test
○ Portscan	13456xxxxx
○ Kein DNS Eintrag	45454

Speichern Abbrechen
Kategorien verwalten Remedies verwalten

Fig. 4. Assembling of an incident message (left) and solution management (right)

Fig. 4 (left) illustrates the assembly of an incident WWW page for the PC user. The incident management system proposes a very basic solution of an incident. The CSIRT admin can now edit each part of the page. The different fields offer text fields for a title to name the incident, a description, orders, explaining links, links for removal tools for malware, and escalation perspectives. If the incident manager is not satisfied by the proposal made by the incident management system, he can open a window, which offers several prototype remedies. There can be several remedies per incident class. If the administrator chooses one of them, the text fields are overwritten by the prototype. The administrator still can edit the text fields to match the needs of the current security incident. Using the "save" button, the content is stored in the database. The "preview" button allows reviewing the designed page.

Fig. 4 (right) shows a small window which is called remedy management (or solution management). The incident manager can create new categories for incident classes. Also, new instances of a category can be maintained. So it is possible to have a category "malware" and many instances describing different types of malware.

3.5 Status of the PRISM system

The implementation of the key features has been finished. The supporting tools like the administrator information tool and the snort IDS-IDMEF aggregator have been implemented and report high amounts of incidents. FAUST [FP06] was deployed and is working perfectly. At the moment, the PRISM system is being deployed and moved to a new server. The next steps will be the training of the network administrators of our university and a test with a few simulated security incidents will be started.

4 Discussion and Case Studies

The generic topology of PRISM components consists of the following components: a ***PC*** is infected with malicious software and a ***sensor*** is triggered. The sensor reports the information via IDMEF to the ***PRISM*** system. The PRISM system scans the WWW pages of the ***AV Company*** and proposes a solution and helpful links to generate the incident WWW page. The incident administrator receives the incident, checks the WWW page, edits the content, and decides to isolate the infected PC. The isolation is done by a policy enforcement server of the ***access network***. The PC now can only access DNS servers, the PRISM system, the update servers, and the systems of the AV Company. A user of the PC wants to access the ***Internet*** and is rerouted via the ***network backbone*** to the PRISM system. The WWW page of the incident is presented and the user is confronted with the incident. In the ideal model the user now has all information to resolve the issue without further action of the CSIRT. This includes the download of patches from the ***update servers***, which still are accessible.

4.1 Case study: academic university networks

A possible usage scenario is a university network. Academic networks are very often realized as open networks to enable non-restrictive research. Usually, such procedures often lead to a security nightmare as a side-effect: many bad administrated PCs, no implemented security policies, and spare time PC administrators provoke many incidents. Frequent infections with malicious software overwhelm the CSIRTs. This is the perfect scenario to use the PRISM tool to support decentralized users and administrators in incident handling. The PRISM toolkit lightens the load of the incident management team because the tool offers support for identifying the user of the infected host. Additionally, it informs the user about the incident and offers help, e.g. removal instructions and tools. The system also tracks the incidents and can remind the CSIRT when there was no reaction in a certain time period.

4.2 Internet service provider

Another possible use case is at an internet service provider (ISP) to care about the security incidents of the customers. The problem here is a combination of mutual disinterest in each other. The customer is not interested in spending time and money for administrating and hardening the PCs. The internet service provider on the other side is not interested in spending time for consulting low budget projects. For the ISP, it is a complicated situation: on the one hand, there are many security incidents reaching the CSIRT and on the other hand there are no resources to handle them. A solution could be the usage of the PRISM system. The special situation here is the necessary coordination with different laws such as privacy protection and the freedom of communication that is protected by the constitution. A solution may be a distributed architecture of the PRISM system and the supporting tools. This demands for a new trusted party. In Germany, there is the BSI (Bundesamt für Sicherheit in der Informationstechnik) state agency, which is responsible for the computer security in public networks. The PRISM architecture may be deployed in the following form:

- The ISP deploys sensors in the network detecting security incidents. Additionally, it maintains transparent WWW proxies to reroute HTTP requests.
- The BSI operates the PRISM tools and a solution management database.

When a security incident is registered by the sensors, the relevant information is delivered to the BSI PRISM system. The system then tries to get information about the incident including information about the user. Additionally, the BSI PRISM system sends commands to the ISP WWW proxies to reroute HTTP requests of the user to the BSI PRISM portal. Now, the user is informed about the security incident and asked to remove its cause. The PRISM portal offers helpful links and manuals to the user. After a certain time, the WWW rerouting is removed. The WWW proxy enforcement will be enabled from time to time with increasing frequency and time span to ensure a proper incident handling.

There are of course some unsolved problems. The first problem is that users are nomadic. Furthermore, “Internet by call”-providers can not be integrated in a trivial way. A nomadic user will have a totally different IP address the next time he is using the Internet and often uses an anonymous login. Finally, the data protection issue is still relevant here. Not only is it questionable to read the content of the user communication but also to track a user over provider boundaries.

5 Conclusion

We introduced the problem of solving large numbers of security incidents in a data network. To distinguish the relevance of security incidents, these are divided into classes according to their threat. The low-priority ones are not necessarily to be solved by a security response team. Instead, these incidents need to be presented to the end user. The delegation of the responsibility towards the user saves much time for both the CSIRT and the users. On the other hand, the maintenance costs must be considered for operating a toolkit that handles all the incidents. This includes the time needed for configuring the incident cases and the pre-selection by CSIRT members. Another open point may be the security of the self-service terminal.

The PRISM tool is a study which is thoroughly programmed and tested in the environment of the backbone network of the University of Erlangen. If ever the PRISM software comes into a wider attention and practical use, more structures and best practices following CSIRT instructions should be considered. A good overview about this subject is a handbook [GK04] written for the Carnegie Mellon University.

The next steps will be the further examination of incident management workflows and escalation procedures. Also, it will be vital to specify the difference between annoying incidents which demand almost no attention by qualified personnel and relevant incidents which are critical. The improvement of the solution management will also be a topic for future work. Here it is of interest how the accuracy in finding a solution can be improved. Good candidates seem to be scoring mechanisms, fuzzy logic, or rule based systems. In this context, it is also important to talk about data and privacy protection issues. Appropriate mechanisms must be included into the solution management module.

References

(1) [FP06] Florian Prester, “Security within networks with ease and One-Command-Philosophy,” Terena Networking Conference, Catalania, Spain, 2006.

(2) [AV05] Alexander Vitzthum, „Implementierung eines Vorfallsmangementsystems für IT-Sicherheit,“ Studienarbeit (Pre Master’s thesis), Institut für Informatik, Lehrstuhl für Kommunikationssysteme, Erlangen, 2005.

(3) [JK04] Jochen Kaiser, “IT-Sicherheit im Nebel,” GUUG Frühjahrsfachgespräch, 2004.

(4) [BP1] Best Practical Request Tracker, http://www.bestpractical.com/rt

(5) [BP2] Best Practical Request Tracker for Incident Response, http://www.bestpractical.com/rtir/

(6) [OT1] Open Source Ticket Request System OTRS, http://otrs.org

(7) [SI1] SIRIOS, http://sirios.org

(8) [CVE] Common Vulnerabilities and Exposures, http://cve.mitre.org

(9) [GK04] Georgia Killcrece, Klaus-Peter Kossakowski, Robin Ruefe, Mark Zajicek, “Organizational Models for Computer Security Incident Response Teams (CSIRTs),” Carnegie Mellon University, 2004.