Grid Services Security
Vulnerability and Risk Analysis

Dr Linda Cornwall CCLRC (RAL)
LCG-OSG-EGEE meeting, CERN, 19-20th June 2006
Contents

• Why we setup the Grid Security Vulnerability Group
• Starting up the GSVG
• Initial strategy and disclosure policy
• Vulnerability Task in EGEE II
• Setup of the GSVG in EGEE II
• Disclosure policy in EGEE II
• Risk Assessments
• Other vulnerability related activities in EGEE II
Why we set up the Grid Security Vulnerability Group (GSVG)

- A lot done concerning Grid Security Functionality
  - Authentication, Authorization
- Not much being done to ask “Is the Grid Secure”
- We know the software isn’t perfect
  - Some vulnerabilities are in the process of being fixed
  - Some are probably waiting to be exploited
- It will be really embarrassing if when the Large Hadron Collider comes on line at CERN we get a serious attack which prevents data being stored or processed
- Hackers Conference HOPE mentioned Grids
  - Unfriendly people without credentials aware of us
  - Cannot rely on security through obscurity
- Real Grids are being deployed
  - No longer a research/proof of concept activity
Starting up the Grid Security Vulnerability Group (GSVG)

- Started up in 2005 with part of my time and a few “best efforts” volunteers
- Aim to inform developers and deployment people of vulnerabilities as they are identified and encourage them to produce fixes or to reduce their impact.
  - Aim to keep grids deployed, avoid incidents, improve Grid Security with time
- Joint effort between Large Hadron Collider Grid (LCG), UK particle Physics Grid (GridPP), and EGEE
- Some people were worried about the legality of not making info available to all
- Defined a policy and strategy for carrying out the work
- Got project management approval of our terms
Initial strategy and disclosure policy

- Log issues, set a Target Date (TD) usually 45 days
- If issue is still open on TD, distribute information including risk assessment to LCG security contacts
  - Vulnerability group does not make issues public
  - Security contacts considered internal
- Includes both software and deployment issues
- Issues entered by anyone
  - Developers enter issues they know about
  - Includes information from internal knowledge
  - Includes impact of missing functionality which will not be available in the short term
Achievements and problems

- Collected 84 potential issues
  - 18 closed (8 fixed, 9 invalid/not a problem, 1 duplicate)
  - 6 fix rolling out/ awaiting the next release
  - 61 reports sent out to the LCG security contacts
- Haven’t been seeing enough issues closed by TD
  - Not enough priority given to investigating or fixing issues
  - TD set to a default (problem prioritizing)
  - Looks worse than it is
    - Some issues are missing functionality
    - Many not directly exploitable
- Some people who would have been very useful in the team didn’t want to join
  - Despite project approval
In EGEE II there is funded manpower for the “Grid Services Security Vulnerability and Risk Assessment” Task 😊

The aim is “to incrementally make the Grid more secure and thus provide better availability and sustainability of the deployed infrastructure”

- This is recognition that it cannot be made perfect immediately

Handling of Vulnerability issues is the largest activity in this task

- Which continues to deal with specific issues
- Continues not to be confined to software vulnerabilities, but also includes issues arising from lack of functionality
The GSVG in EGEE II consists of

- **Core Group Members**
  - Run the general process

- **Developers from the various development Clusters**
  - Can confirm/check information on issues and fix issues

- **Risk Assessment Team (RAT)**
  - Carry out Risk Assessments

- **RAT people are security experts, experienced system administrators, deployment experts and developers**
Process of the GSVG in EGEE II

- Issue logged in Database
  - Anyone can submit an issue
  - Only GSVG members can read or modify
  - Issues can also be submitted by e-mail
- Issue is allocated to Risk Assessment Team (RAT) member
- RAT member
  - Checks information – need to work with appropriate developer
  - Carries out a Risk assessment
- 2 other RAT members also carry out Risk Assessment
- Target Date (TD) set according to Risk
  - To improve prioritizing
- The issue is then allocated to the appropriate developer
• We plan to move to a responsible public disclosure policy
• On Target Date, information on the issue is made public
  – Regardless of whether a fix is available

• This depends on management approval,
  – We need to prove we can do good Risk Assessments
  – Agree formula for setting the TD according to Risk
Main changes

- A risk assessment is carried out straight after issue is entered
- Improved Risk Assessments
- Target Date is set according to Risk
  - By TBD formula
- Information to be made public on the Target Date

- Good Risk Assessments and setting of TD according to risk is key to making the improved process work
  - Which effectively prioritizes issues
Currently establishing the best way to carry out Risk Assessments

- Risk Assessment Team (RAT) is working on this
- It is important that the risk assessment, and setting of TD is objective, not subjective

Common Vulnerability Scoring System (CVSS)

Location of issue on “Who can Exploit” “Effect” matrix

Any other method RAT members propose

What RAT members actually think
• CVSS is the Common Vulnerability Scoring System
• Takes account of various factors e.g.
  – Can it be exploited remotely
  – Access complexity
    ▪ (whether it can be exploited at will or only in certain circumstances)
  – Authenticated or not
• Modifies this according to temporal factors
  – Availability of exploit code
  – Availability of fix or workaround
• Modifies according to the environment
  – This could be considered the Grid environment
• CVSS provides a score between 0 and 10
  – Possibility of translating this to a Target Date
CVSS limitations

- It is designed for information systems, not Grids
- Does not take into account some factors important on the Grid
  - In particular, “who can exploit” is restricted to authenticated or not
- We cannot, for example, ignore that one Grid node can affect others
  - E.g. one sysadmin should not be able to setup a system that disrupts the Grid
Exploit/effect matrix

- Site security officers most fear an attack that gives access to the whole site
  - Especially if it can be carried out anonymously
  - DOS tends to be considered no more than medium risk
- A vulnerability that can be exploited by an authorized user is less serious than one that can be exploited without credentials
- We can’t ignore the possibility that credentials may be stolen
- Nor can we ignore that we may have a rogue sysadmin
  - 100s sites in 10s countries
  - Grid expanding globally
- This is considered useful
<table>
<thead>
<tr>
<th></th>
<th>Root Access</th>
<th>Local Account</th>
<th>Authz</th>
<th>Authn</th>
<th>No Cred</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>System info</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local grid service</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disruption</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confidential Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Restricts usage for certain applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unauthz usage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grid-wide Disrupt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impersonate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attack other systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Site Access</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Root Access</td>
<td><strong>Restricts usage for certain applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Propose 4 categories of risk

- Extremely Critical
- High
- Moderate
- Low
Extremely Critical

Examples

• Trivial compromise of core grid component
• Remotely exploitable issue that can lead to system compromise
• Root access with no Credentials
• Trivial Grid Wide DoS with no Credentials

• Target date – 48 hours
• Special process (details TBC) for handling
  – Alert OSCT +TCG immediately
  – Quick patch – in isolation with no other release, tested at the front of the queue
  – A.s.a.p. but in any case within 2 working days
  – Unrelated to release process
• Expectation – Very rare if ever
Examples

- Remote exploit against middleware service
- Spoofing – carrying an action on someone’s behalf
- Exploit against MW component that gives elevated access
- Grid-wide DoS?
- Target Date 3 weeks
Examples

• Confidential issues in user information
• Local DoS
• Potentially serious, but hard to exploit problem.
  – E.g. hard to exploit buffer overflow
• Race conditions that are hard to exploit

• Target Date 3 months
Examples

• Small system information leak
• Impact on service minimal

• Note – if 2 low risk issues could produce problem, this should be entered as a higher risk issue

• Target Date – 6 months
• If anyone thinks an issue is highly critical – all available RAT members should look at it

• Other than that vote –
  – E.g. If 3 look initially, and 2 say moderate, 1 low – set as moderate

• The Risk classification could change
  – Rise if information is available publicly or issue has been exploited
  – Fall if more information comes to light, e.g. part of the code not aware of mitigates problem

• Formula for setting TD is not for the RAT to decide unilaterally
  – We can propose
  – Need to agree with management
Issues arising from missing functionality

- These will continue to be entered
- We will carry out risk assessment
- But handle differently
  - Inform TCG of what we consider to be the risk
  - If appropriate – inform appropriate people for documentation purposes
  - But not set TD
• Advisory on issue is written when the risk assessment is carried out
  – By the RAT member the issue is allocated to, consulting other RAT members (if necessary) and appropriate developers

• Advisories available publicly on Target Date (or earlier if fix is available)

• Advisories will always include what to do
  – Solution
  – Patch/work around – which may reduce the service functionality
  – In worst case – advice to stop a service

• Advisories will not describe how to exploit issue
Other activities currently ramping up

• Providing best practise guidelines for developers

• Co-ordinating code walkthroughs/reviews
  – Largely to be commissioned by the Integration, testing and Certification Task in SA3

• The vulnerability task is involved in co-ordinating the specific (security) tests as part of the certification process

• The possibility of carrying out Ethical Hacking is being considered
Summary

- The GSVG is attempting to prioritize specific issues identified through risk assessments and setting Target Dates
- Other activities for reducing vulnerabilities
  - Testing
  - Code walkthroughs
  - Ethical hacking
- This work is important, the Grid has been described as a “Beautiful Amplifier” of vulnerability issues