

# OSG Site Administrators Workshop Vanderbilt, 2010

## Security

infrastructure, certificates and recommendations

Igor Sfiligoi  
for the OSG Security team

# OSG Security

## Part One

# OSG Security model

A high level overview

# OSG Security model

- Multiple administrative domains; each Site
  - Decides how to run its own resources
  - Decides which users to support
- Federated trust
  - Too many users and too many sites to require each user to register at each site
  - Virtual Organizations (VOs) as a middle man
    - A VO trusts its own users
    - A Site trusts a VO

# Authentication structure

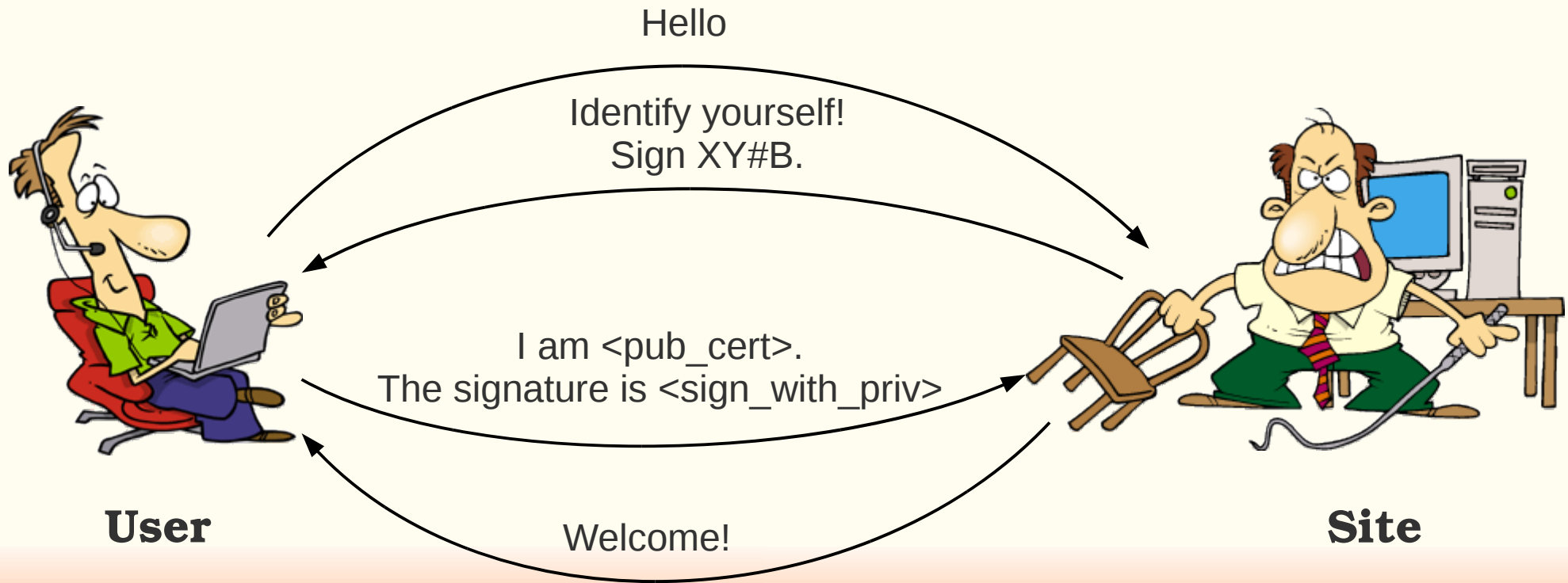
- Users want a single sign-on to run on all sites
  - Remember, they are not registering with all the sites
- Username+password cannot be used
  - That would require all sites to synchronize the password/shadow files -> not practical
- Public Key Infrastructure (PKI) used instead
  - In particular X.509 certificates and proxies
  - Sites only need to know the “user name”
    - PKI takes care of the security aspect

# PKI – x.509 certificate

- The user is issued a certificate, which is composed of 2 parts:
  - A public part, containing
    - The user name (also known as the **DN**)
    - Validity period
    - The public key
    - The signing chain (more on this later)
  - A private part (containing the private key)
- **The private part MUST be kept private**
  - The public part can (and will) be sent around

# PKI – How it works?

- User proves who he is by signing using the private key
  - The public key in the pub\_cert allows for verification



# PKI – What is a signature?


- A digital signature proves who you are
  - Because **only you own the private key**
- It is strongly correlated to the public key
  - Not enough time to go into technical details here, consult wikipedia if interested:  
[http://en.wikipedia.org/wiki/Digital\\_signature](http://en.wikipedia.org/wiki/Digital_signature)

# PKI – Signature validation

- The site must validate the signature
  - Else the user may just fake it!
- So the Site uses the public key sent by the user to do the validation
  - **But why should a site trust the public key sent?**
- The public key itself is signed by a trusted entity (in the signing chain)
  - By a **trusted** Certification Authority (CA)
  - The site must already have the CA public key **pre-installed** locally (typically getting it through the OS or the VDT)



# PKI – What is a CA?



Not all CAs  
are trusted!

- A CA is someone who issues certificates
- A **trusted** CA is someone who you trust to issue user certificates **only if** they know that user
  - i.e. User **X** cannot get a certificate with username **Y**
- There are relatively few **trusted** CAs in existence
  - At least compared to the number of users
  - Pre-installing their public keys is thus manageable
- A CA can also revoke a user certificate
  - By publishing its public key in a **Certificate Revocation List (CRL)**
  - **Make sure you download the updated CRL often!**

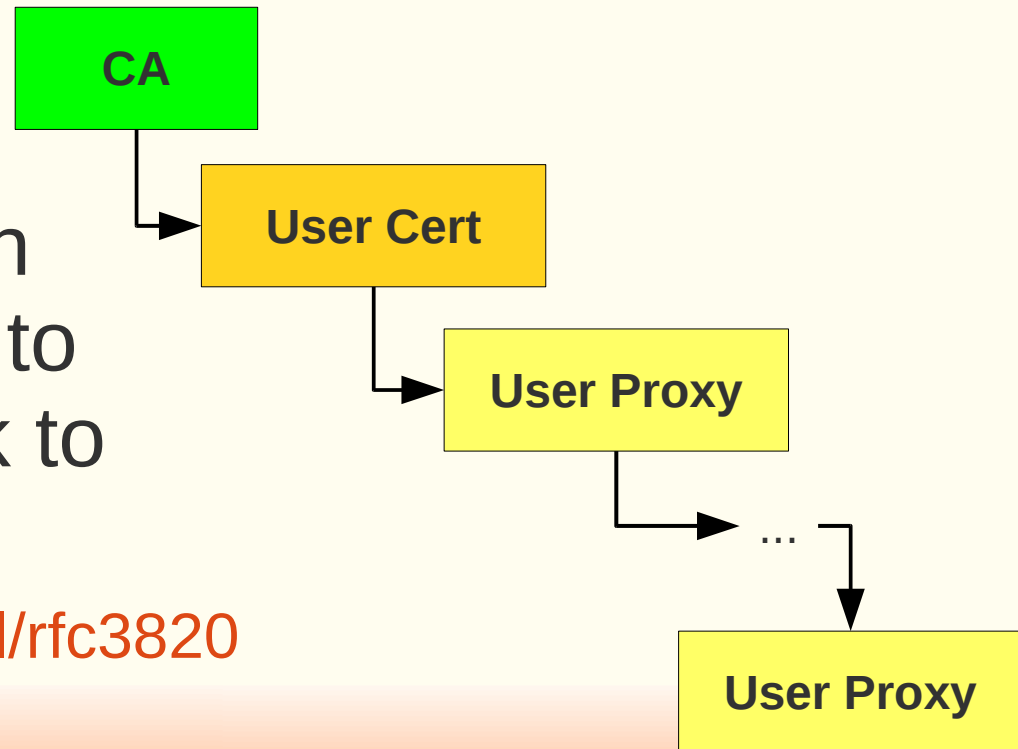


Self signed certs  
not issued  
by a trusted CA

# PKI – And what is a proxy?

- You probably have heard about proxies
- A proxy is just a certificate derived from a user proxy
  - Possibly many times!
- The signing chain contains the info to safely climb back to the CA

<http://tools.ietf.org/html/rfc3820>



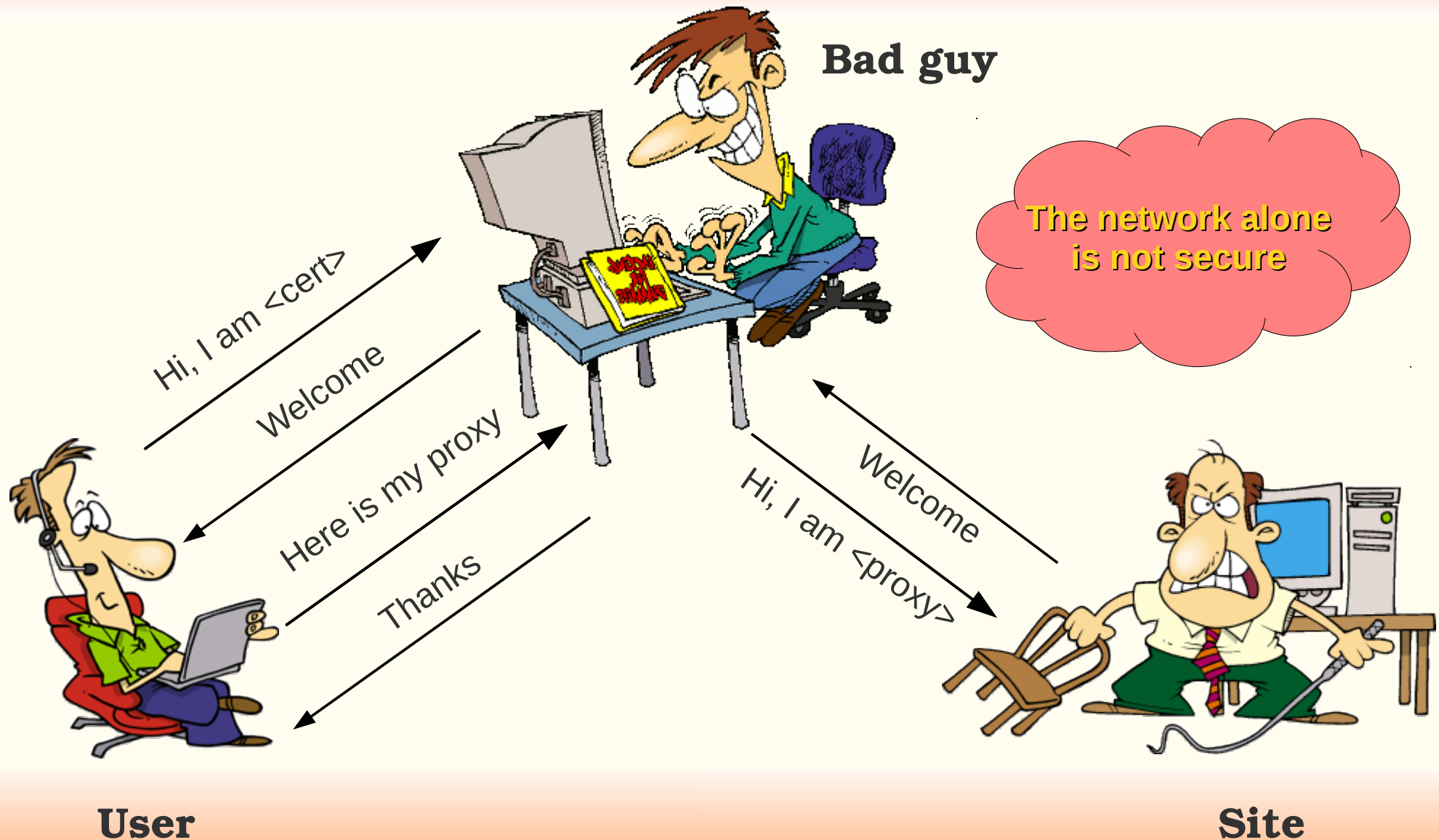
# PKI – Why a proxy?

- The user jobs may need to talk to a remote service when running on the worker nodes
  - But cannot access the user cert's private key!
- A proxy is thus sent (delegated) with the job to the worker node
  - **And the proxy contains a private key!**
  - So the job can impersonate the user
- Of course, delegating a private key is dangerous
  - Mitigated by the fact that proxy lifetime is short (much shorter than the user certificate one)

# PKI – Sites have certificates, too

- Security only if mutual authentication
  - The Site trusts the User and the User trusts the Site
- The Site must prove who he is to the User
  - Especially if a proxy is being delegated there!
- All nodes with services at a Site thus need a host or service certificate
  - Similar to a user certificate, but issued by a CA for a specific DNS host (can only be used on that DNS address)

# Example: One way authentication



# Authorization

- Just because someone can authenticate, does not mean a Site will authorize him/her to run on its resources
  - Authorization is a separate step
- The Site may also want to give different privileges to different users
  - The user must be mapped to a local security domain
  - Certificate DN -> (typically) UNIX UID

# VO-based Authorization

- As mentioned in the introduction, Sites trust VOs (not users directly)
  - Each VO will keep a list of user DNs
  - Through a service called **VOMS**
- OSG provides a list of trusted VOs and their VOMS servers
  - The Site needs to pick which VOs to support
  - Should always support the MIS VO (OSG operations)
- Users authenticate with a VOMS-extended proxy (voms-proxy-init -voms ...)

# Mapping

- OSG provides **GUMS** for mapping
  - Talks to VOMS servers to get the list of user DNs
- Site admin must decide the mapping
  - Still VO based, possibly based on VO groups
  - Either pool (**recommended**) or group mappings
- The admin must also create all the necessary UNIX accounts
  - Part of “*administrative autonomy*” principle



# Pool vs group mapping

- **Pool mapping** maps each user to a different UNIX username/UID
  - Something like *uscms0001,...,uscms2345*
  - May need lots of accounts!
- **Group mapping** maps all the users to the same UNIX username/UID
  - Something like *mis*
  - No protection between users
- Pool accounts recommended (unless VO explicitly asks for a group account)

# Additional reading

- List some interesting links here
- OSG Certificate page  
<https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/CertificateWhatIs>
-

# OSG Security

Part two

## Technical details

# Which CAs do we use

- DOEGrids CA (OSG Recommended)
  - <https://pki1.doe grids.org/ca/>
- CERN CA (Used by WLCG)
  - <https://ca.cern.ch/ca/>
- Fermilab CA (Fermilab-based users)
  - Converts krb5 tickets into certificates
- Foreign Country CAs
  - Each country has at least one CA
- Commercial CAs
  - Verisign, Thawte, GoDaddy, etc.

# CAs supported as a Site

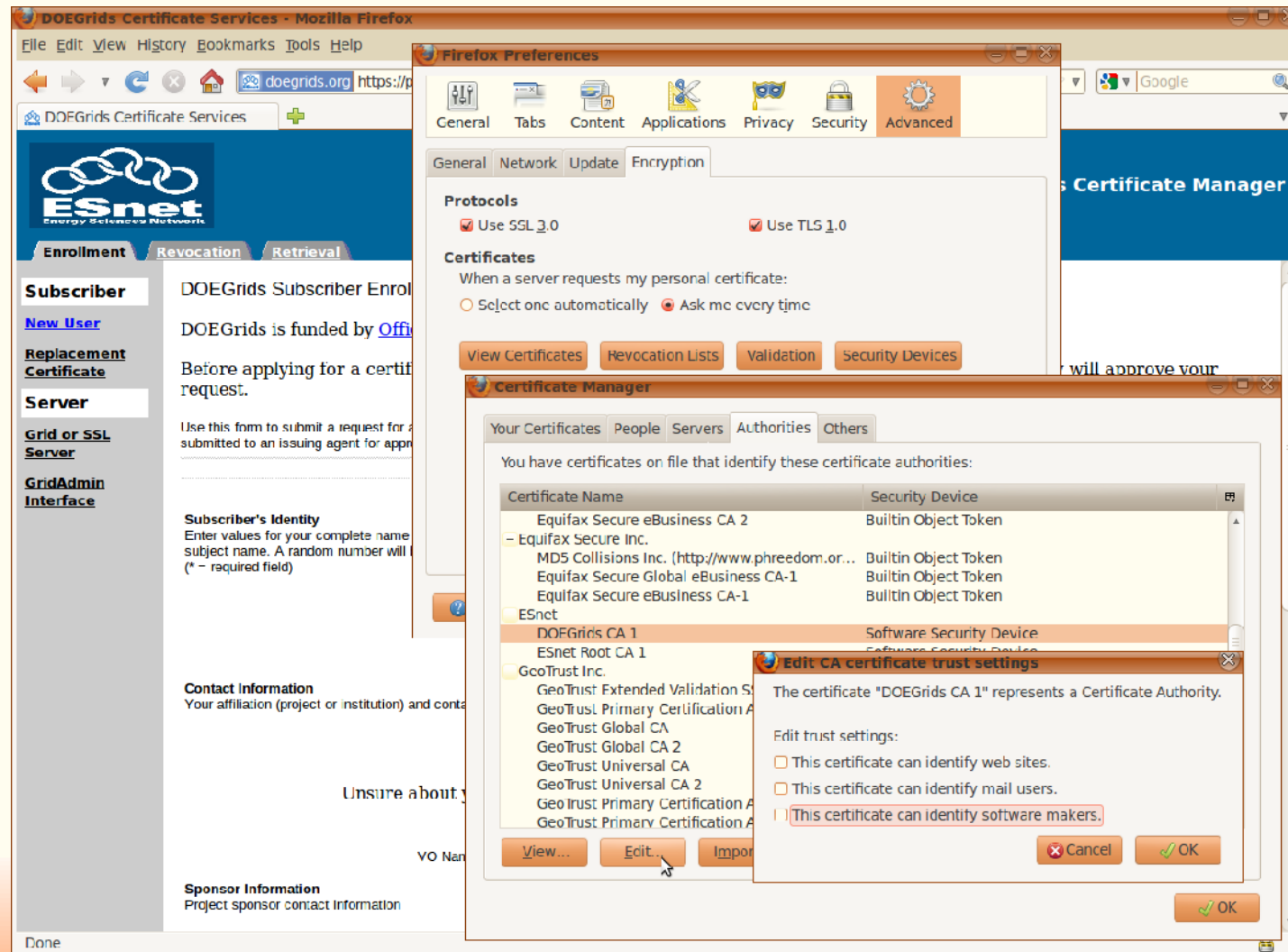
- OSG provides a list of trusted CAs known to be used by OSG-affiliated VOs
  - Get them through VDT  
<http://software.grid.iu.edu/pacman/cadist/ca-certs-version>
- You likely want to support all those CAs
  - But you are free to remove the ones you know are not being used
  - And add additional ones for non-OSG users
- Make sure you keep the CRLs updated
  - *fetch-crl*

# CAs supported as a User

- Two interfaces
  - Command line
  - Web browser
- Command line based on VDT
  - See previous slide
- Web browser mostly for Web pages
  - Commercial CAs come with the OS
  - The other CAs need to be imported  
<https://www.tacar.org/repos/>

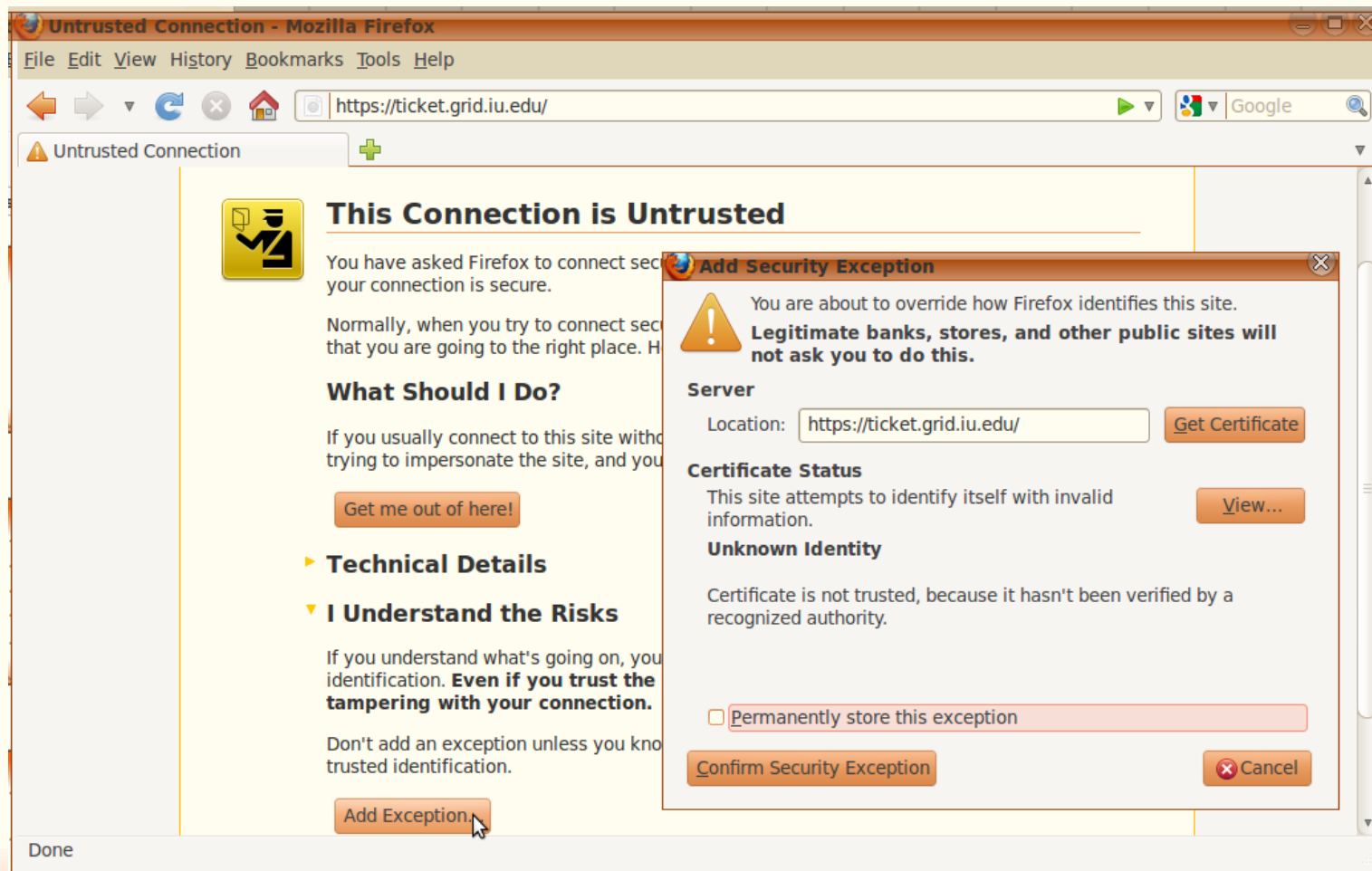
# Installed but disabled CAs

- Some browsers require explicit CA activation



# Browser security

- Do not override browser security!





# Requesting a certificate

- You likely want to use DOEGrids
  - Both for personal and service certificates
- You can request them either through the Web interface or  
<https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/CertificateGetWeb>  
through the command line interface  
<https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/CertificateGetCmd>
  - Command line easier for bulk requests (e.g. for service certificates)

# How long does it take?

- Getting a certificate can take days
  - So plan accordingly
- Delay due to security implications
  - Someone must validate your request/identity
  - A Registration Agent (RA)  
typically associated with the VO
- For user certificates you also need to register with the VO VOMS server
  - Procedure VO-specific

# Certificate format

- Two formats
  - .p12 – single file, containing both public and private part
  - .pem – two files, one for public (cert.pem) and one for private part (key.pem)
- .p12 and key.pem must be private to the user
  - No group or world read permissions!
- Can convert between them

```
openssl pkcs12 -clcerts -nokeys -in cert.p12 -out usercert.pem
openssl pkcs12 -nocerts -in cert.p12 -out userkey.pem
```

# Services accepting certificates

- Compute Element (CE)/ Globus Gatekeeper
  - Submits jobs to the local batch system
  - Handles user proxies
- Storage Element (SE)/ SRM/ gridFTP
  - Interface to the disk storage area
- Web server (optional)
- All of the above need a service certificate

# Additional reading

- List interesting links here

# OSG Security

Part three

## **Security recommendations**

# What is security?

- Security is much more than just technology
  - It is as much a social problem
- We have a secure system only if the participants act responsibly
- Malicious participants are obviously removed from the system
  - But a careless one can make almost as much damage!

# Knowing who is out there

- Knowing the participants is the first step
- Each Site should have a designated security contact
  - Interface to the rest of the Grid
- The OSG repository for such information is OIM  
<https://oim.grid.iu.edu/oim/home>
- Please make sure you keep your information updated there
  - You will need a user certificate to interact with it



# Security communication

- Security contacts will receive security notifications through e-mail
  - Please read and act upon them
  - The e-mails will be signed  
<https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/OSGSecurityNotifications>
- Know and possibly be in contact with your Campus/Institution cyber security team
  - They can provide invaluable help both in preventing and fixing security incidents

# Technical tasks

- Keep all the software up-to-date  
(mostly patching, but also upgrades as needed)
  - Operating system
  - System services
  - OSG/VDT provided software
- Keep security data up-to-date
  - List of trusted CAs
  - Associated CRLs
  - List of supported Vos
- Without, the risk of a compromise raises significantly

# Advanced technical tasks

- If possible, actively look for sign of a compromise
- Log files can provide a lot of info  
<https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/SearchLogFiles>
- Yes, it can take a lot of time
  - But it pays big dividend, if you can afford it
  - A security incident can make a Site unusable for weeks (or worse)



# What if you have a security incident?

- If you suspect a compromise,  
notify the OSG security team

<https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/IncidentDiscoveryReporting>

- Even if it turns out it was a false alarm,  
better safe than sorry (just don't do it every day!)
- Involving ALSO your local Campus/Institutional  
security team is also a good idea
  - Especially if you are fairly sure you have a problem

# Additional reading

- OSG Site Security Responsibilities

<https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/SecuritySiteResponsibilities>

- OSG Security Hands On Training

<https://twiki.grid.iu.edu/bin/view/Security/SecurityHandsOnTraining>

- Security Session at the 2009 OSG Admin Workshop

<http://indico.fnal.gov/sessionDisplay.py?sessionId=4&slotId=0&confId=2497#2009-08-06>

-

# Summary

- To be written

# Copyright notice

- These slides contain copyrighted images by ToonADay.com
- All such images have been licensed to Igor Sfiligoi for use in presentations
- Extracting such images and use them in other context is not permitted