**Title:** Addressing the Complex Security Aspects of Network Services in GÉANT Multi-Domain Environments

Gerard Frankowski  
PSNC – Poznań Supercomputing and Networking Center, Noskowskiego 12-14, 61-704 Poznań, Poland  
E-mail: gerard.frankowski@man.poznan.pl

Tomasz Nowocien  
PSNC – Poznań Supercomputing and Networking Center, Noskowskiego 12-14, 61-704 Poznań, Poland  
E-mail: tomasz.nowocien@man.poznan.pl

Wayne Routly  
DANTE – Delivery of Advanced Network Technology to Europe, City House, 126-130 Hills Rd, Cambridge  
CB2 1PQ, England  
E-mail: Wayne.Routly@dante.net

**Paper type**  
Case study

**Abstract**

<table>
<thead>
<tr>
<th>Purpose of this paper</th>
<th>This paper has been written in order to present a multipart approach to securing multi-domain services within the GÉANT network.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design/methodology/approach</td>
<td>Due to the complexity of the environment we have chosen the security-in-depth approach for securing GÉANT multi-domain services. As part of the layered approach, we secure the application layer (by providing knowledge on secure programming with training, cookbooks and an on-demand expert consultancy team) as well as its environment (by performing risk assessments and penetration testing). Finally, procedures for developing processes for handling multi-domain incidents, together with testing them during fire-drills have been created. These multi-layered solution approaches have been combined into an optimal solution, drawn from the knowledge and experience of security experts from European NREN CERT teams.</td>
</tr>
<tr>
<td>Findings</td>
<td>The proposed approach to securing services is encompassing of all security requirements for a multi-domain environment thereby, minimizing the probability of introducing security bugs into the release phase of all multi-domain systems. An area of improvement for the task to implement is that of security awareness among non-security people. Whilst their level of awareness is growing, it should be...</td>
</tr>
</tbody>
</table>
Research limitations/implications (if applicable) | Numerous security exploits on different levels have been found and resolved. At present the current case study is limited due to some of the components described in this paper having been in production for a short period of time, therefore the results are still been processed. Of those results already gathered, several beneficial outcomes have been derived.

What is original/value of paper | Experiences and the approach taken to protect multi-domain services and systems in the GÉANT project will be beneficial for the participants of other Research & Development projects. Also that the steps taken to improve and secure code will be used as a basis for implementing security where software is being developed – and, more generally – for those who are going to protect large software-oriented projects.

Keywords
IT security, defence-in-depth, GÉANT, multi-domain network

1 Introduction

1.1 GÉANT Network, GN3 Project and MDS Tools
GÉANT is a pan-European data network dedicated to the research and education community. Together with Europe's national research networks, GÉANT connects 40 million users in over 8,000 institutions across 40 countries. The project groups 32 European NRENs (National Research and Education Networks), DANTE (Delivery of Advanced Network Technology to Europe) and TERENA plus four associate NRENs [1].

The GN3 project is focused on developing and rolling out multi-domain services to NRENs and their users (institutions, projects or researchers) in order to enable them to get the best performance from the GÉANT network. Dedicated to this purpose are a suite of tools developed under the auspices of SA2 (Service Activity 2) – Multi-Domain Network Services. Within this activity is a task dedicated to security of the services within the activity. SA2T4 (Service Activity 2 Task 4) – Multi-Domain Network Services – Security, is this task, which focuses on providing appropriate protection not only for the implemented tools that facilitate using GÉANT Multi Domain Services (MDSs) but also for the service life cycle. For marketing purposes the task has been rebranded as the Multi-Domain Security Team (MDST) and further information can be found at http://mdst.geant.net

1.2 Today's IT Security Climate
The ever increasing threats that one are faced with on the Internet have become an inescapable problem and therefore a constant reminder of the security problems that must be resolved when designing and deploying new services. There are numerous reports that link the increasing number of known vulnerabilities in written code and of financial losses to the activity of crackers and cyber-criminals (e.g. [3]). GÉANT Multi Domain Services have are being developed and deployed in numerous NRENs. As such there is an increased risk that vulnerabilities could be introduced as a result of the widespread use of systems that do not employ consistency in secure programming awareness for the developers, or make aware to system and service administrators across the project of the importance of implementing anti-hacker best practices. The geographical and institutional heterogeneity in the GÉANT project adds an additional complexity to the mix. This complexity coupled with the services being of a multi-domain nature; make this an even more significant factor when having to secure the services and tools. To exacerbate this situation is the established fact that software vulnerabilities are actually unavoidable in produced software [4], especially for those non-commercial applications that are not built by large vendors. It is also recognized that the relative number of security bugs in applications has exceeded the
number of vulnerabilities in operating systems [2], which makes the goal of securing software produced within the project an even more significant goal.

2 The Security Economy and Defence-In-Depth Strategy

2.1 Dynamic and Multifaceted IT Security

Security should be viewed as an on-going process, not a single state [5]. Even if it was possible to achieve a state of “perfect security” could be achieved in an IT environment (which, as we will show later, cannot be considered as a real opportunity), the internal and external conditions of that environment are subject to change - and any security measures implemented for the environment may be no longer be applicable after the change. An example of internal change may be the installation of a new operating system or merely adding some new user accounts. A new operating system may contain insecure functionality and as a result one of the new users may define a weak password for their account. Externally changing conditions focus primarily on extending the knowledge of security bugs and discovering new attack types or analysing tools.

To make matters worse, ‘perfect’ security, although possible in theory, cannot be achieved in software environments. This is especially valid for software projects and it is worth a note that software is the foundation upon which IT environments are built: an example of these types of software ecosystems are the web application, web server, database server, the underlying operating system and the local firewall are all software systems and in some instances their source code size is measured in millions lines.

There are different estimates of the average number of software bugs in the source code. Research results from the Carnegie Mellon University's CyLab Sustainable Computing Consortium suggest that 20-30 bugs in 1000 lines of are an average in commercially produced source code [6]. The results of the Vulnerability Discovery and Remediation Open Source Hardening Project, initiated by US Department of Homeland Security in 2006, pointed to approximately. 0.3 bugs per 1000 lines of LAMP (Linux, Apache, MySQL, and Perl/PHP/Python) source code [7]. The actual number of bugs may depend on many factors, for example the assumed definitions and methodology, whether the source code is open, how critical the code is and whether the code is built by a large corporate vendor. The main outcome from the above gives rise to the conclusion that the actual software is never absolutely perfect and will always contain bugs within the code.

2.2 IT Security Economy

The advantage always rests with the attacker when exploiting vulnerabilities. If any software being attacked contains N critical bugs (for medium-sized and large software packages usually it may be assumed N>1), to make a successful attack only one of those N bugs need be found to prove successful. On the other hand, the vendor in order to provide a perfect defence would have to identify all N bugs. Finding an unquantifiable number of bugs is more time consuming as it is an unknown factor as external factors often limit the time available (deliverable deadlines in the case of an R&D project) at the vendor side for bug resolution. However, as most attacks are financially motivated it can be expected that the system will not be attacked if the expected return (which does not have to be strictly monetary) will be less than the attack. An attacker’s return depends on the gains per use then multiplied by opportunities to exploit (the number of bugs) but is also decreased by the cost of acquiring a vulnerability and the creation of an appropriate exploit mechanism [8].

As a result of the financial aspect; the best way to defend the system would be to make finding an exploitable bug as difficult and as unrewarding as possible. That goal may be obtained in different ways, for example by producing more reliable software and using dedicated security technologies like strong cryptography or even investing in more sophisticated security and monitoring systems. It must be also understood that there is always a cost limit when implementing security measures. For instance, it is theoretically possible to mathematically prove the software 'correctness', but this cannot be transferred to real applications. J. Naresh et al. [9] compare the bug density and the cost of producing 1000 lines of source code (aka KLOC) for industry software against the NASA space shuttle control software as a comparative benchmark. While in the former case they state approximately 5 bugs in the KLOC code and $5 000 USD for producing this code, in the latter the values are 0.004 bugs per KLOC code and $850 000 USD, respectively. Therefore the crucial factor is to be able to apply optimal security countermeasures that can be obtained within the realm of available resources.

2.3 The Defence-In-Depth Strategy

The Defence-In-Depth strategy is another way to bring the attack cost/gain balance into a desired state. This is actually a direct translation of the attack economy rule set onto the multi-layered nature of current IT environments. The idea is understood as the ability to manage risks with diverse and defensive strategies, so that if one layer of defence turns out to be inadequate another layer of defence will expectantly prevent a full breach [10]. By multiplying the number of obstacles that an attacker would need to overcome, increases both the attack cost and probability that unauthorized activities will be detected in their initial stage. The concept behind this is
that different classes of security solutions are able to identify different threats, and even different solutions of the same class may detect slightly different attack vectors, thus, when combined, providing a better attack vector coverage.

There are many layers where security measures may be applied: the application itself, its background servers, the operating system, local and corporate firewalls, log monitoring systems, IDS/IPS systems and so on. Whilst these are technical solutions, another aspect such as the organizational (procedural) facet must not be overlooked. This is especially important in the GÉANT multi-domain environment, where the services are maintained by the software development and administration teams from different National Research and Education Networks. In these NRENs we will have different security policies and incident handling procedures, creating a very heterogeneous security approach. Having established procedures not only helps to retain knowledge on how to secure systems for future infrastructures, but also resolves issues related to practical competency and facilitates a prompt response in the event of an attack. It must be borne in mind that if organizational measures do not support technical solutions, raising the security level will offer temporary relief.

2.4 Secure Project – Implementing Security in All Project Stages

Any project such as the GÉANT multi-domain service – is usually developed according to some accepted development lifecycle. By following a standard set of best practices it creates another dimension for embedding security within the project. The development lifecycle usually consists of several phases (stages), for instance defining the requirements, the design stage, the implementation stage, testing and maintaining.

Each project stage may contribute towards adding security vulnerabilities found in the final product. For instance, the inaccurate definition of requirements may result in a required security feature not been designed and implemented. For this reason, from security aspect, it is beneficial that security is implemented at each stage when the development lifecycle is iterative. Therefore it is possible to make corrections to the requirements list and design when planning for and producing the next release. During this process it is advisable that a security expert should be involved in the whole development life cycle, from the beginning of the project.

Different project stages will have diverse economic impacts to the project as a whole. M. Morana shows [11] that security bugs are more expensive when required to be rectified in the latter stage of the project life cycle and even more when the produced software has already been released. This can be seen as the primary reason why security must not be an extra feature that is added to the project in its final stage, but should rather be considered throughout the entire project life at each stage. This approach has been followed by GÉANT SA2T4 by looking at each stage of the lifecycle and implementing a process or mechanism to ensure security is considered for new and existing systems developed within the GÉANT project.

3 The MDST Approach to Applying Multifaceted Security

3.1 MDST Structure – Addressing the Different Layers

Securing applications and systems requires a multi-faceted approach. By implementing multiple diverse layers of security one is able to cover more aspects and therefore areas of potential threats.

Firstly, two subtasks are distinguishable within the SA2T4 task: Security Expertise Delivery (SED) and Multi-Domain Security (MDS). The SED concentrates on using and transferring security expertise between the security experts and development teams. The MDS directly involves Multi-Domain Services tools and the incidents that may take place in their environments.

Before progressing with implementing secure solutions and mechanisms, a greater understanding of what we were protecting was required. To address this potentially extensive problem, the team posed a series of questions that clarify the objective and goals the team wants to achieve within the activity.

The questions posed by the MDST to provide solutions for who, what and why when protecting Multi-Domain Systems and Services are:

1. What are we doing to handle incidents?
2. What are the tools we can use to improve the security of multi-domain services?
3. What are we doing to protect deployments?
4. What are we doing to improve the security of the multi-domain service tools?

These questions are answered by mapping tools / mechanisms / services to each of these questions thereby providing an answer and a means to resolve these issues.
In the following sections each of the various ‘components’ will be touched upon to show their links to other sections and how from a single component to the entire activity each one is linked in order to provide a multi-layered security approach to protecting multi-domain systems and services.

### 3.1.1 What are we doing to handle incidents?

In order to handle multi-domain security incidents, a virtual security team (VST) was created. This team assists in disseminating incident information to other deployments. This provides a specialised conduit to every deployment that could potentially be affected by this incident. The team is comprised of CERT personal from nine CERT Teams that contribute to SA2T4 as well as a representative from the NRENs that have deployed the MDS tools. This specialised team allows them to react quickly once there is a notification and to involve only relevant parties in the notification.

Next, in order to ensure the virtual security team had a process to follow to ensure that all the RELEVANT and CORRECT people are informed when dealing with an incident, a workflow was created. The MDS Incident Workflow provides a roadmap as to who to inform when one has specific options available such as if the incident is software bug or malware related. The workflow provides a standardised approach to handling incidents in a multi-domain services environment.
To support the VST in processing incidents Request Tracker (RT) was used as the ticketing system. In order to store the information from incidents a knowledgebase was created using RTFM (Request Tracker FAQ Manager) to provide the VST with a mechanism to store, track and monitor incidents related to multi-domain systems. The knowledgebase has a long term benefit in that it also provides a repository of possible solutions when similar types of incidents are reported.

3.1.2 What are the tools we can use to improve the security of multi-domain services?

To improve the security of multi-domain services as a whole one should understand the threats and risks posed to these systems. As such SA2T4 perform risk assessments on all new systems enabling security specialists to review the security of a deployed solution against predefined areas thereby enabling risk mitigation mechanisms to be implemented.

An additional tool that can be used to protect multi-domain systems is by performing pen testing. Pen testing Enables the multi-domain security team to look at deployments from an attacker’s perspective. For instance, performing a scan of the machine to see if there are any undocumented ports open? Also what happens if we change the parameters in DB calls (SQL injection) or even privilege escalation in systems? These types of findings are critical as they show that these systems are vulnerable in open environments. Such steps form a proactive approach to securing these systems.

To deal with new and existing incidents, you must ensure that team and processes are capable and fit to handle security breaches. The fire drill component is therefore critical as it ensures that if there are incidents they are dealt with correctly and efficiently. The fire drill not only covers the virtual security team aspect but also ensures that the interaction from and to the NREN CERTs are handled correctly and processed timeously. The fire drill is a barometer to ensure that all the components we have put in place function correctly when put together.

In all security environments there exists a requirement to provide security expertise for all users. The security expertise consultancy is this pool of expertise that handles all security related queries, be they for deployments or deployers. It is a resource that the community can refer to, to get advice and recommendations. The security expertise consultancy is also the tool that all the information that we glean from pen testing, risk assessments, fire drills, secure code training and the knowledge base gets fed into and the outcome results in consultancy requests where the multi-domain security team can proactively assist and advise on.

3.1.3 What are we doing to protect deployments?

A key area in multi-domain systems is protecting the actual deployed system. This is where the tool and/or service are at its most vulnerable. Therefore we use the risk assessments to gauge the threats and risks that this deployment is exposed to, for instance if the recommendations for this deployment with respect to firewall configurations have not been followed then there is a greater risk. This information is valuable for deployers to understand where there are areas of improvement in their systems configurations. The information gained from this assessment can be used by the security expertise consultancy to provide advice on best practices and recommended configurations.

The next available tool is the pen testing activity that provides an additional layer of surety for those deployments in that they can see if there is a need to harden or review their security implementation for multi-domain services. Information from these tests can be used to assist the security expertise consultancy to recommend changes and also to pass the information about possible new exploits to the developers to implementation the recommendations in the next versions or even provide a patch that the VST can then advise the community to install.

In order to transfer the knowledge gained from analysing multi-domain systems to the developers and deployers, the concept of a cookbook was created. The cookbook is a guidebook, it provides tangible information on what areas they can get consultancy from the security expertise consultancy service. It also tells them from a deployment perspective, what areas their tools map to so that their questions can be tailored to specific areas and also to help them understand what areas are affected and have a security implication for them. The current version of the cookbook deals with issues discovered during the SA2T4 teams investigation of multi-domain tools.

The underlying component that links all the outcomes from the various tools is the security expertise consultancy service. It uses the pen tests and risk assessments to provide assistance to specific deployers and then uses this information to pass onto the VST to inform the rest of the community and developers to the possible solutions and or workarounds found
3.1.4 What are we doing to improve the security of the multi-domain service tools?

Providing developers with the knowledge to develop secure code is the first step in protecting applications. As such there is a need to provide support that directly assists the developers of services in order to mitigate the threats they could be exposed to when they become production systems. This is achieved by providing a secure code training course. Some of the topics covered are to make developers aware of the effect of SQL injection, how to prevent it and how to prevent privilege elevation in systems by checking user’s rights before calling specific instructions.

Performing pen tests in addition to checking deployments, is a valuable litmus test from a security perspective for developers to ensure that they have not left backdoors for maintenance open and also that they have done everything to secure the system.

As in all areas the security expertise consultancy provides a conduit for security related questions raised through the secure code training sessions and the pen testing activities.

In the next sections, each of the tools, process raised in the current chapter will be covered in detail in order to provide insight into the multi domain security teams approach to multi layered security.

3.2 Security Expertise Delivery (SED)

3.2.1 Cookbooks

The task of reviewing code for new and existing systems is a critical aspect of a peer review process and as such should form an integral part of the development of all systems. A process of reviewing systems and in some instances the code has been completed whereby specialist teams reviewed either the development code and or had hands on access to production systems in order to test them for vulnerabilities.

The compendium of issues as found through the analysis of the code and various multi domain systems resulted in the cookbook series of documents containing security issues related to four GÉANT development projects involving MDS tools namely:

- AutoBAHN
- I-SHARE
- perfSONAR
- cNIS

The security issues listed for each of the tools were evaluated according to the severity of the issue found. For each of the issues highlighted a solution is proposed as to how the issue can be solved or at least mitigated. In order to provide valuable security based solutions for developers, each issue is provided in a structured format allowing developers an ease in identifying critical issues and also the ability to concentrate only on those issues that affect their systems. In some instances where issues were common across all the tools, the security team provided tailored solutions for each of the individual tools so as to ensure that solutions were tailored to the tool in question.

To provide an easy to navigate document to quickly identify relevant areas the document was broken down into the below structure as shown in Figure 2, listing firstly the “Problem”; which was a short description of the problem. This was followed by “Activity” which is the GÉANT project activity that this system was created under in order to pinpoint ownership of the tool and therefore the associated problem. Next, was the issues severity as listed under “Severity”. The severity was broken down into three levels, Critical; requiring urgent remediation, Warning; meaning that there is a potential risk of exploitation and the issue needs to be resolved in the next release and lastly, Reminder; that inferred that whilst the issue was not urgent it still needed to be investigated and eventually solved.

Whilst highlighting issues within the various systems, the primarily goal was to provide security support to developers and this was delivered by listing a ‘Proposed Solution’ for every issue noted. The proposed solution provided a high level description of the issue. Adjoining this, a technical solution was included to assist developers in implementing the proposed solution.

In order to track the ‘effectiveness’ of the cookbook, the team performs a cyclical security review of the systems highlighting new vulnerabilities and also checking if existing exploits have been remedied. The solution implementation auditing is achieved by using the “Reported date” against each item and at the review performing a sample analysis on the tool in question. This assists in ensuring secure code compliancy and also shows the mitigation strategies are been implemented when performing risk assessments against GÉANT software systems.

An example of an issue listed for the Bandwidth on Demand service tool can be found below in Figure 2.
3.2.2 Security Consultancy Requests
Within security task an on-demand security expertise consultancy service has been established to support security queries for the multi-domain services. Its main goal is to help the administrators and the developers of multi-domain services in dealing with security aspects that relate to their project work specifically the design, development and operations of MDS tools. The consultancy service does not provide support for handling multi-domain security incidents that affect GN3 services and most importantly does not replace the standard security support that project participants receive from their NREN security administrators or CERTs. It is used as a parallel to (or after) tackling the incident support mechanism.

The areas of consultancy that are covered include:

- Network Security at each network layer
- End Host Security and management for popular operating systems
- Application security at each point of the lifecycle of the application
- Monitoring of network or systems
- Other security areas (e.g. security policies and incident handling procedures)

SED consultants are comprised of specialists from NREN CERT teams, therefore making them the ideal candidates to assist in this environment with security issues. Each area of security expertise is assigned two specialists in order to ensure that there is always an expert able to answer consultancy requests for any specific area of security queries. Consultancy requests can be submitted via email to a dedicated mailing list that is controlled by a moderator responsible for ensuring a prompt response to all queries and also to assign the query to a security specialist. All responses to the mailing list are handled based on a first come first serve basis and within agreed time frames.

SED consultancies provide a valuable mechanism to improve the security knowledge amongst the GÉANT community and to assist GÉANT staff in further securing their infrastructure.

3.2.3 Secure Code Training
There is an age old saying that states that prevention is better than the cure. This concept is the impetus in the creation of proactive and not only reactive mechanisms been designed within SA2T4. The result of this process was the provisioning of the Secure Code Training (SCT) for MDS Tools developers. The main goal of this activity is to raise awareness of threats resulting from security bugs embedded within the source code of MDS systems. In addition to this it is a means by which the developers are exposed to best practices and principles that will lead to them producing software that contains fewer programming errors.

To date, SCT courses have been delivered every year since 2010 and the next will be held in the summer of 2012. The training sessions are limited to a maximum of 18 participants per course in order to ensure individual attention can be given to all attendees. Each SCT course covers several topics such as static source code analysis. Topics included in the course are common software bugs (like buffer overflows or XSS), and programming languages like Java which is the programming language used in the majority of the MDS tools. In addition to technical secure code programming topics, the training includes topics on programming best practices as delivered under the auspices of the Software Governance activity (SA4). This link-up ensures that attendees are
given the knowledge on how to secure their code, but also guidance in implementing best practice in their development projects.

The content of each training session are adjusted according to two main criteria. Firstly, the SCT team takes as input security issues identified during their security assessments of MDS tools. This allows attendees to be able to apply their knowledge directly on the actual issues as found in the MDS. Secondly, each new course incorporates feedback from the previous sessions to ensure that the course grows in line with the needs of the developers in the community. The current SCT courses include an interaction aspect in the form of practical workshops, exercises, code analyses and open discussion segments that allows participants to interact with each other and the presenters to assist in knowledge transfer.

The SCT courses besides allowing for cross pollination of knowledge, allows for the integration of two sides, the security experts and the service development teams. This creates an environment in which all can learn the goals and needs of each team and also how one can help the other to achieve secure systems, The Holy Grail that results from this investment is having fewer software errors and therefore security vulnerabilities in the future.

3.2.4 Penetration Testing / checklists
Penetration testing or pen testing is one of the forms of security assessments that are defined by the National Institute of Standards and Technology (NIST) as “security testing in which evaluators mimic real-world attacks in an attempt to identify ways to circumvent the security features of an application, system, or network” [17]. SA2T4 offer penetration testing as a free standing service to interested parties under the auspices of SA2T4’s multi-domain security team. Penetration testing is conducted from the point of view of a potential attacker, which makes it an excellent measure to complement other methodologies. This approach is referred to as “black box” testing where the tester is not given any extra information on the tested application and its environment except for its URL, or possibly a standard user account. In some instances it may be conducted as “grey box” testing where one has partial knowledge about the analysed system.

There are numerous specifications and definitions for performing penetration testing. An example of a comprehensive penetration testing scheme, taken as the initial approach may be found at [18]. As the MDST does not have the full resources of a commercial provider a subset of the steps within the full testing scheme are performed. The items covered during an MDST Penetration test are the initial Information Gathering then moving onto Network Mapping, checking for vulnerability identification and completing the test with an attempted penetration within the assessment stage. Not covered during the tests are the more intrusive items such as the compromising of remote users and sites as well as maintaining access to the systems and the covering of tracks.

One of the major advantages of performing penetration testing is the high degree of realism with respect to how systems respond to and are protected from attacks. Whilst penetration testing can be comprehensive from an external perspective, not all security flaws are exploitable from the external network. The SA2T4 apply compliment the pen testing activity with additional forms of security assessments so as to create a complete view of the security stance of the deployed systems.

The penetration testing activity is an optional service and provided on an on-demand basis. The Service Development Team (SDT) representatives may apply for the pen test in a similar way to the SED consultancy case (see the chapter 3.2.2).

The activity recently performed a comprehensive audit of I-SHARE that also included performing a pen test against the application of which the results were fed back to the development team resulting in major improvements from a security standpoint of the application. The test generated a substantial amount of interest from the Service Development Team for performing additional tests against other applications in the multi-domain services suite of applications.

An additional sub component of the pen testing activity has been the creation of a series of pen testing checklists. These checklists focus on a particular facet that is applicable to one or more MDS tools, for instance authentication forms, uploading files functionality, SSL/TLS configuration, information disclosure, XSS and SQL Injection. At present the activity has ten checklists that can be used to perform specific pen tests against. These checklists are viewed as live documents and as such are regularly been updated by the experts within the MDST.

3.3 Multi-Domain Security (MDS)
3.3.1 Risk Assessments
The risk management process is one of the most important processes in security management and also in project management. Of the risk management concept, the aspect of risk assessments is the most well-known and
A risk assessment is a method of identifying and managing threats that may negatively impact an organization or project. The concept behind a security risk assessment is about understanding the risks that users face from using a specific piece of software that is visible outside of their organisation or control. A risk assessment gives you a clear and measurable picture of the level risk that an infrastructure administrator and system deployments face.

Within the SA2T4 task, risk assessments can be performed by either of two sides; the SED experts and or the developers/deployers of MDS tools. The main task of the SED experts is to define appropriate checklists for every MDS tool, including its platform and environment, and then to keep all checklists up to date. These checklists form the basis of the risk assessment system. The experts are also involved in helping developers/deployers minimize the risk they face by providing feedback post each risk assessment by means of supplying advice on mitigation strategies for each risk item. The responsibility of developers/deployers is to make periodical risk assessments of the system using the appropriate checklist.

The risk assessment process is iterative and is started at point 1 in Figure 3 by selecting the appropriate checklists which are tool specific and contain a series of questions that will enable a risk score to be calculated for the system. Performing a risk assessment can be triggered by various events such as the finding of new vulnerabilities or the release of a new version of the software. Web forms are used to enable users to access specific checklists. An MDS tool administrator, developer or deployer can use the online risk assessment system to perform a risk assessment using one of the pre-defined checklists that are applicable for their application. Each checklist is annotated so as to assist users in selecting the appropriate checklist.

Once the form has been completed and submitted the system will automatically calculate the risk score based on the inputs provided. As risk scoring is a numerical scoring system, the lower the risk scores the lower the risk exposure the MDS tool has. To ensure those that perform the risk assessments are able to ascertain the evolution of their changes to the risk of their systems, authorized entities can review previously stored results.

To measure the risk we use a numerical grading (0 to 5 points) scale that has been designed specifically for use by the project. The analyzed MDS tool is then assessed in several categories:

- The MDS tool environment, which could be the OS, Web server or the attached database server.
- The configuration of the platform itself.
- The MDS tool in use.

Each of the above mentioned categories is evaluated separately resulting in a specific score which is then...
combined with the other categories to provide one single risk exposure score for the MDS tool. The risk assessment caters for the provision of additional points been added for particular flaws. An example of this in practice may if the provided version of the underlying OS contains known security bugs one point is added. Additionally if those bugs are known as highly critical then two points are added. If the particular component contains an additional security measure such as the use of an IDS then one point is subtracted from the sections score. To ensure that the concepts as found in performing risk assessments are applied throughout the project it is proposed that performing a risk assessment becomes mandatory. This approach goes a long way towards having those that create the systems as well as those individuals responsible for supporting them understand the levels of risk their systems are exposed to in a production environment.

3.3.2 Multi-Domain Incident Handling
Handling security incidents are an everyday activity for CERT (Computer Emergency Response Teams). The project has delivered services that are truly multi-domain in that a logical system, for instance AutoBAHN can span multiple domains. In this case it is possible to have a security incident that impacts one part of the overall system yet puts at risk the remaining segments. It is therefore important that a mechanism exists that specialises in providing a commutation conduit for multiple teams investigating a single incident. The resulting output of this research was the creation of the Multi-Domain Security Incident Workflow.

This workflow defines the interaction between various parties when a multi-domain incident has been reported. The approach that was taken in creating the workflow was that multiple teams may be involved in investigating an incident; therefore this workflow must facilitate inter-team communication.

The initial step of the workflow deals with incident notification where any third party may report a potential security incident to the Virtual Security Team. The initial step can be found in the workflow diagram in Figure 4. The Virtual Security Team (VST) is a logical grouping of security experts that will assist teams in notifying affected parties and assisting when additional security expertise is required. By using the VST approach it enables specialist expertise to be added to the resource pool at short notice and also ensures that no single team or individual carries responsibility for monitoring the ticketing system. The VST are alerted to incidents in the Notification stage of the workflow. At the evaluation stage it is the VST’s responsibility to analyse the notification in order to determine if it relates to a multi-domain incident. If it does not pertain to a multi-domain GÉANT service it will be forwarded onto the relevant NREN CERT. If the incident does affect a GÉANT service, the VST will notify all NREN’s CERT teams involved in the service of the incident to raise awareness of the possible impact on their MDS deployments. Should assistance be required during the evaluation phase, the VST will provide a consultative service to the respective parties upon request from the NREN CERT Team.

The next stage of the incident handling workflow is the where the bulk of the activities either reside with the respective CERT teams of the NREN or with the development team of the service. Before reaching this stage, the VST will perform a triage on the incident in order to establish whether this is a security or a software bug related incident. The important aspect of the investigation and resolution stage is that the NREN’s Operations and Service Development Teams internal processes and procedures apply. The VST merely acts as a conduit between the various teams if and should they be required.
The final segment of the workflow deals exclusively with the outcome from the respective operations and development teams. It is the responsibility of the affected parties to provide the VST with information pertaining to the resolution and mitigation of the incident. This could be in the form of advisories, bulletins or reports that the VST will assist in disseminating to all domains involved in the multi-domain service. The VST will archive the information into its internal knowledge base that will be used to create reports and perform a trend analysis of incidents reported for each of the tools. The information from the knowledge base will be a valuable source of incident and mitigation data for CERT teams and the development teams in order to assess the number and types of incidents handled over a particular period of time.

The Multi-Domain Security Incident Handling Workflow is a process that must be followed to ensure that all the
relevant and correct people are kept informed when dealing with an incident. The objective of the workflow is that it provides a standardised approach with respect to the handling of incidents in a multi-domain services environment.

3.3.3 Multi-Domain Security Team Knowledgebase

The multi-domain security team through its ticketing system has a valuable repository of incident and mitigation data. In order to leverage this information for the improvement of services and as a source of incident data, the multi-domain security team have extended their use of RT (Request Tracker) to be used as a knowledgebase through implementing the add-on RTFM (Request Tracker FAQ Manager).

As a pool of possible solutions when similar type incidents are reported, the knowledgebase provides the multi-domain security team with information that could be used to provide security expertise services throughout the project when deploying security systems, dealing with new incidents and protecting and creating secure code for multi-domain services.

3.3.4 Fire Drills

From an IT perspective, a fire drill exercise is a simulation of a computer security incident. Its main purpose is to test the incident response procedures especially in the field of the communication between NREN’s CERT/Security teams. In other words, a fire drill has been designed to test the communication, policies and procedures in response to various computer security incidents and to identify where further planning and process improvements are needed for those NRENs participating in multi-domain services. The fire drill exercise is performed as a simulated event with no real-world effects on production systems.

The main objectives of the fire drill activity are to:

- Test incident response and incident handling procedures,
- Test the effectiveness of the cooperation between different teams,
- Improve the points of contact in the involved teams,
- To improve the cooperation between incident response teams.

The activity have developed various types of fire drill exercises designed especially to test incident handling efficiency in the face of different kinds of incidents such as DDoS attacks, spam, phishing site hosting and system compromises. This ensures that the widest range of possible incident types are covered in the fire drill exercises. The fire drills are carried out on a periodic basis to test and improve the effectiveness of communication within the teams that are responsible for handling multi-domain security incidents.

4 A Comparison of Existing Security Processes

How does SA2T4’s approach to security compare with existing research and development (R&D) projects? In our approach we have proposed the security-in-depth principle as found in several R&D projects. The approach taken in two particular projects will be described in more detail in the following section in order to create a comparative analysis of the various methodologies. The first, Clusterix, was built several years ago, when the concept of applying complex security mechanisms was not common practice in R&D projects. The second project, PL-Grid, has recently been completed and it took the approach of applying coverage of the IT security problem using technical and organizational measures. Apart from the above mentioned cases, complex security strategies have been applied e.g. to the “Virtual Laboratory of Interactive Teaching” [14] the “National Data Storage 2” and “Advanced architecture of the Integrated IT Platform with high security level” [15] projects.

4.1 Clusterix

The primary goal of the Clusterix project (implemented between 2003 and 2006) was to build a new generation distributed PC-cluster. This was to be based on the Polish Optical Network PIONIER in order to deploy a production capable grid environment that consisted of local PC-clusters using 64 and 32-bit Linux machines to be located in independent centres across Poland [12].

The projects architecture had security-oriented approach and a separate work package was devoted specifically to security issues. Defined threats were associated when the possibility of dynamically attaching and detaching additional (aka “dynamic”) clusters to the basic infrastructure was possible. Security was implemented in a multi-layered manner but from a review process there were no software security assessments as well as no penetration testing of the whole infrastructure was performed. No specialisation security training was delivered to any of the individuals involved in the project. One of the downsides of the used security measures introduced potential bottlenecks within the local clusters, for example a local firewall controlling all the network traffic from and to local clusters and as a result contributed to the increased complexity of the architecture. The Clusterix project was an example of an overestimation of the security requirements not been considered against usability.
4.2 PL-Grid

The Polish Infrastructure for Information Science Support in the European Research Space PL-Grid project operated between 2009 and 2012. The goal of the project is to provide the Polish scientific community with an IT platform based on grid computer clusters, enabling e-science research in various fields. The created infrastructure is both compatible and interoperable with existing European and worldwide Grid frameworks. PL-Grid is the Polish NGI (National Grid Initiative) [16].

The application of the security-in-depth approach started with addressing organizational and procedural issues for the project. Similarly to Clusterix, a separate work package for security was created. The ‘Security Centre’ for the project was established with the Security Coordinator as the head that has responsibility for all security-oriented activities within the project including communication between the centre members. The Security Centre represents the Polish grid as NGI PL in the European Grid Initiative project. It was involved in every stage of development of the infrastructure and the development of the software and undertaking both reactive and proactive actions. The first task of the Security Centre was to review current security standards that were applicable to the project and to prepare and deploy suitable security policies and procedures for possible implementation. Tailored security requirements for administrators and programmers including operational policies such as those concerning security monitoring and assessment were issued. Other Security Centre tasks included the development of a methodology for grid-specific penetration tests which was based on well-known security testing approaches. Other tasks were the keeping of a separate PKI with a Certificate Authority as the main component of Grid authority management, providing input to software deployment procedures, performing penetration testing of the infrastructure and implemented services and finally implementing source code reviews of the software and operational actions such as incident handling [13].

What differentiates the handling of security in the PL-Grid project from other mentioned projects is that the security assessment the software packages were compulsory and officially built into the development lifecycle. The security centre had carte blanche in rejecting a releasing if the software contained security bugs that had been assessed as critical (the Project Management Board could enforce the release, but would accept the risk for any potential security breach resulting from that release). The Security Centre proved to be successful in their role as during the entire project lifetime only two security incidents occurred. Areas of improvement for this setup are implementing dedicated security training for administrators and in particular developers. The optimization of the procedures for disseminating knowledge on certain security vulnerabilities or achieving full consistency of security reporting as issued by assessment teams from different centres could be an additional area of improvement in the project [13].

5 Current Outcomes and Future Results

At this stage of the GÉANT project SA2T4 is able to provide tangible results for the work done in the activity. Due to the infancy of some of the activities undertaken, it is expected that in the latter stages of the GÉANT project, additional results will be realised, especially those outcomes from the risk assessment activity (see chapter Error! Reference source not found.).

Current outcomes from operational SA2T4 activities:

1. 30 GÉANT developers have been trained in how to apply secure programming best practices for application in the development of MDS tools. Based on feedback from the attendees the majority noted that their awareness about potential security problems that may occur due to programming mistakes had been insufficient going into the training session and had significantly increased due to the participation in the training course.
2. 27 security areas of improvement have been identified involving MDS tools and the prescribed solutions have been provided in the Security Cookbook (7 for AutoBAHN, 5 for i-SHARE, 11 for PerfSonar and 4 for cNIS).
3. 16 distinct security issues were identified during the first security audit of an MDS tool that included penetration testing plus an automatic and manual source code review. Of these 16 issues 2 where highly critical and 9 where moderately critical bugs.
4. 5 security issues such as the sending of unencrypted passwords across the network, applying insecure access settings in JNLP files, using unsafe defaults in the interfaces as well as applying insecure direct references to sensitive operations were found in the same audit. These issues have the potential to affect all of the MDS tools and will be added to the next version of the Security Cookbook.
5. To-date the GN3 multi-domain services have not suffered any security incident.

It is expected that the designed and implemented security measures, when combined together and applied
6 Summary

The approach taken by SA2T4 / the multi-domain security team in applying the defence-in-depth strategy has proven to be highly successful. Whilst the concepts are not new, the principle of applying these concepts in a structured and goal driven format has given rise to a truly unique and all-encompassing formula for protecting research and development networks.

By setting a benchmark for security in research and development projects, SA2T4 have set a precedent that could assist other similar projects when implementing a security framework for their networks throughout the global networked community.

In an age of ever increasing Internet based threats, it is critical that researchers strive towards protecting their systems to ensure that the benefit of these networks can be gained from all that use these networks. The multi-domain security team have risen to the challenge of achieving their goal of protecting multi-domain services in the GÉANT project. The activity will continue to evolve their security concepts to meet future threats and to ensure that best practice is at the forefront of security in the GÉANT project.

References

[1] EC Factsheet - GÉANT’s role at the heart of the EU’s e-Infrastructure strategy, http://www.geant.net/Media_Centre/Media_Library/Media%20Library/geant3_research_infra.pdf.
Biographies

Gerard Frankowski received the M.Sc. degree in Computer Science from the Poznan University of Technology (1999) with specialization in Intelligent Decision Support Systems. He works in PSNC since 2003 in the PSNC Security Team as a computer system analyst. He took part in several European and national R&D projects involving security of HPC technology, networks and systems: EGEE-II, EGEE-III, GN3, Clusterix, SGIGrid, PL-Grid, National Data Storage, Polish Platform for Homeland Security, Integrated IT Platform for the Polish Police and others. His security research interests concentrate on vulnerability research, secure programming and penetration tests techniques.

Tomasz Nowocień graduated from the Poznan University of Technology (1998) as a computer science engineer. In 2000 received M.Sc. degree in information technology specialized with designing of information systems. For several years worked as application programmer, developer and designer. Form 2002 works for PSNC as PSNC Security Team member. Since 2005 he became security system analyst and from 2009 obtained a Information Security Supervisor function. He took part in several national and European projects like: EGEE, PL-GRID, GN3, Polish Platform for Homeland Security and others. His interests are focused especially on security management, risk analysis, administration of secure platforms.

Wayne Routly graduated from the Port Elizabeth Technikon in 2004 with a M.Sc. Information Technology. After graduating he joined the Nelson Mandela Metropolitan University as their Security Engineer and Information Security Officer. After moving to London in late 2005 he joined Fabric Technologies as their Infrastructure Engineer and Team Leader, before joining DANTE in 2008 as a Security Engineer, and acquiring his CISSP certification. He is the Security Engineer and Security Officer at DANTE and is responsible for security in the GEANT network. Wayne leads the GEANT project Multi-Domain Security Team task which is responsible for securing Multi-Domain Services.