Adapting the Ticket Request System to the Needs of CSIRT Teams

PAVEL KÁCHA
CESNET-CERTS Computer Security Incident Response Team
CESNET, z. s. p. o.
Zikova 4, 160 00 Prague 6
THE CZECH REPUBLIC
ph@cesnet.cz

Abstract: CSIRTs (Computer Security Response Teams) are the natural response to the widespread internet threats. Many of them have grown of small, but focused groups of people, by streamlining and expanding of what they have been already doing as part of their IT administrative work. Formalisation of the procedures and workflows brings the need for specialised tools, helping with incident categorisation, authorisation of incident origin and general workflow. Also, special nature of incoming report emails introduces a new issues to otherwise well-known spam and backscatter fighting methods. As well as low level know-how, important part of security team practices are also higher level statistical analyses for pinpointing potential threats and trends. This paper proposes approaches to these problems and describes their implementation as modifications and supportive applications for Open Ticket Request System (OTRS), as well as experience from usage in the real world medium-sized security team.

Key-Words: OTRS, CSIRT, CERT, security incident, metadata, issue management, Bayesian analysis, antispam, backscatter, statistics

1 Introduction

In order to refine the basic need of any CSIRT [2] team, let us first analyse the life-cycle of a typical security incident report.

Once the report is received, its relevancy is assessed and, where necessary, additional information is requested. Next, reports are categorized according to the networks affected and forwarded to their respective administrators, after consulting internal databases or WHOIS information. The responsible administrator then communicates directly with the original complainant (if needed) and finds a solution. If everything goes fine, from this point onwards CSIRT acts only as a spectator and a recorder. According to the seriousness of the report, the relevant administrator responsible may be contacted and response requested in case CSIRT had not been informed about the resolution in time. Afterwards, the report is finalised and marked with the appropriate outcome.

Report may of course arise from CSIRT team itself, based on network monitoring, audit systems [15] or proactive tools (as IDS [16]).

A range of tools for issue management exists (see [8] for an overview of suitable ones), however, none of them directly supports the incident report handling work-flow.

1.1 Real setup

Through this paper we will sometimes refer to a particular real world scenario as implemented in our project, so we should initially provide brief overview.

During its lifetime and growth, security response team inevitably reaches point, where workflow starts to be ineffective. Main members get overwhelmed by routine work, with decreasing time for solving complex incidents. Therefore we have split incident handling and involved Network Monitoring Center personnel.

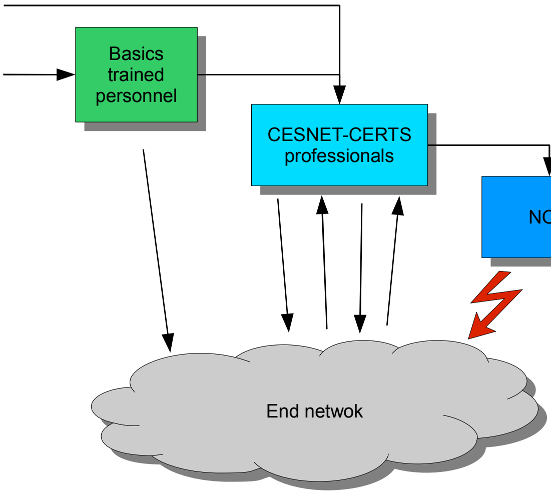


Fig. 1: Incident handling hierarchy

We provided them with necessary education and initial mentoring and they are now able to identify basic low priority incidents, as well as usual spam which misses filters (discussed later). They hand them to responsible end network administrator, and sieved out medium to high severity incidents go to original, well-trained CERTS staff.

CERTS personnel coordinate incident treatment (possibly) to succesfull solution, or (in case of unsatisfactory or no response) call on to Network Operation Center to restrict or completely block the incident origin.

Our organization also works as LIR (Local Internet Registry) and takes care to keep all network blocks, assigned to member and customer networks, registered in RIPE Regional Internet Registry database, along with correct abuse contact.

2 Basic problems

Let us list basic problems (apart from rudimentary issue management), that the majority of incident report handling teams is facing.

Searchable and reliable metadata. Each incident should be accompanied at least with the IP address of origin, and possibly also with the associated network name and responsible person's contacts. Human analysis and manual metadata extraction is repetitive and rather error prone. Possible automated method would set the basis for more advanced processing.

Incident categorisation. Classification as per the incident type (and consequently its seriousness) forms the basis for statistics and trend analysis.

Incoming traffic sanitization. Spam, virus and backscatter are well known and documented fields of expertise. However in a specific case of incident reports, usual statistical and heuristic methods face unexpected challenges. An incoming incident report itself may contain a sample of spam, virus or unsolicited bounce, and often gets classified as such as a whole. Additional measures are therefore necessary.

Lifetime and bulk checks. Incident reports often get stale, without any downstream response. On the other hand, individual responses may be swift, but the number of incidents of a particular origin may reach suspicious amounts. Simplistic human processing in this case is error prone.

3 OTRS introduction

OTRS (Open source Ticket Request System) is GPL licensed, Perl based trouble ticket (or issue management) system, used as the basis for our applications.

3.1 Tickets

The ticket is composed of a series of articles – textual updates to its state, usually e-mails. The ticket keeps a complete history of the changes made to it, either by human interference or through some automatic means. The ticket can be split into two, possible independent, cases, and more tickets relating to one case can be merged.

Each article is in fact an email message in the RFC 2822 format, in the same form in which it was received (or generated). That allows for a seamless integration of signatures and encryption – in that way, OTRS utilizes existing standards, both S/MIME and OpenPGP.

Saving messages in the original format is an ideal solution for archiving security team's communication. The message does not need to be reconstructed; the binary image of the message is not tampered, and can be used for security data mining, origin analysis, or used as evidence, especially when supplied with the electronic signature.

Aside from the usual data, the ticket can bear an arbitrary name/data pairs. This metadata can be unalterably named by the administrator, or left changeable for the storing of any information that seems to fit in the time of the creation of the article.

3.2 Queues and states

Tickets are organized into several queues that can be created by the administrator and connected to particular users with defined rights. The typical scenario in the security team could be two queues: incoming one which would be managed by the first line of basic-trained personnel who are able to solve or delegate via mail the basic types of incidents. The remaining ones would be moved into another queue, managed by specialists and highly-trained staff who can then then focus only on important or unusual incidents.

During its lifetime, each ticket goes through series of states. A state is property completely orthogonal to the queue which can represent important turning points in its history – for example external update, timeout or closing reason.

For example of workflow analysis, see [5].

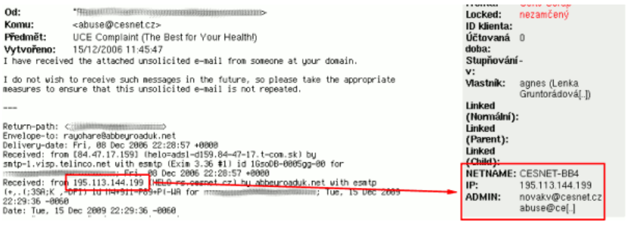
4 Automated metadata extraction

OTRS is able to store key/data pairs along with the data. These pairs can be arbitrary, but key names can be specified and defined as unchangeable. As we plan to attach at least an IP address, its network pertinence (according to RIPE database) and a responsible administrator's contact, deduced from network block information, we have defined these keys as the fixed first three metadata values, under the names NETNAME, IP, ADMIN.

These fields are editable, so any human operator can spot and correct possible errors. However, data should be pre-filled in some way, to ease the burden of hunting them down and filling them up by hand.

We considered various schemes of an automatic mail analysis. After some testing we finally came up with an automated approach.

An overwhelming majority of incidents contains only one IP address from a particular autonomous system. Our analyser breaks mail into its MIME sub-parts and searches in subject, main body and all attached data recursively for anything conforming to an IP address format. This can result in a large number of addresses, which have no connection with CSIRT constituency networks, thus we filter out only those belonging into governed network space and remove any duplicities. This usually yields only a single IP address. Where the result contains more addresses, we leave the decision on the human operator at a later stage. Only a human, being aware of the respective context from the mail message, can conclude whether the incident report concerns more IP addresses (and should be separated into two tickets) or whether the second address is a bogus.



Od: [redacted]
Komu: <abuse@cesnet.cz>
Předmět: UCE Complaint (The Best for Your Health!)
Vytvořeno: 15/12/2006 11:45:47

```
I have received the attached unsolicited e-mail from someone at your domain.

I do not wish to receive such messages in the future, so please take the appropriate
measures to ensure that this unsolicited e-mail is not repeated.

---

Return-path: <[redacted]>
Envelope-to: rayohare@abbeyroaduk.net
Delivery-date: Fri, 08 Dec 2006 22:28:57 +0000
Received: from [84.47.17.159] (helo=adsl-d159.84-47-17.t-com.sk) by
smtp-1.visp.telinco.net with esmtp (Exim 3.36 #1) id 1GsoDB-0005gg-00 for
[redacted]; Fri, 08 Dec 2006 22:28:57 +0000
Received: from 195.113.144.199 (HELO rs.cesnet.cz) by abbeyroaduk.net with esmtp
(+,.(;3SA;K ,-OPI) id H4+9II-P89+PI-WA for [redacted]; Tue, 15 Dec 2009
22:29:36 -0060
Date: Tue, 15 Dec 2009 22:29:36 -0060
```

Locked: nezamčený
ID klienta:
Účtovaná doba: 0
Stupňování: -
Vlastník: agnes (Lenka Gruntorádová[..])
Linked (Normální):
Linked (Parent):
Linked (Child):
NETNAME: CESNET-BB4
IP: 195.113.144.199
ADMIN: novakv@cesnet.cz abuse@ce[..]

Fig. 2: Metadata filled in from extracted IP

Obtained addresses are then screened through the RIPE database.

Custom developed module asks RIPE database for info related to extracted IP address. RIPE textual output gets analyzed and important information gets parsed out.This is an example of RIPE output:

```
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and
% Conditions.
% See http://www.ripe.net/db/support/db-
% terms-conditions.pdf

inetnum:         195.113.134.128-
                 195.113.134.255
netname:         CESNET-BB2
descr:           CESNET, z.s.p.o.
descr:           Prague 6
country:         CZ
admin-c:         WS9876-RIPE
tech-c:          WS9876-RIPE
status:          ASSIGNED PA
mnt-by:          TENCZ-MNT
mnt-lower:       TENCZ-MNT
remarks:         Please report network abuse
                 -> abuse@cesnet.cz
changed:         tgwr@cesnet.cz 20060413
source:          RIPE

route:           195.113.0.0/16
descr:           CESNET-TCZ
origin:          AS2852
mnt-by:          AS2852-MNT
remarks:         Please report abuse
                 -> abuse@cesnet.cz
changed:         tgwr@cesnet.cz 20060626
source:          RIPE

person:          Wenceslas Smith
address:         CESNET, z.s.p.o.
address:         Zikova 4
address:         Praha 6
address:         160 00
address:         The Czech Republic
e-mail:          xnovnov@cesnet.cz
abuse-mailbox:   abuse@cesnet.cz
nic-hdl:         WS9876-RIPE
notify:          notify@ces.net
changed:         tgwr@cesnet.cz 20070904
source:          RIPE
```

Some heuristics must be applied here, because many networks worldwide do not have working abuse contact defined, be it in recently added specific *abuse-mailbox* field or in (more commonly used) *remarks* field. In that case we analyze person data referred in *admin-c* and *tech-c* fields for *abuse-mailbox* or *remarks*, and in case none of them exists, real addresses from *e-mail* fields are used. Moreover, as *remarks* fields are meant for arbitrarily formed text, email addresses must be searched for and extracted carefully.

Of course, in cases where company in which the CSIRT team operates provides also LIR (Local Internet Registry), validity of its constituency data can be ensured by systematic monitoring and by defined workflow processes.

Resulting addresses, along with IP and network name information are inserted into mail headers in a form understandable by OTRS, which extracts the data and assigns it to the respective metadata fields.

This is an example of the generated headers:

```
X-Otrs-TicketKey1: NETNAME
```

```
X-Otrs-TicketValue1: CESNET-BB4
X-Otrs-TicketKey2: IP
X-Otrs-TicketValue2: 195.113.134.228
X-Otrs-TicketKey3: ADMIN
X-Otrs-TicketValue3: abuse@cesnet.cz
```

We are keeping an eye on the IODEF [4] and IDMEF [3] incident and intrusion description formats, as these provide a more precise target identification and standard form of further distribution. However, their proliferation is yet very low, and our approach would have to stay as a fallback even if these formats managed to gain wider audience.

We should note here, that according to our experiences in more widespread network (the whole Czech Republic IP space), vast number of major internet providers work also as Local Internet Registry for their constituency, but violate the RIPE LIR policy by providing IP address blocks to their customers and by not embedding corresponding network contact information back into the RIPE database. This alone makes incident report distribution and contacting responsible administrators very difficult, sometimes near to impossible, due to the internal policies of their providers, who are often more than unwilling to be of any help, even though incident originates in address space assigned to them.

5 Automated incident categorization

Each incident bears its characteristic features and can be categorized as a well known type. Categorization can be managed by human intervention, however if we could achieve a reliable machine classification beforehand, we would get a valuable clue on how to process a particular incident. Categorization is also necessary for further statistical and trend analyses.

Similar and a well studied problem is spam identification – free form mail text is analysed to decide whether message is allowed to reach the destination mailbox or whether it is malicious or unsolicited commercial message. Statistical methods, based on Naïve Bayesian probability analysis which are used for the purpose of spam identification, constitute a two-way decision process.

In general, these methods generate a weighted histogram of words (or of n-tuples of words) or larger meshes as in the case of the hidden Markov model, based on previous learning history. Histogram values undergo a statistical cleaning and the combined representative value (based on particular method, it can be some kind of average or median value) determines the spam rate of a message.

However, there is nothing inherently two-way in these methods – see [12] for principles. One of the first Bayes statistics based filters, Jason Rennie's *ifile* [11], supports n-way filtering. By means of several custom scripts we inserted *ifile*'s classification into the incoming queue. The analyser output is then added as an associated header, and later it is used directly as an incident category in OTRS.

The success of statistical methods stands and falls with quality of learning. Our current work-flow guarantees that at the most one day old incidents are already reviewed and processed by human operator. To eliminate human slips, we use all tickets older than two days as the basis for building up the *ifile*'s database.

5.1 Incident taxonomy

We use a simplistic (but coherent) approach to incident taxonomy. As exhaustive enumeration is not necessary, only incident types of nowadays highest proliferation have been used. As several incident types traces overlap (for example spam is a part of phishing), we declared a rule of the most fitting modus operandi – incident type which contains incident symptoms completely fits.

1. *Spam* – usual unsolicited commercial email.
2. *Bounce* – mail backscatter (usually caused by spam).
3. *Phishing* – spam is used as advertisement for a website which imitates some well known institution in order to gain its clients' personal information (bank account credentials, credit card information).
4. *Pharming* – similar to phishing. More sophisticated DNS attacks are used to cover the redirection of the client to a fraudulent site.
5. *Copyright* – copyright infringement, usually by means of peer-to-peer networks.
6. *Trojan* – malicious code on a server attempting to attack server clients and spread on (by defaced web page or active probing).
7. *Malware* – malicious code on a client workstation, for example keylogger, rootkit or malware as a part of botnet. Trojan and Malware classes partially overlap, in many cases they can be in fact the same code. However we are trying to distinguish the situation where primary function is to spread

and attack another machines (Trojan), while Malware mainly collects user data, sends spam, etc.

8. *Probe* – probing servers and networks. Portscan, portsweep, SSH (or other service) scan or unsuccessful attempts to crack service.
9. *DOS* – simple or distributed. Again it partially overlaps with a probe but DOS's primary aim is denying the service, not a compromise.
10. *Crack* – generally any other compromise.
11. *Other* – anything we are not able to classify into previous categories. Meant as a fallback category, which should get reviewed regularly, and the results of which should get incorporated back into this taxonomy.
12. *Unknown* – it is not possible to clearly state the incident type from report (usually some additional clarification from the complainant is needed).

6 Incoming traffic sanitization

The world of email nowadays is widely infected with unsolicited commercial emails, backscatter bounces and various kinds of worms and viruses. Some kind of filtering of incoming mails is therefore necessary to keep amounts of messages to be handled manageable.

However, an incident handling mailbox may face expectable problems – incident report messages themselves can contain samples of spam, bounce or viruses. Usual anti-spam and anti-viral methods fail and some kind of additional treatment is necessary.

6.1 Spam

This section does not offer a silver bullet – we have yet to find a reliable method to distinguish spam. This is even more true for spam in incident reports.

6.1.1 Safe methods

In the case of incident reports, whose ambiguous nature renders most of (data analysing) anti-spam methods unreliable, we have two options.

First option is resignation for automatic spam detection methods. OTRS supports more tiers of incident report management, so if we have cheap manpower at our disposal, we can train these personnel to sieve incoming unsolicited emails (and possibly some trivial incidents). However, in larger than the smallest setups this way quickly becomes economically unrealistic. Our spam ratio estimates corellate with Sophos [14] findings – about 96.5 % of incoming messages are spam. Human work to get rid of it causes increased human error ratio, which gradually overweights benefits.

Second (and inevitable) option is to deploy at least some compromise anti-spam methods. Readily applicable are methods, which avoid examining contents of mail messages. This involves mostly (adaptable) blacklist methods – DNSBL, Greylisting and Nolisting. In case of DNSBL we must make sure that we use only header checking lists, otherwise we fall in the same trap as before. Greylisting and Nolisting capitalizes on usual spammer behavior at the very border of mail system. It is unviable for spammer to wait and check errors of SMTP communication, so temporary rejecting of unknown sources (and expecting them to try again according to well defined and widely accepted rules) keeps number of unsolicited mail away. Spammer also tries only one mail exchanger – usually first or last – in its attempt to deliver mail. When we set first and last MX records for domain to machine, which rejects SMTP traffic, legitimate mail transfer agents will correctly try next MX, according to priority. The spammer who does not check result of transfer attempt, inevitably fails.

Also, heuristics like SpamAssassin with body introspection manually turned off can be used without problems.

6.1.2 Pessimistic method

After deployment and tuning of previous methods, we found out that ration of spam still stays unpleasantly high to process by human. Thereby we have decided to switch to pessimistic approach.

We have enabled full body heuristics (by means of SpamAssassin), and during initial "soft" phase, consisting of only tagging, not separating of vast number of existing incident reports we have created manually selected subject-keyword whitelist. Messages which contain any of these words or phrases in *Subject* line bypass spam analysis and are allowed to enter the system directly, notwithstanding that they were marked as spam by preceding filter.

The list is maintained in the form of a regular expression for SpamAssassin:

```
/abuse mail|abuse-mail|abuse of|
abuse report|abuse spam|e-mail spam|
multiple spam|received spam|report
abuse|reported spam|reporting spam|
returned spam|spam:|spam abuse|spam
complaint|spamcop|spam from|spam
mail|spammails|spam mails|spammer|
spamming|spam-rbl|stop the spam|
ube:|ube-uce|ube\/uce|uce:|uce-ube|
uce\/ube|ube from|uce from|\[uce\]|\
    [spam\]|spam received|uce complaint|
    ube complaint|phish|fraud/
```

Effectivity of whitelist is regularly monitored to identify possible false positives (incident reports marked incorrectly as spam), however new patterns have not been added in a long time.We experienced two false positives last year (from over 1136 incident reports), which means we stay under 0,2 %.

6.2 Unsolicited bounces

As bounces (or backscatter) we characterize mail delivery report messages, whose origin is not message sent by us. This is usually caused by spam with forged envelope headers – destination servers have low to no possibility to check the authenticity of the sender. Spam generating trojans usualy use random contact data from addressbook for both sender and recipient and bounce messages generated by destination servers, rejecting unknown addresses, go to forged source.

In the case of mail bounces we have achieved a significant advantage. We know we should only get bounces to messages originated by us. Therefore we are able to keep track of ticket identification numbers (which are injected into subject lines of each message sent). No bounce message (identifiable by an empty *Return-Path* header line) which does not contain existing ticket identifier younger than two months (to keep machine work low) anywhere in the *Subject* line or body is allowed to enter the system.

We face a problem here – the format of mail delivery messages [10] is specified very vaguely. There are strict requirements to some of message headers, but *Subject* and body of the message are completely free form. Some mail delivery agents (mainly certain *qmail* versions) do not attach enough of the original message to keep the ticket identifier. However according to our analysis conducted on nearly seven thousand of bounce messages shows only 0.5 % of such messages which is very acceptable loss ratio. Anyway, the situation with such stubborn agents has generally been improving.

6.3 Viruses

All mail is handled and sanitized for viewing by OTRS. OTRS is a web based application, so security precautions before rendering arbitrary email content into a browser are necessary. The content is completely stripped of scripts and HTML tags, thus mere viewing is secure. The only risk remaining is for the operator to open mail attachments directly, however this can be addressed by a policy or necessary tools (antivirus, anti-malware) can be installed on operator workstations, should the used platform need it.

7 Lifetime and bulk tests

7.1 Lifetime checks

Incident reports handed downstream to responsible security teams or administrators are usually handled on a timely basis, however not all teams have the same expectations, human resources and priorities for particular incident responses. Also, possible human error should be considered. A higher level team must therefore take care of reports during their whole lifetime, ask for updates, take actions when there is no response, and inform the claimant properly.

Human or technical errors are likely to occur even within the CSIRT team itself.

We have developed a set of modules for monitoring open tickets timeline. A ticket, which does not get proper treatment within expected timeframe (2 days in case of downstream team, 30 minutes in case of first-tier local operators) is raised and other members of the team can be informed.

OTRS supports regular check of tickets for some conditions and changing them accordingly but the time can be checked only in relation to the ticket creation, not its update. However the time of the last update is internally stored by OTRS. We have thus created an auxiliary script (running as one of OTRS's cron scripts), which goes through open tickets, checks the time of their last update, and tickets exceeding some timeframe change the state. Timed-out tickets are thus not rotting in the queue until somebody accidentally spots them.

While developing the script, we had to step aside from the usual OTRS ways and combine a direct access to the database with the internal object model. We execute usual SQL statements over the relational repository, which gives us a list of affected ticket identifiers. We then use this list to instantiate real OTRS ticket objects, and use their methods for a full featured manipulation. This ensures that all auxiliary structures are updated accordingly along with history messages.

7.2 Excessive number of reports

We consider some incident reports solely as informational. However, a higher number of common incidents reports on one particular IP address from various sources may foreshadow a more serious problem going on, so seriousness of such incidents should be re-evaluated by human operator.

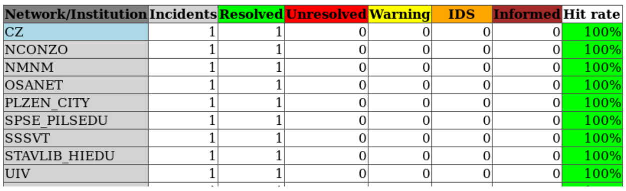
Again, based on previous work and principles, we created a module for checking unusual amount of incidents from one IP address and sending email notifications if a certain threshold is exceeded.

Results usually correlate with data from the CESNET Intrusion Detection System [16].

8 Statistics

Reliable incident source authority identification and automatic incident classification gave us interesting data source for further statistical analysis to be able to compare the incident solving hit rate of our members and constituency, and to review incident type proportion rate trends.

OTRS has some basic statistical module, however its functionality is limited to basic time/state/queue based counts. As the basic data model of OTRS is nicely transparent, fetching more complex data is just a case of straightforward use of conveniently crafted SQL queries. Again, we used our own module with subsequent processing of results and formatting them into a visually and factually convenient output. We were also able to add some data from other sources (annotate institutions with their whole names instead of RIPE shortcuts) or apply some more visually convenient elements.



Network/Institution	Incidents	Resolved	Unresolved	Warning	IDS	Informed	Hit rate
CZ	1	1	0	0	0	0	100%
NCONZO	1	1	0	0	0	0	100%
NMNM	1	1	0	0	0	0	100%
OSANET	1	1	0	0	0	0	100%
PLZEN_CITY	1	1	0	0	0	0	100%
SPSE_PILSEDU	1	1	0	0	0	0	100%
SSSVT	1	1	0	0	0	0	100%
STAVLIB_HIEDU	1	1	0	0	0	0	100%
UIV	1	1	0	0	0	0	100%

Fig. 3: Example of statistics by organization

(This example data are artificial, we are not allowed to disclose real values.)

Tabular data are not usually easily comprehensible for bystanders or management without further description, so we usually use more perceivable graphical representation.

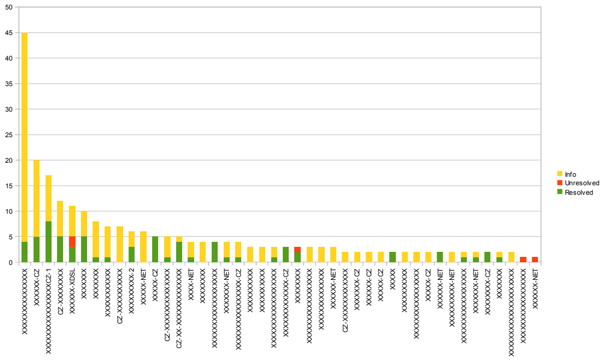


Fig. 4: Example of visualized statistics by organization

(Graph labels have been anonymised.)

These IP subblock statistical reports are of immense value as a tool for showing constituency network representatives their weak spots, pointing out number of incidents in their network and their effectivity in solving them in comparison with surrounding of similar networks.

Based on data gathered by automated incident categorization we can extract interesting data about particular incident type proliferation and their ratio.This is important indicator of where efforts for security, education and prevention should be directed.

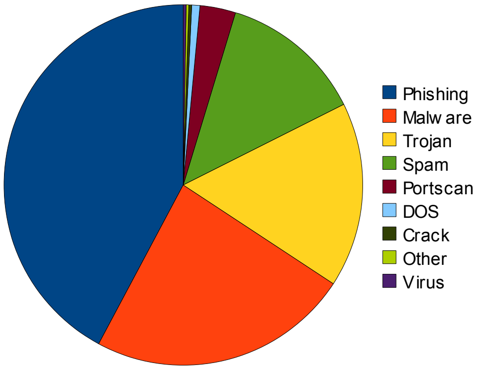


Fig. 5: Example of incident type distribution graph

Important data can be also change-over visualizations, which indicate results of previous efforts or changes in trends in incident type distribution.

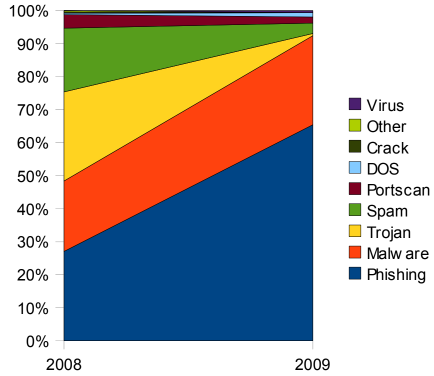


Fig. 6: Example of year to year trends

9 Architecture

Our mail setup accepts mails for *certs@*, *abuse@* and *postmasters@* addresses from main CESNET domains.

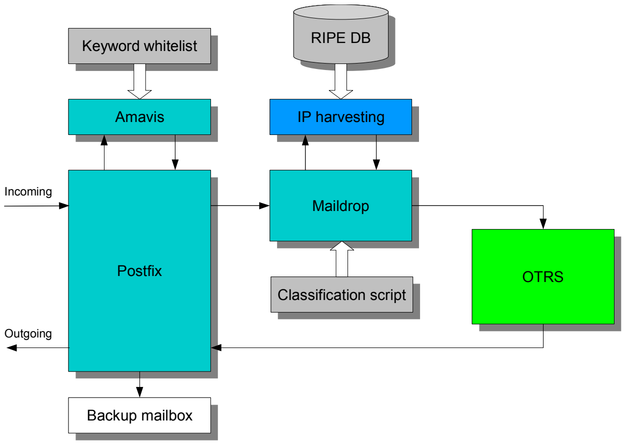


Fig. 7: System architecture

OTRS is able to accept mails by piping it to its auxiliary *bin/PostMaster.pl* script or by POP3 polling. We have used the former method, mainly because of its flexibility. During the initial deployment, the mail was dispatched by Postfix directly into this script through alias file. Currently we are using the Maildrop [6] mail delivery agent as a wrapper and caller for metadata extraction, incident categorization and anti-spam/anti-virus/anti-backscatter modules.

Incoming mail is accepted and processed by the usual Postfix setup.

Also a backup mailbox where all incoming and outgoing mail is copied in real time has been set up. We used the usual alias record method for incoming mail and OTRS capacity to duplicate all outgoing mail for outgoing mail:

```
$Self->{'SendmailBcc'} = 'backup@example.cz';
```

Usefulness of Maildrop shows up in connection with OTRS special headers handling. OTRS understands a definite set of mail headers the content of which can modify its behaviour - choose a particular queue or add some metadata. OTRS itself has a way to classify and define specific actions on mails, but this support is limited, which makes using of the real delivery agent a natural choice.

10 Applicability to alternatives

OTRS is not the only ticket request system, which can be used for CSIRT incident reports management.

The recent growth in open source development community needs has initiated a number of bug tracking projects with sound and dynamic groups of developers created around them. The fact that they are strongly development-orientated, with centralized architecture and weak support for external communication may be seen as their drawbacks.

10.1 RTIR

RTIR [1] is a tool, created especially for incident response teams, and adds functions suitable in particular for large enterprise teams (for example vulnerability management). It is tailored on top of its creators, Best Practical Solutions, flagship product – Request Tracker.

Most of the principles, described in this paper, are applicable, however considerable part of code would need to be rewriten and adapted to RTIR internal structures and architecture.

Metadata extraction script is mostly usable – all basic logic can stay in place, but part, which generates OTRS headers would be rewritten to provide metadata by RT mechanisms. Similar situation arises in case of automatic incident categorization – adaptation to fetching ticket bodies and setting metadata in RT data model and particular database would be necessary.

Spam and virus detection is completely independent on target system, so principles and whitelist described apply without problems.

Another situation arises in unsolicited bounce detection – analyzing of email bodies would need to be updated again for fetching ticket identifiers from RT database.

Significant problems will arise with lifetime and excessive reports checks. Created scripts are tightly coupled with OTRS internal structure and adaptation (even though RT is written in Perl, the same programming language as OTRS) would in fact need substantial rewrite. Statistical modules are also based on intimate knowledge of OTRS internal data model and adaptation would not be straightforward.

10.2 RoundUp

Another active community has grown around the relatively new Python programming language. Several ticket systems have been developed based on Python. If we put aside those based on complex frameworks (Zope) which carry the burden of nontrivial management with them, RoundUp [7] issue tracking system is worth keeping an eye on, and if it successfully passes its infancy and design shake up period, it may become a viable contender.

We face similar difficulties in adapting original code here. Metadata extraction and incident categorization, along with unsolicited bounce detection seem relatively easy – only adaptation of the specific routines to the RoundUp database model or API is necessary.

Spam and virus detection is applicable without changes, considering that administrator should use highly similar system architecture.

Lifetime and excessive report checks represent real problem here – considering another language, complete rewrite (based on described principles) would be necessary.

10.3 Traditional e-mail workflow

Small and/or young security teams in their early stages, especially those grown from group RFC addresses management administrators, usually start with incident report management through standard mail, or (to involve more team members) shared IMAP mailbox.

This approach is easy to set up – Several report managers may view the same set of IMAP folders - changes made to the folder by someone else are instantly visible in all modern IMAP clients. The messages can be archived in a hierarchy of folders according to their state and affected networks.

The strength of this approach is the ability to handle signed and encrypted messages easily, be it PGP or S/MIME. This functionality is usually an inherent feature of latest email clients. Some clients even support email templates adequately.

The weaknesses include complicated linking of a particular message with its author and threading and merging of messages belonging to one case. Standard email capabilities of Message-Id and references are often broken, be it by obsolete email clients and remailers or by users during chain of forwards of organizational hierarchy cruising. This includes also the inability to track split and merged reports. Some email clients support manipulation of mail threads explicitly, for example Mutt.

Automated metadata extraction and incident categorization can be helpful here in case of MUA, which supports prominent display of particular headers and/or work with them. Our team workflows have originated this way, we have been using Mozilla Thunderbird with Mnenhy extension for header display and manipulation.

Spam and virus checking is again decoupled from the incident report management, so application is straightforward. Unsolicited bounces detection seems harder – the administrator would have to implement gathering of message identifiers of outgoing mail, either by hooks in message delivery process, or by monitoring mail transfer agent logs.

Lifetime and excessive amounts, along with statistics, are however not applicable, because of unordered nature of email communication – automat is not easily connect related messages to filter out duplicities.

11 Code

All work is released under the GPL license on the CESNET FTP server:

ftp://ftp.cesnet.cz/local/otrs/otrs-2.1.2/

Available files are published mostly in the form of patches, except for statistical, metadata and categorization modules, which are prepared as archives with all needed scripts inside.

12 Conclusions

Finding a tool which would be an added value to the incident response team and would not have any significant drawbacks is by no means an easy task. As it turns out, no ticket management tool is readily usable for small or mid-sized teams. Even the most advanced projects include nontrivial management or programming requirements.

Our OTRS ticketing system installation currently holds around 3800 tickets, not counting spam and unsolicited bounces. The OTRS interface is used by five core team members as well as six Monitoring centre operators to manage incident reports for several hundreds of assigned network ranges.

Automated metadata extraction and IP address identification through network range sieving works well on the CESNET networks. Later we also started to operate analogous service across the whole Czech Republic address range. The system works better than expected; current experience shows the need for a manual review of the data for less than 3 % of incident reports only.

The error rate of the statistical incident type deduction also remains similarly low, under 1 %. Our suspicion that accuracy of identification will slowly degrade over time due to human errors or omissions in correction (which would lower the quality of statistical database for Bayesian analysis) does not justify. As several other more advanced methods for text classification are being under research [13, 9], we may incorporate some of them in future.

Our handmade whitelist worsens the efficiency of the anti-spam filter; however it is the price to pay for lowering the false positives rate to nearly zero. It is nevertheless the least satisfying part of this project, we are keeping an eye on the progress in anti-spam technology for ideas on how to raise exactness and lower the need of human intervention.

Detection of unsolicited bounces also works flawlessly, despite not really helpful state of mail delivery error message format and mechanism idiosyncrasies. We are not aware of any loss of valid delivery message on our side. However, the note must be taken that proposed algorithm does work only on setups, where administrator keeps control on all outgoing mail for particular email address or domain, otherwise the identificator database would be incomplete, causing loss of legitimate delivery messages.

Timeout robots and excessive incident number detectors help us mitigate human errors and pinpoint possible anomalies in time.

Statistical tools have shown as an immense source of information and as a way to visualize efficiency of particular downstream organizations in combating the electronic crime. Also, incident report type distribution and trend visualizations help to identify growing threats for preparation of the right resources and strategies.

In spite of fact that our work is heavily based on OTRS system, there are parts, which are not that tightly coupled. Moreover, the mechanisms proposed are general, not platform dependent, and after appropriate adaptation their reimplementation should work at any analogous environment.

According to the configuration and development experience as well as users' observations, the work invested into the customizations and the code is paying off, and the course set has worked well so far.

13 Acknowledgment

I would like to thank CESNET, z.s.p.o. for support and contributions to this work, especially colleagues Andrea Kropáčová and Pavel Vachek. The work has been supported by the research grant Optical Network of National Research and Its New Application (MSM 6383917201) of the Ministry of Education of the Czech Republic.

References:

[1] Best Practical Solutions LLC., *RTIR: RT for Incident Response*, URL: http://bestpractical.com/rtir/

[2] Brownlee, N., Guttman, E., *RFC-2350: Expectations for Computer Security Incident Response*, (c) The Internet Society, June 1998

[3] Debar, H., Curry, D., Feinstein, B., The *Intrusion Detection Message Exchange Format (IDMEF)*, (c) The IETF Trust, 2007. URL: http://www.rfc-editor.org/rfc/rfc4765.txt

[4] Demchenko, Y., Danyliw, R., Meijer, Jan., *Incident Object Description and Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition*, TERENA IODEF WG, Feb 2002. URL: http://www.terena.org/activities/tf-csirt/iodef/docs/draft-terena-iodef-xml-005-final.txt

[5] Donko, D., Traljic, I., IT Service Management and Normatively Regulated Activities, *Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006*, ISBN: 960-8457-45-9

[6] Double Precision Inc., *maildrop - mail delivery agent with filtering abilities*, URL: http://www.courier-mta.org/maildrop/

[7] Jones, R., *Roundup Issue Tracker*, URL: http://roundup.sourceforge.net/

[8] Kácha, P., *OTRS: Issue Management System Meets Workflow of Security Team*, Technical report 7/2006, Prague: CESNET, 2006. URL: http://www.cesnet.cz/doc/techzpravy/2006/tickets-review/

[9] Lin, N. P., Hao-En, Ch., A Multi-Categorization Method of Text Documents using Fuzzy Correlation Analysis, *Proceedings of the 10th WSEAS International Confenrence on APPLIED MATHEMATICS, Dallas, Texas, USA, November 1-3, 2006*, ISBN: 960-8457-55-6

[10] Moore, K., *RFC-3464: An Extensible Message Format for Delivery Status Notifications*, Network Working Group, 2003

[11] Rennie, J. D. M., *ifile*,

URL: http://people.csail.mit.edu/jrennie/ifile/

[12] Rennie, J. D. M., *Improving Multi-class Text Classification with Naive Bayes*, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 2001

[13] Solares, C., Sanz, A. M., Bayesian Network Classifiers. An Application to Remote Sensing Image Classification , *Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS, Lisbon, Portugal, June 16-18, 2005*, ISBN: 960-8457-24-6

[14] Sophos Plc., *Only one in 28 emails legitimate, Sophos report reveals rising tide of spam in April - June 2008*, July 2008, URL: http://www.sophos.com/pressoffice/news/articles/2008/07/dirtydozjul08.html

[15] Vachek, P., CESNET Audit System, *Proceedings of the 13th WSEAS International Conference on COMPUTERS*, Rodos Island, July 23-25, 2009, ISBN: 978-960-474-099-4

[16] Vachek, P., *CESNET Intrusion Detection System*, Technical Report 10/2007, Prague: CESNET, 2007. URL: http://www.cesnet.cz/doc/techzpravy/2007/cesnet-ids/