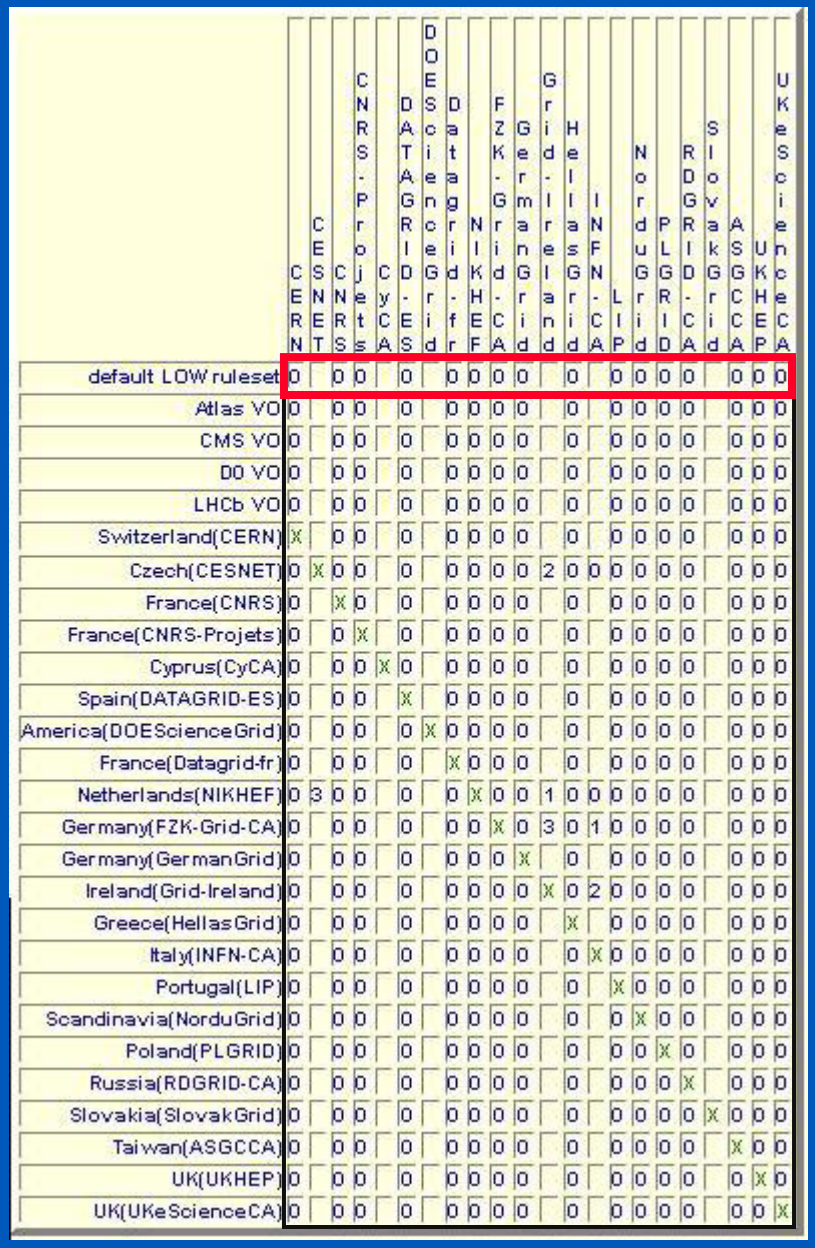
DataGrid WP6/CA

Passing the Default Ruleset

Trinity College Dublin (TCD)
Brian Coghlan



	CERN	CESNET	CNRS	CNRS-Projets	CyCA	DATAGRID-ES	DOESciencegrid	Datagrid-fr	NIKHEF	FZK-Grid-CA	GermanGrid	Grid-Ireland	HellasGrid	INFN-CA	LIP	NorduGrid	PLGRID	RDIG-CA	SlovakGrid	ASGCCA	UKHEP	UKeScienceCA
default LOW ruleset	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
Atlas VO	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
CMS VO	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
D0 VO	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
LHCb VO	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
Switzerland(CERN)	X		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
Czech(CESNET)	0	X	0	0		0		0	0	0	0	2	0	0	0	0	0	0		0	0	0
France(CNRS)	0		X	0		0		0	0	0	0		0		0	0	0	0		0	0	0
France(CNRS-Projets)	0		0	X		0		0	0	0	0		0		0	0	0	0		0	0	0
Cyprus(CyCA)	0		0	0	X	0		0	0	0	0		0		0	0	0	0		0	0	0
Spain(DATAGRID-ES)	0		0	0		X		0	0	0	0		0		0	0	0	0		0	0	0
America(DOEScienceGrid)	0		0	0		0	X	0	0	0	0		0		0	0	0	0		0	0	0
France(Datagrid-fr)	0		0	0		0		X	0	0	0		0		0	0	0	0		0	0	0
Netherlands(NIKHEF)	0	3	0	0		0		0	X	0	0	1	0	0	0	0	0	0		0	0	0
Germany(FZK-Grid-CA)	0		0	0		0		0	0	X	0	3	0	1	0	0	0	0		0	0	0
Germany(GermanGrid)	0		0	0		0		0	0	0	X		0		0	0	0	0		0	0	0
Ireland(Grid-Ireland)	0		0	0		0		0	0	0	0	X	0	2	0	0	0	0		0	0	0
Greece(HellasGrid)	0		0	0		0		0	0	0	0		X		0	0	0	0		0	0	0
Italy(INFN-CA)	0		0	0		0		0	0	0	0		0	X	0	0	0	0		0	0	0
Portugal(LIP)	0		0	0		0		0	0	0	0		0		X	0	0	0		0	0	0
Scandinavia(NorduGrid)	0		0	0		0		0	0	0	0		0		0	X	0	0		0	0	0
Poland(PLGRID)	0		0	0		0		0	0	0	0		0		0	0	X	0		0	0	0
Russia(RDGRID-CA)	0		0	0		0		0	0	0	0		0		0	0	0	X		0	0	0
Slovakia(SlovakGrid)	0		0	0		0		0	0	0	0		0		0	0	0	0	X	0	0	0
Taiwan(ASGCCA)	0		0	0		0		0	0	0	0		0		0	0	0	0		X	0	0
UK(UKHEP)	0		0	0		0		0	0	0	0		0		0	0	0	0		0	X	0
UK(UKeScienceCA)	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	X

16/22 FAIL !!!

- **Since:** (condition) → (graded issue)
- then must define condition per feature → {rules}
- e.g.: (name eq 'NIL') → (graded issue)
  - thus: if (name eq 'NIL') (graded issue) == (coefficient @ class)
  - per class: (severity) == Σ(graded issues)$|^{\text{limit=1.0}}$

- Allow for classes: [minor | major | severe]
- Allow for security levels: [high | medium | low]
- **Syntax:** (condition) severity = (level, class, weight)

```
CA_key_lifetime
    if_gt ( 1825 ) severity = (low, severe, 100%)
```

- EDG can define its default ruleset
- each VO can define its own VO-specific ruleset overloadings
- each CA can define its own CA-specific ruleset overloadings

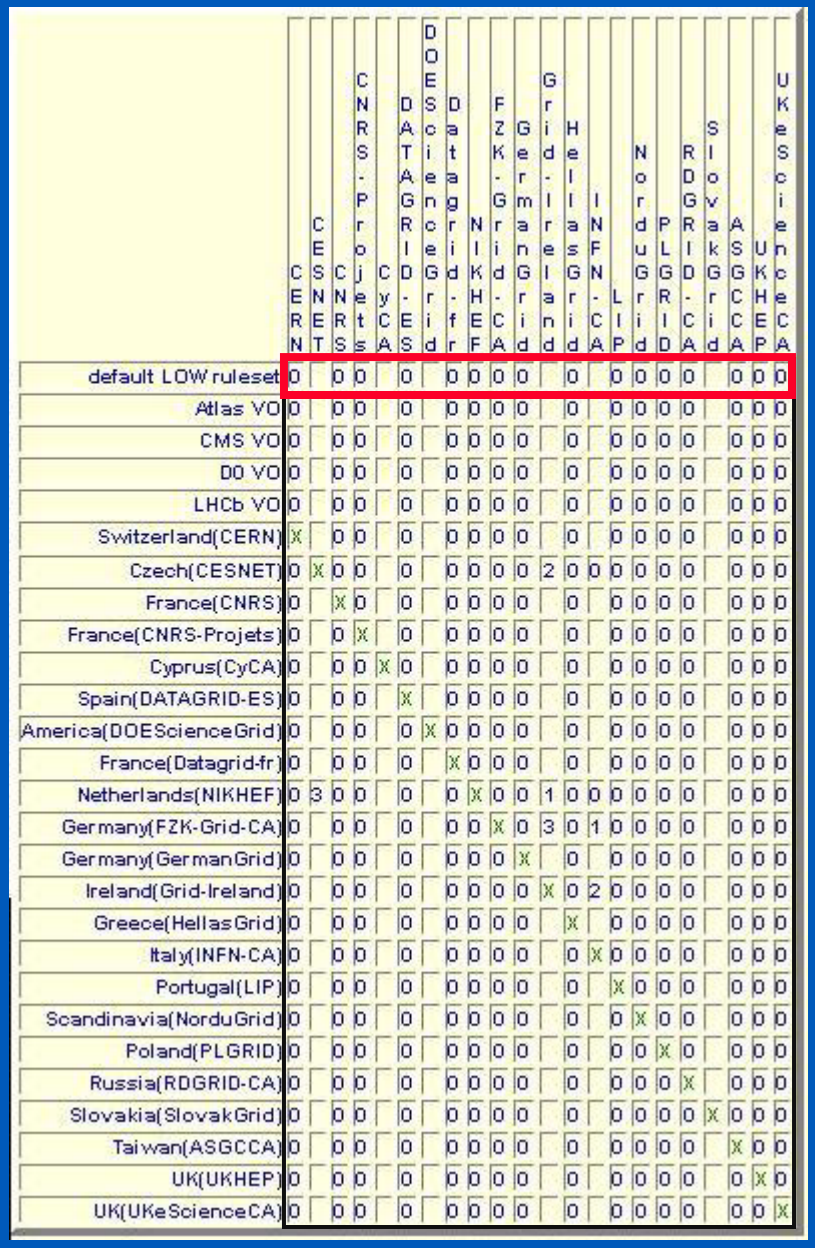
```
#
# Default Ruleset
#
inspecting_CA:
  name = "default LOW ruleset"                    # " < name > "
   if_eq ( "NIL" ) severity = (low, severe, 100%)
  alias = default-low                             # < alias >
   if_eq ( "NIL" ) severity = (low, severe, 100%)
  country =                                       # < country >
   if_eq ( "NIL" ) severity = (low, severe, 100%)
  country_ID =                                    # < US | IT | CH | IE | ... >
   if_eq ( "NIL" ) severity = (low, severe, 100%)
 CP_and_CPS:
   RFC2527_compliant =                 # < true | false >
    #if_ne ( "true" ) severity = (low, minor, 20%)
   OID_identifier =                               # < OID >
    #if_eq ( "NIL" ) severity = (low, minor, 20%)
   OID_in_cert =                       # < true | false >
    #if_ne ( "true" ) severity = (low, minor, 20%)
```

```
#
# Default Ruleset   [CONTINUED]
#
CA_web_server:
  publishes_CA_cert =                  # < true | false >
   if_ne ( "true" ) severity = (low, severe, 100%)
  publishes_CRL =                      # < true | false >
   if_ne ( "true" ) severity = (low, severe, 100%)
  publishes_CP =                       # < true | false >
   if_ne ( "true" ) severity = (low, severe, 100%)
  cert_publication_max_latency =       # < days >
   if_gt ( 7 ) severity = (low, severe, 100%)
  CRL_publication_min_freq =           # < freq in days >
   if_gt ( 23 ) severity = (low, severe, 100%)
  CRL_publication_max_latency =        # < days >
   if_gt ( 0 ) severity = (low, severe, 100%)
```

```
#
# Default Ruleset   [CONTINUED]
#
cert_issuance:
  CA_obtains_proof_of_key_possession =   # < true | false >
    #if_ne ( "true" ) severity = (low, minor, 100%)
  subject_keys_generated_by_CA =                    # < true | false >
    if_ne ( "false" ) severity = (low, severe, 100%)
  CRLs:
  lifetime =                                        # < period in days >
    if_gt ( 30 ) severity = (low, severe, 100%)
  lifetime_after_revocation =         # < period in hours >
    if_gt ( 1 ) severity = (low, severe, 100%)
  cert_signing_host:
  controlled_physical_access =       # < true | false >
    if_ne ( "true" ) severity = (low, severe, 100%)
  CA_private_keys:
  backed_up =                                # < true | false >
    #if_ne ( "true" ) severity = (low, major, 50%)
```

```
#
# Default Ruleset   [CONTINUED]
#
certs:
  CA_key_size =                          # < key size in bits >
    if_lt ( 2048 ) severity = (low, severe, 100%)
  CA_key_lifetime =                      # < duration in days >
    if_gt ( 1825 ) severity = (low, severe, 100%)
  minimum_subject_key_size =             # < key size in bits >
    if_lt ( 1024 ) severity = (low, severe, 100%)
  maximum_subject_key_lifetime =         # < duration in days >
    if_gt ( 420 ) severity = (low, severe, 100%)
```

```
#
# Default Ruleset   [CONTINUED]
#
cert_profile:
  version = X.509v3                                  # < X.509v1 | X.509v2 | X.509v3 >
    if_ne ( "X.509v3" ) severity = (low, severe, 100%)
  cert_extensions:
    SubjectKeyIdentifier =                           # < present | absent >
      #if_ne ( "present" ) severity = (low, minor, 20%)
    AuthorityKeyIdentifier =                         # < present | absent >
      #if_ne ( "present" ) severity = (low, minor, 20%)
    BasicConstraints =                               # < absent | critical | non_critical >
      if_ne ( "critical" ) severity = (low, severe, 100%)
      BasicConstraints_value =                       # < notCA | CA | yet_to_be_defined >
        if_ne ( "CA" ) severity = (low, severe, 100%)
    KeyUsage =                                       # < absent | critical | non_critical >
      if_ne ( "critical" ) severity = (low, severe, 100%)
CRL_profile:
  version =                                          # < X.509v1 >
    #if_ne ( "X.509v1" ) severity = (low, severe, 100%)
```



	CERN	CESNET	CNRS	CNRS-Projets	CyCA	DATAGRID-ES	DOEScienceGrid	Datagrid-fr	NIKHEF	FZK-Grid-CA	GermanGrid	Grid-Ireland	HellasGrid	INFN-CA	LIP	NorduGrid	PLGRID	RDIG-CA	SlovakGrid	ASGCCA	UKHEP	UKeScienceCA
default LOW ruleset	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
Atlas VO	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
CMS VO	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
D0 VO	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
LHCb VO	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
Switzerland(CERN)	X		0	0		0		0	0	0	0		0		0	0	0	0		0	0	0
Czech(CESNET)	0	X	0	0		0		0	0	0	0	2	0	0	0	0	0	0		0	0	0
France(CNRS)	0		X	0		0		0	0	0	0		0		0	0	0	0		0	0	0
France(CNRS-Projets)	0		0	X		0		0	0	0	0		0		0	0	0	0		0	0	0
Cyprus(CyCA)	0		0	0	X	0		0	0	0	0		0		0	0	0	0		0	0	0
Spain(DATAGRID-ES)	0		0	0		X		0	0	0	0		0		0	0	0	0		0	0	0
America(DOEScienceGrid)	0		0	0		0	X	0	0	0	0		0		0	0	0	0		0	0	0
France(Datagrid-fr)	0		0	0		0		X	0	0	0		0		0	0	0	0		0	0	0
Netherlands(NIKHEF)	0	3	0	0		0		0	X	0	0	1	0	0	0	0	0	0		0	0	0
Germany(FZK-Grid-CA)	0		0	0		0		0	0	X	0	3	0	1	0	0	0	0		0	0	0
Germany(GermanGrid)	0		0	0		0		0	0	0	X		0		0	0	0	0		0	0	0
Ireland(Grid-Ireland)	0		0	0		0		0	0	0	0	X	0	2	0	0	0	0		0	0	0
Greece(HellasGrid)	0		0	0		0		0	0	0	0		X		0	0	0	0		0	0	0
Italy(INFN-CA)	0		0	0		0		0	0	0	0		0	X	0	0	0	0		0	0	0
Portugal(LIP)	0		0	0		0		0	0	0	0		0		X	0	0	0		0	0	0
Scandinavia(NorduGrid)	0		0	0		0		0	0	0	0		0		0	X	0	0		0	0	0
Poland(PLGRID)	0		0	0		0		0	0	0	0		0		0	0	X	0		0	0	0
Russia(RDGRID-CA)	0		0	0		0		0	0	0	0		0		0	0	0	X		0	0	0
Slovakia(SlovakGrid)	0		0	0		0		0	0	0	0		0		0	0	0	0	X	0	0	0
Taiwan(ASGCCA)	0		0	0		0		0	0	0	0		0		0	0	0	0		X	0	0
UK(UKHEP)	0		0	0		0		0	0	0	0		0		0	0	0	0		0	X	0
UK(UKeScienceCA)	0		0	0		0		0	0	0	0		0		0	0	0	0		0	0	X

16/22 FAIL
Default Ruleset !

EXAMPLE

Autoevaluation Report:

inspecting_CA=default LOW ruleset

inspected_CA=CNRS

inspected_CA:

publishes_CA_cert=

issue with default rule: if_ne (true) severity = (low,severe,100%)

publishes_CRL=

issue with default rule: if_ne (true) severity = (low,severe,100%)

publishes_CP=

issue with default rule: if_ne (true) severity = (low,severe,100%)

IGNORE (doesn't work)

EXAMPLE

Autoevaluation Report [CONTINUED]

cert_signing_host:

controlled_physical_access= ***issue with default rule:*** if_ne (true) severity = (low,severe,100%)
CA_key_lifetime=7300 ***issue with default rule:*** if_gt (1825) severity = (low,severe,100%)
minimum_subject_key_size= ***issue with default rule:*** if_lt (1024) severity = (low,severe,100%)
_BasicConstraints=non_critical ***issue with default rule:*** if_ne (critical) severity = (low,severe,100%)
_KeyUsage=non_critical ***issue with default rule:*** if_ne (critical) severity = (low,severe,100%)

CA	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CERN		X	X													
CNRS				X	X	X	X	X	X		X		X		X	
CNRS_projet				X	X	X	X	X	X		X		X		X	
datagrid_ES				X	X	X	X	X			X		X		X	
datagrid_fr	X						X		X		X		X		X	
NIKHEF															X	
FZK																X
GermanGrid				X	X	X	X	X		X	X			X	X	
HellasGrid	X								X							
LIP														X	X	
NorduGrid				X	X	X	X	X		X	X		X		X	
PolishGrid		X							X							
Russia										X	X		X	X	X	
Taiwan				X	X	X	X	X		X	X	X	X	X	X	
UK HEP						X	X	X		X	X	X	X	X	X	
UK eScience						X	X	X			X				X	

FAILURES

CRL

Web server

key gen

cert details

empty entries

CA	1	2	3	4	5	6	7	8	
CERN		X	X						
CNRS				X	X	X	X	X	
CNRS_projet				X	X	X	X	X	
datagrid_ES				X	X	X	X	X	
datagrid_fr	X						X		
NIKHEF									✓
FZK									✓
GermanGrid				X	X	X	X	X	
HellasGrid	X								
LIP									✓
NorduGrid				X	X	X	X	X	
PolishGrid		X							
Russia									✓
Taiwan				X	X	X	X	X	
UK HEP						X	X	X	
UK eScience						X	X	X	

CRL

Web server

key gen

empty entries

(a) extracting from cert helps a lot

(b) talk to D.Chadwick