ADQL versus Firewalls

Tom Donaldson

IVOA Interop – Northern Fall 2022
Normal TAP Queries – Gaia Search in Topcat
Normal TAP Queries – RegTAP via PyVO

```python
allwise_image_services = vo.regsearch(servicetype='image', keywords=['allwise'])
allwise_image_services.to_table()[['ivoid', 'short_name', 'res_title']]
```

[4]: $\text{Table length}=1$

<table>
<thead>
<tr>
<th>ivoid</th>
<th>short_name</th>
<th>res_title</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>object</td>
<td>object</td>
</tr>
<tr>
<td>ivo://irsa.ipac/wise/images/allwise/l3a</td>
<td>AllWISE L3a</td>
<td>AllWISE Atlas (L3a) Coadd Images</td>
</tr>
</tbody>
</table>
But then...

You make a small change, like:

- add an ORDER BY
- come into the office for the first time
- just try the same thing a month later
What Happened?

Is the service down?

Is my internet access having problems?

But if I remove ORDER BY, it works.

Some queries work, some don’t. Hmm
Wait! We've seen this before. Sort of...

A few years ago
- We noticed that remote clients for MAST TAP services had similar errors
- But the same queries on-site worked fine.

After working with network security team
- Web Application Firewall (WAF) was blocking those queries because they looked like SQL Injection
- Solved: Add exceptions to WAF rules to not check for SQL injection for MAST TAP endpoints

But my recent errors are different
- I’m on-site at STScI
- Some failed queries are to external services like the GAVO RegTAP service.
- Fails on internal network, guest network, and eduroam
  - Need to go home or use my iPhone
What Changed?

Could a firewall be blocking outbound traffic? Traffic within STScI?

• Yes

Now more partitions and firewalls within the STScI network

• Not just the WAF. Also have perimeter firewall and user firewalls.
  – All have more advanced threat detection built in
• Traffic is scanned for SQL injection in all directions

Aside about SQL Injection...
What is SQL Injection and Why Do We Care?

What is SQL Injection?

• User input anticipates how the SQL query will be constructed; breaks out to do something else.

```javascript
var user_input = Request.getParam("obs_id"); // ["myobsid"]
var query = "SELECT * FROM ObsCore WHERE obs_id = " + user_input + ";";
var results = Sql.execute(query);
```

• User inputs like these could break out of intended behavior:
  - 1 or 1=1 ➔ All rows returned
  - 1; DROP TABLE ObsCore ➔ Drops the ObsCore table

Why do we care?

• Worst case successful attack is quite damaging and embarrassing
• Injection is #3 on the OWASP* top 10 list of web app security risks
• STScI is probed pretty much continuously, mostly by bots

(* OWASP is the Open Web Application Security Project, "an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted."
SQL Injection Mitigation

→ Use "prepared statements" instead of building a string for SQL to execute.

```java
String[] params = ADQLParser.getParams(); // ["myobsid"]
String query = ADQLParser.getQueryTemplate(); // "SELECT * FROM ObsCore WHERE obs_id = ? ";

PreparedStatement pstmt = connection.prepareStatement( query );
for (int i=1; i<=params.length; i++) {
    pstmt.setValue( 1, params[i] );
    ResultSet results = pstmt.executeQuery( );
}
```

• Challenging with ADQL’s dynamic queries.
  - E.g., not possible to use prepared statement with ORDER BY value.
• *Very* tempting to pass large sections of user input directly to DB.

See the OWASP Cheat Sheet
SQL Injection Mitigation (2)

→ Don’t give the web service write access to the DB.
  • Helps for attacks that try to modify the DB.
  • Not always practical or possible

→ Sanitize user input
  • Very difficult to catch everything, especially with ADQL that allows most SQL syntax.

→ Use firewalls to block requests that look like SQL injection.
For inbound requests, added firewall exceptions for the MAST TAP endpoints.

But there are > 120 registered TAP endpoints

- Can we add exceptions for all of them?
  - Even so, sometimes want to access unregistered services
- How many is too many?
  - How would we maintain the exceptions when registrations change?
  - Could too many exceptions affect firewall performance?
- Regex can be used, but what patterns to search for?
  - “LANG=ADQL” is required, so can look for it in GET params or POST data
  - “/sync” or “/async” must be at the end of the URL (prior to GET params)
Consider changes to TAP standard?

Assume STScI adds firewall exceptions allowing inbound and outbound TAP requests.

• Great for STScI TAP services!
• Great for TAP users based at STScI!

But what if universities and research institutions become more security conscious?

• New firewalls could cause astronomers’ TAP queries to fail mysteriously.
• Situation may just get worse as more automation is adopted and inspection becomes more sophisticated.

Is there anything the IVOA could or should do?

• Use https for TAP services?
  – Temporary measure only. STScI doesn’t unpack https for inspection now, but expects to soon.
• Support (or require) some kind of encoding of the ADQL?
  – Actually a tactic used by malicious actors to bypass some detection
• New syntax?
  – That’s a big change, but would move us further away from SQL injection, real and imagined.