

w.c.s. - Bug #3403

Erreur lors d'un GET sur /saml/manageNameIdReturn

01 août 2013 09:57 - Benjamin Dauvergne

| | | | |
|--|--------|----------------------|--------------|
| Statut: | Fermé | Début: | 01 août 2013 |
| Priorité: | Normal | Echéance: | |
| Assigné à: | | % réalisé: | 0% |
| Catégorie: | | Temps estimé: | 0:00 heure |
| Version cible: | | Planning: | |
| Patch proposé: | | | |
| Description | | | |
| Le code de load_identity() essaie de charger un dump qui n'est pas là. | | | |
| Exception: | | | |
| <pre>type = '<type 'exceptions.TypeError'>', value = 'argument 2 must be string, not None'</pre> | | | |
| Stack trace (most recent call first): | | | |
| <pre>File "/usr/lib/python2.6/dist-packages/lasso.py", line 1845, in setIdentityFromDump 1843 @param dump: XML identity dump 1844 ''' > 1845 rc = _lasso.profile_set_identity_from_dump(self._cptr, dump) 1846 Error.raise_on_rc(rc) 1847</pre> | | | |
| locals: | | | |
| <pre>self = <lasso.NameIdManagement object at 0x3715890> dump = None</pre> | | | |
| File "/usr/lib/pymodules/python2.6/wcs/qommon/saml2.ptl", line 85, in load_identity | | | |
| <pre>83 else: 84 if session: > 85 profile.setIdentityFromDump(session.lasso_anonymous_identity_dump) 86 87 def saml2_status_summary(response):</pre> | | | |
| locals: | | | |
| <pre>profile = <lasso.NameIdManagement object at 0x3715890> session = <BasicSession at 3a01c50: None> request = <qommon.http_request.HTTPRequest instance at 0x399a8c0></pre> | | | |
| File "/usr/lib/pymodules/python2.6/wcs/qommon/saml2.ptl", line 716, in manage_name_id_return | | | |
| <pre>714 715 def manage_name_id_return(self, manage, message): > 716 load_identity(manage) 717 try: 718 manage.processResponseMsg(message)</pre> | | | |
| locals: | | | |
| <pre>message = '' manage = <lasso.NameIdManagement object at 0x3715890> self = <modules.saml2.Saml2Directory object at 0x3bfa410></pre> | | | |
| File "/usr/lib/pymodules/python2.6/wcs/qommon/saml2.ptl", line 712, in manageNameIdReturn | | | |
| <pre>710 711 message = get_request().get_query() > 712 return self.manage_name_id_return(manage, message) 713 714</pre> | | | |
| locals: | | | |
| <pre>message = '' manage = <lasso.NameIdManagement object at 0x3715890></pre> | | | |

```

self = <modules.saml2.Saml2Directory object at 0x3bfa410>

File "/usr/lib/pymodules/python2.6/quixote/directory.py", line 67, in _q_traverse
65         raise TraversalError
66         elif hasattr(obj, '__call__'):
> 67             return obj()
68         else:
69             return obj

locals:
component = 'manageNameIdReturn'
self = <modules.saml2.Saml2Directory object at 0x3bfa410>
obj = <bound method Saml2Directory.manageNameIdReturn of <modules.saml2.Saml2Directory object
at 0x3bfa410>>
name = 'manageNameIdReturn'
path = []

File "/usr/lib/pymodules/python2.6/wcs/qommon/saml2.ptl", line 111, in _q_traverse
109             '%s unavailable - lasso is not installed' % rel_path)
110         raise errors.TraversalError()
> 111         return Directory._q_traverse(self, path)
112
113     def log_profile_error(self, profile, error, what):

locals:
path = ['manageNameIdReturn']
self = <modules.saml2.Saml2Directory object at 0x3bfa410>

File "/usr/lib/pymodules/python2.6/quixote/directory.py", line 63, in _q_traverse
61         if path:
62             if hasattr(obj, '_q_traverse'):
> 63                 return obj._q_traverse(path)
64             else:
65                 raise TraversalError

locals:
component = 'saml'
self = <modules.root.AlternateRootDirectory object at 0x259dd90>
obj = <modules.saml2.Saml2Directory object at 0x3bfa410>
name = 'saml'
path = ['manageNameIdReturn']

File "/usr/lib/pymodules/python2.6/extra-wcs-au-quotidien/modules/root.ptl", line 787, in _q_tra
verse
785             return redirect('%s%s/%s' % (base_url, f.category.url_name, uri_rest))
786
> 787         raise e
788
789

locals:
path = ['saml', 'manageNameIdReturn']
self = <modules.root.AlternateRootDirectory object at 0x259dd90>
session = <BasicSession at 3a01c50: None>
response = <qommon.http_response.HTTPResponse instance at 0x399a4d0>

File "/usr/lib/pymodules/python2.6/quixote/publish.py", line 253, in try_publish
251         permanent=True)
252         components = path[1:].split('/')
> 253         output = self.root_directory._q_traverse(components)
254         # The callable ran OK, commit any changes to the session
255         self.finish_successful_request()

locals:
path = '/saml/manageNameIdReturn'
self = <publisher.WcsPublisher instance at 0x25d0cf8>
request = <qommon.http_request.HTTPRequest instance at 0x399a8c0>

```

```

components = ['saml', 'manageNameIdReturn']

File "/usr/lib/pymodules/python2.6/wcs/qommon/publisher.py", line 492, in try_publish
490     for extra_source in self.extra_sources:
491         self.substitutions.feed(extra_source(self, request))
> 492     return Publisher.try_publish(self, request)
493
494     def get_site_language(self):

locals:
    pystatsd = <module 'qommon.vendor.pystatsd' from '/usr/lib/pymodules/python2.6/wcs/qommon/vendor/pystatsd.py'>
    request = <qommon.http_request.HTTPRequest instance at 0x399a8c0>
    script_name = ''
    extra_source = <class 'modules.abelium_domino_vars.DominoVariables'>
    canonical_hostname = 'cud.au-quotidien.com'
    self = <publisher.WcsPublisher instance at 0x25d0cf8>

File "/usr/lib/pymodules/python2.6/wcs/publisher.py", line 140, in try_publish
138     if request.get_header('X_WCS_IFFRAME_MODE', '') in ('true', 'yes'):
139         request.response.iframe_mode = True
> 140     return QommonPublisher.try_publish(self, request)
141
142     set_publisher_class(WcsPublisher)

locals:
    self = <publisher.WcsPublisher instance at 0x25d0cf8>
    request = <qommon.http_request.HTTPRequest instance at 0x399a8c0>

File "/usr/lib/pymodules/python2.6/quixote/publish.py", line 281, in process_request
279     except:
280         # Some other exception, generate error messages to the logs, etc.
> 281         output = self.finish_failed_request()
282         output = self.filter_output(request, output)
283         self.logger.log_request(request, start_time)

locals:
    start_time = 1375310450.3601241
    request = <qommon.http_request.HTTPRequest instance at 0x399a8c0>
    self = <publisher.WcsPublisher instance at 0x25d0cf8>

File "/usr/lib/pymodules/python2.6/wcs/qommon/publisher.py", line 784, in process
782     def process(self, stdin, env):
783         request = HTTPRequest(stdin, env)
> 784         self.response = self.process_request(request)
785         return self.response
786

locals:
    self = <publisher.WcsPublisher instance at 0x25d0cf8>
    stdin = <open file '<socket>', mode 'r' at 0x3cf2ed0>
    request = <qommon.http_request.HTTPRequest instance at 0x399a8c0>
    env = {'HTTP_X_AU_QUOTIDIEN_LOCKED': 'False', 'SERVER_SOFTWARE': 'Apache/2.2.16 (Debian)', 'SCRIPT_NAME': '', 'REQUEST_METHOD': 'GET', 'PATH_INFO': '/saml/manageNameIdReturn', 'HTTP_FROM': 'googlebot (at) googlebot.com', 'SERVER_PROTOCOL': 'HTTP/1.1', 'QUERY_STRING': '', 'CONTENT_LENGTH': '0', 'HTTP_USER_AGENT': 'Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)', 'HTTP_CONNECTION': 'Keep-alive', 'SERVER_NAME': 'cud.au-quotidien.com', 'REMOTE_ADDR': '66.249.78.47', 'SERVER_PORT': '80', 'SERVER_ADDR': '81.93.255.234', 'DOCUMENT_ROOT': '/usr/share/wcs/web/', 'SERVER_ADMIN': 'webmaster@au-quotidien.com', 'SCRIPT_URI': 'http://cud.au-quotidien.com/saml/manageNameIdReturn', 'HTTP_HOST': 'cud.au-quotidien.com', 'SCRIPT_URL': '/saml/manageNameIdReturn', 'REQUEST_URI': '/saml/manageNameIdReturn', 'HTTP_ACCEPT': '*/*', 'REMOTE_PORT': '62640', 'SCGI': '1', 'HTTP_ACCEPT_ENCODING': 'gzip,deflate'}

File "/usr/lib/pymodules/python2.6/quixote/server/scgi_server.py", line 27, in handle_connection
25         env['PATH_INFO'] = path[len(prefix):] + env.get('PATH_INFO', '')
26
> 27         response = self.publisher.process(input, env)

```

```

28         try:
29             response.write(output)

locals:
  output = <open file '<socket>', mode 'w' at 0x3cf2c90>
  self = <qommon.scgi_server.QommonHandler instance at 0x2d73ab8>
  conn = <socket object, fd=6, family=2, type=1, protocol=0>
  env = {'HTTP_X_AU_QUOTIDIEN_LOCKED': 'False', 'SERVER_SOFTWARE': 'Apache/2.2.16 (Debian)', 'SCRIPT_NAME': '', 'REQUEST_METHOD': 'GET', 'PATH_INFO': '/saml/manageNameIdReturn', 'HTTP_FROM': 'googlebot (at)googlebot.com', 'SERVER_PROTOCOL': 'HTTP/1.1', 'QUERY_STRING': '', 'CONTENT_LENGTH': '0', 'HTTP_USER_AGENT': 'Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)', 'HTTP_CONNECTION': 'Keep-alive', 'SERVER_NAME': 'cud.au-quotidien.com', 'REMOTE_ADDR': '66.249.78.47', 'SERVER_PORT': '80', 'SERVER_ADDR': '81.93.255.234', 'DOCUMENT_ROOT': '/usr/share/wcs/web/', 'SERVER_ADMIN': 'webmaster@au-quotidien.com', 'SCRIPT_URI': 'http://cud.au-quotidien.com/saml/manageNameIdReturn', 'HTTP_HOST': 'cud.au-quotidien.com', 'SCRIPT_URL': '/saml/manageNameIdReturn', 'REQUEST_URI': '/saml/manageNameIdReturn', 'HTTP_ACCEPT': '*/*', 'REMOTE_PORT': '62640', 'SCGI': '1', 'HTTP_ACCEPT_ENCODING': 'gzip,deflate'}
  input = <open file '<socket>', mode 'r' at 0x3cf2ed0>

File "/usr/lib/pymodules/python2.6/wcs/qommon/scgi_server.py", line 39, in handle_connection
37     def handle_connection(self, conn):
38         self.number_of_connection_handled = self.number_of_connection_handled + 1
> 39         quixote.server.scgi_server.QuixoteHandler.handle_connection(self, conn)
40         # input, output and conn are closed, long running jobs could be done
41         # here.

locals:
  self = <qommon.scgi_server.QommonHandler instance at 0x2d73ab8>
  conn = <socket object, fd=6, family=2, type=1, protocol=0>

File "/usr/lib/pymodules/python2.6/scgi/scgi_server.py", line 71, in serve
69         conn.setblocking(1)
70         os.close(fd)
> 71         self.handle_connection(conn)
72
73

locals:
  self = <qommon.scgi_server.QommonHandler instance at 0x2d73ab8>
  fd = 5
  conn = <socket object, fd=6, family=2, type=1, protocol=0>

File "/usr/lib/pymodules/python2.6/scgi/scgi_server.py", line 194, in spawn_child
192             # the connection in the child
193             os.close(child_fd)
> 194             self.handler_class(parent_fd).serve()
195             sys.exit(0)
196         else:

locals:
  self = <qommon.scgi_server.SCGIServer instance at 0x2d73878>
  pid = 0
  flags = 2
  child_fd = 4
  parent_fd = 3
  conn = None

File "/usr/lib/pymodules/python2.6/scgi/scgi_server.py", line 172, in __init__
170         self.max_children = max_children
171         self.children = {} # { pid : fd }
> 172         self.spawn_child()
173         self.restart = 0
174

locals:
  max_children = 5
  host = 'localhost'

```

```

handler_class = <function create_handler at 0x29b7c08>
port = 3001
self = <qommon.scgi_server.SCGIServer instance at 0x2d73878>

File "/usr/lib/pymodules/python2.6/wcs/qommon/scgi_server.py", line 77, in run
75
76     s = SCGIServer(create_handler, host=host, port=port,
> 77         max_children=max_children)
78     s.serve()
79

locals:
create_handler = <function create_handler at 0x29b7c08>
handler_connection_limit = -1
max_children = 5
create_publisher = <bound method classobj.create_publisher of <class publisher.WcsPublisher a
t 0x340ad10>>
script_name = None
host = 'localhost'
spawn_cron = True
port = 3001

File "/usr/lib/pymodules/python2.6/wcs/ctl/start.py", line 67, in execute
65
66     try:
> 67         run_function(publisher.WcsPublisher.create_publisher, **run_kwargs)
68     except socket.error, e:
69         if e[0] == 98:

locals:
publisher = <module 'publisher' from '/usr/lib/pymodules/python2.6/wcs/publisher.py'>
self = <wcs.ctl.start.CmdStart instance at 0x2d05b90>
args = []
sub_options = <Values at 0x2d735f0: {'http': False, 'silent': False, 'app_dir': None, 'extra'
: ['/usr/lib/pymodules/python2.6/extra-wcs-au-quotidien/'], 'max_children': None, 'handler_connect
ion_limit': None, 'script_name': None, 'data_dir': None, 'port': 3001}>
run_function = <function run at 0x2d72d70>
base_options = <Values at 0x2d01290: {'configfile': '/etc/wcs/wcs-au-quotidien.cfg'}>
run_kwargs = {'port': 3001, 'spawn_cron': True}

File "/usr/lib/pymodules/python2.6/wcs/qommon/ctl.py", line 65, in run
63     if sub_options.data_dir:
64         self.config.set("main", "data_dir", sub_options.data_dir)
> 65     return self.execute(base_options, sub_options, args)
66
67     def parse_args(self, args):

locals:
sub_options = <Values at 0x2d735f0: {'http': False, 'silent': False, 'app_dir': None, 'extra'
: ['/usr/lib/pymodules/python2.6/extra-wcs-au-quotidien/'], 'max_children': None, 'handler_connect
ion_limit': None, 'script_name': None, 'data_dir': None, 'port': 3001}>
args = []
base_options = <Values at 0x2d01290: {'configfile': '/etc/wcs/wcs-au-quotidien.cfg'}>
self = <wcs.ctl.start.CmdStart instance at 0x2d05b90>

File "/usr/lib/pymodules/python2.6/wcs/qommon/ctl.py", line 140, in run
138     command_class = qommon._commands[command]
139     cmd = command_class()
> 140     return cmd.run(args, options)
141

locals:
cmd = <wcs.ctl.start.CmdStart instance at 0x2d05b90>
args = ['--extra', '/usr/lib/pymodules/python2.6/extra-wcs-au-quotidien/']
self = <wcs.qommon.ctl.Ctl instance at 0x29a7a28>
command_class = <class wcs.ctl.start.CmdStart at 0x2d0a470>
command = 'start'

```

```
cmd_prefix = 'wcs.ctl'
options = <Values at 0x2d01290: {'configfile': '/etc/wcs/wcs-au-quotidien.cfg'}>
```

```
File "/usr/sbin/wcsctl", line 8, in <module>
```

```
6
7 ctl = wcs.gommon.ctl.Ctl(cmd_prefixes=['wcs.ctl'])
> 8 ctl.run(sys.argv[1:])
9
```

```
locals:
```

```
wcs = <module 'wcs' from '/usr/lib/pymodules/python2.6/wcs/__init__.pyc'>
__builtins__ = <module '__builtin__' (built-in)>
__file__ = '/usr/sbin/wcsctl'
__package__ = None
sys = <module 'sys' (built-in)>
ctl = <wcs.gommon.ctl.Ctl instance at 0x29a7a28>
__name__ = '__main__'
__doc__ = None
```

Form:

Cookies:

Environment:

```
CONTENT_LENGTH '0'
DOCUMENT_ROOT '/usr/share/wcs/web/'
HTTP_ACCEPT '*/*'
HTTP_ACCEPT_ENCODING 'gzip, deflate'
HTTP_CONNECTION 'Keep-alive'
HTTP_FROM 'googlebot (at) googlebot.com'
HTTP_HOST 'cud.au-quotidien.com'
HTTP_USER_AGENT 'Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html) '
HTTP_X_AU_QUOTIDIEN_LOCKED 'False'
PATH_INFO '/saml/manageNameIdReturn'
QUERY_STRING ''
REMOTE_ADDR '66.249.78.47'
REMOTE_PORT '62640'
REQUEST_METHOD 'GET'
REQUEST_URI '/saml/manageNameIdReturn'
SCGI '1'
SCRIPT_NAME ''
SCRIPT_URI 'http://cud.au-quotidien.com/saml/manageNameIdReturn'
SCRIPT_URL '/saml/manageNameIdReturn'
SERVER_ADDR '81.93.255.234'
SERVER_ADMIN 'webmaster@au-quotidien.com'
SERVER_NAME 'cud.au-quotidien.com'
SERVER_PORT '80'
SERVER_PROTOCOL 'HTTP/1.1'
SERVER_SOFTWARE 'Apache/2.2.16 (Debian)'
```

Historique

#1 - 07 août 2013 10:01 - Frédéric Péters

- Statut changé de Nouveau à Résolu (à déployer)

ok, poussé.

#2 - 20 novembre 2013 12:09 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

| | | | |
|---|--------|--------------|--------------------|
| 0001-saml2-only-load-anonymous-identity-dump-if-present.patch | 1,1 ko | 01 août 2013 | Benjamin Dauvergne |
|---|--------|--------------|--------------------|