Grid Security Vulnerability Group

Post mortem of the Proxy Generation Tool Vulnerability

Romain Wartel, SA1

EGEE-II conference, Geneva, 2006

www.eu-egee.org

Information Society and Media

Overview

- Initialization of the process
- Expectations
- Reality
- Conclusion
- Lessons learned

Main actors

- Akos Frohner – risk assessment, software expertise
- Alain Roy - VDT contact
- Andreas Unterkircher - ia64 and tarball
- Di Qing - Certification
- Ian Neilson – initial coordination
- Joni Hahkala - Security subsystem integrator
- Maarten Litmaath – Everything
- Maite Barroso/Nick Thackray/Antonio Retico - SA1 coordination
- Oliver Keeble: Integration/Certification/Testing/Release
- Robert Harakaly - Meta rpms and build advice
- Romain Wartel - Risk assessment, coordination and advisory
- Valerio Venturi - Patch developer
- Vincenzo Ciaschini - Package advice
- Other GSVG members – General advices, comments and support!

Initialization of the process

- **Globus released a security advisory for GT3/GT4:**
  http://www-unix.globus.org/mail_archive/security-announce/2006/08/msg00002.html
- **Akos Frohner (GSVG) picked up the problem immediately and contacted GSVG**
- **It then took approximatively one week to:**
  - Confirm that GT2 and VDT 1.2.x were affected
  - Realize that no patch would be available quickly for VDT 1.2.x
  - Discover all the LCG/gLite components affected by this bug
  - Understand that grid-proxy-init and myproxy-init were affected (but stealing proxy certificates was not possible)
  - Understand that voms-proxy-init was affected (and it was trivial to steal the proxy certificate of an arbitrary user)
  - Contact all the appropriate and available developers

Expectations

- Problem was simple and the patch was easy to implement
- Testing would be easy, as the component does only one thing
- No configuration change necessary for the upgrade
- Developers, integration/certification/release team were taking the problem very seriously
- All appropriate people already in the loop

Conclusion: A few days should be sufficient to release updated components.

Reality

- VDT team incomplete and priority was to fix VDT 1.3.x first
- Several developers involved at different sites, which involved significant communications delay
- Lots of people involved and a massive email flow was generated
- Nearly all our node types were affected (in principle)
- Different versions of affected components on many nodes
- LCG 2_7_0 affected, but the LCG build system has been replaced by the gLite build system
- Some ia64 builds caused additional problems and delay
- Many people away in the integration/certification/release team
- Several requests submitted (GGUS, etc.) about the problem

Conclusion

- It actually took three weeks (in total) to release updated packages (two days before our “target date”)
- We released our first security advisory
- Lots of people worked hard and overtime
- Maarten Litmaath handled the interactions with VDT/Globus, produced the patch and some packages, and some testing
- Oliver Keeble handled the build, integration, certification, testing, installation notes and release
- Good learning exercise

Advisory

- **It is available from: http://glite.web.cern.ch/glite/packages/R3.0/updates.asp**
- **It has been included in the release notes**
- **It has been sent to the LCG Security Contacts**
- **Disclosure timeline:**

2006-08-15	Vulnerability announced by Globus
2006-08-16	Initial response from the Grid Security Vulnerability Group
2006-08-16	Initial response from the VOMS developers
2006-08-18	Initial response from the VDT developers
2006-08-25	First updated sources received by the integration team
2006-08-29	All updated sources received by the integration team
2006-09-01	Updated LCG and gLite packages available
2006-09-04	Certification and release preparation completed
2006-09-05	Public disclosure

Lessons learned

- The overall aim is to be able to produce a patch quicker in the future
- When necessary, patches should be prioritized (based on the middleware version, affected component, architecture, etc.)
- Coordinating the vulnerability process took a lot more time and efforts than initially expected
- The GSVG process is being changed accordingly
- Most people involved believe they will be able to complete their tasks much quicker next time

Lessons learned

- **We identified the following roles regarding the risk management:**
  1. Confirm the vulnerability
  2. Assess the risk of the vulnerability
  3. Produce an advisory
  4. Repeat 1. and 2. if new information is revealed
  5. Consult and advise WRT to the risk involved by the vulnerability ex: Should we delay exotic builds?

Lessons learned

- **We identified the following roles regarding the vulnerability coordination:**
  1. Reply and keep the reporter informed
  2. Call the RAT for an audit of the bug and receive the initial advisory from the RAT
  3. Contact OSCT if the vulnerability is rated "extremely critical"
  4. Enter a Savannah ticket (and its JRA1 mirror)
  5. Identify and establish contact with appropriate development and deployment teams
  6. Deal with requests from external groups
  7. Establish contact and follow up with the integration/certification/release teams and pass them the advisory
  8. Ensure the process does not stall
  9. If there is suspicion no patch will be available to meet the target date, contact OSCT
  10. Publish the advisory