Exploiting ServiceNow@Tufts:

how thousands of current and past student, staff, and faculty data was left unsecured

by: yoji watanabe

What are we talking about?

ServiceNow®

- Cloud computing company
- Services Tufts

TechConnect

- Tufts' implementation of the ServiceNow system
- Allows users to open "incidents" to be reviewed by the Technology staff
- Each Tufts affiliate has a user profile with name, UTLN, email address, physical address, and phone number
- Example: Forms from the Research Cluster access page and special software requests are handled by TechConnect
- First vulnerabilities found on March 15, 2018

What Happened?

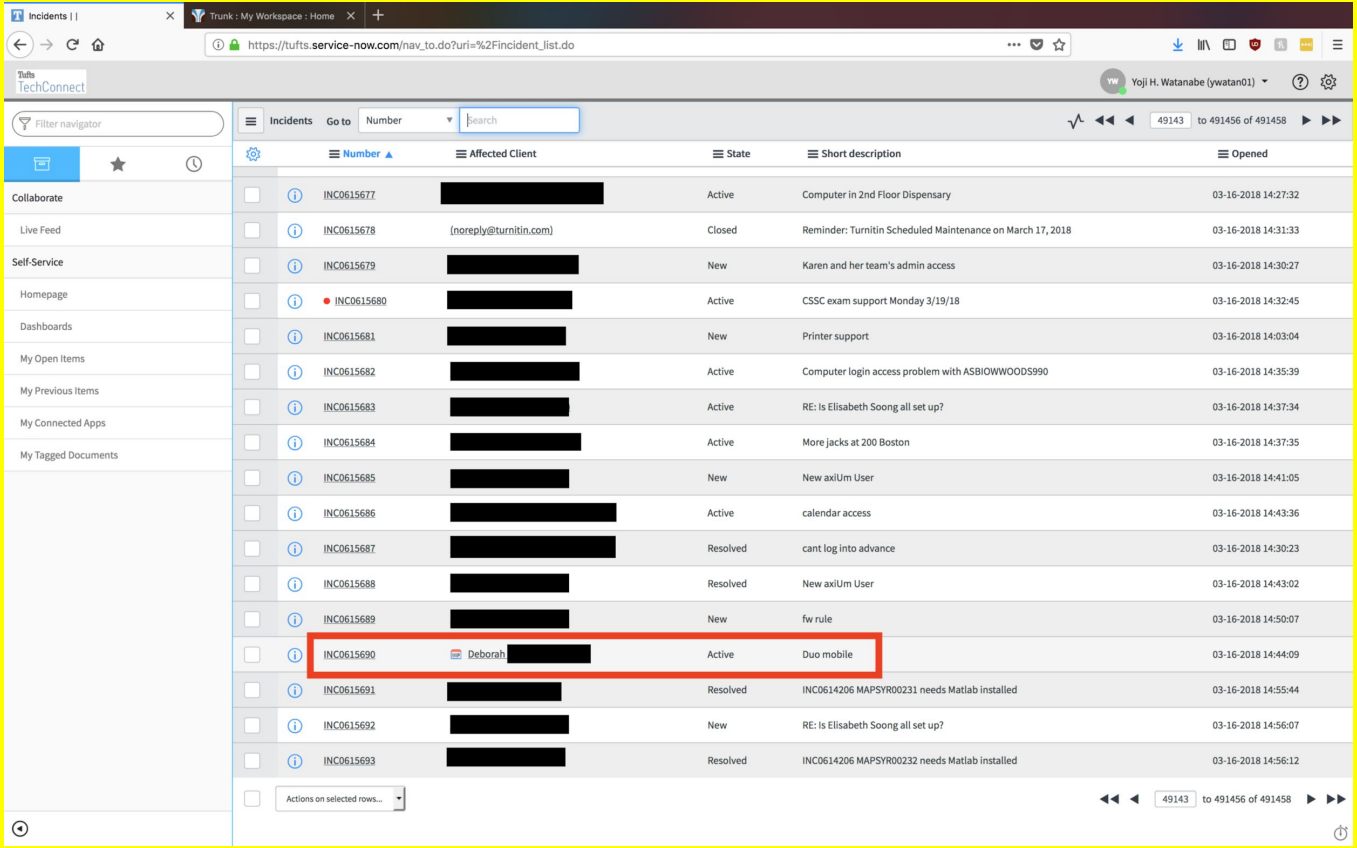
>491,458

instances of user info was exposed

>26,820 users

had their information exposed

Yeah, but you couldn't do much, right?



Retrieving the list of incidents

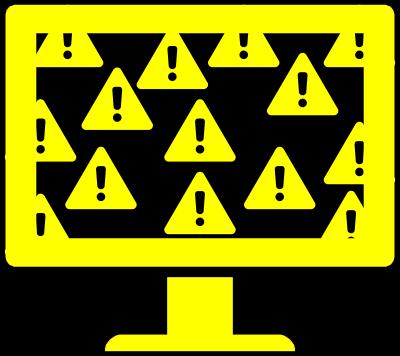
Source code of "my open items" tab


```
<table id="task_table" class="data_list_table list_table table table-hover list_header_search_disabled" glide_list_has_hierarchy="false" total_rows="1"
glide_list_edit_ref_qual_tag="" glide_list_has_actions="true" list_name="task" parsed_choice_query="<?xml version="1.0" encoding="UTF-8"?><xml table="incident">…e"
newquery="false" operator="=" or="false" value=""/></xml>" data-list_id="task" glide_list_has_header="true" can_hide_nav="false" glide_list_can_delete="false"
glide_list_edit_id="task" glide_list_edit_type="disabled" query="active=true^opened_by=javascript:gs.getUserID()^ORref_incide…^stateNOT IN3,4,7,6^numberNOT
LIKETASK^numberNOT LIKECALL^ERO" filter-enabled="true" glide_list_omit_links="false" sort="task.number" glide_list_edit_insert_row="false"
glide_list_query="active=true^opened_by=javascript%3Ags.getUserID()^ORref_inci…OES NOT CONTAINTASK^numberDOES NOT CONTAINCALL^ORDERBYnumber"
glide_list_can_create="true" rows_per_page="20" grand_total_rows="1" maintain_order="false" parsed_query="<?xml version="1.0" encoding="UTF-8"?><xml table="incident">…
e" newquery="false" operator="=" or="false" value=""/></xml>" first_row="1" last_row="1" style="width: 100%; margin-bottom: 0px; outline: medium none currentcolor;"
glide_table="task" sort_dir="ASC" tabindex="0">
```


1. Incidents are represented in a XML table
2. There are searchable tags associated with incidents
3. My Open Items is made by making a query to an incident database for incidents opened by the current user

Exploit overview

- Use information disclosed to figure out how to make database queries
- Use code injection techniques to make queries to database
- Make use of lack of access controls to access all entries



NOW WHAT?

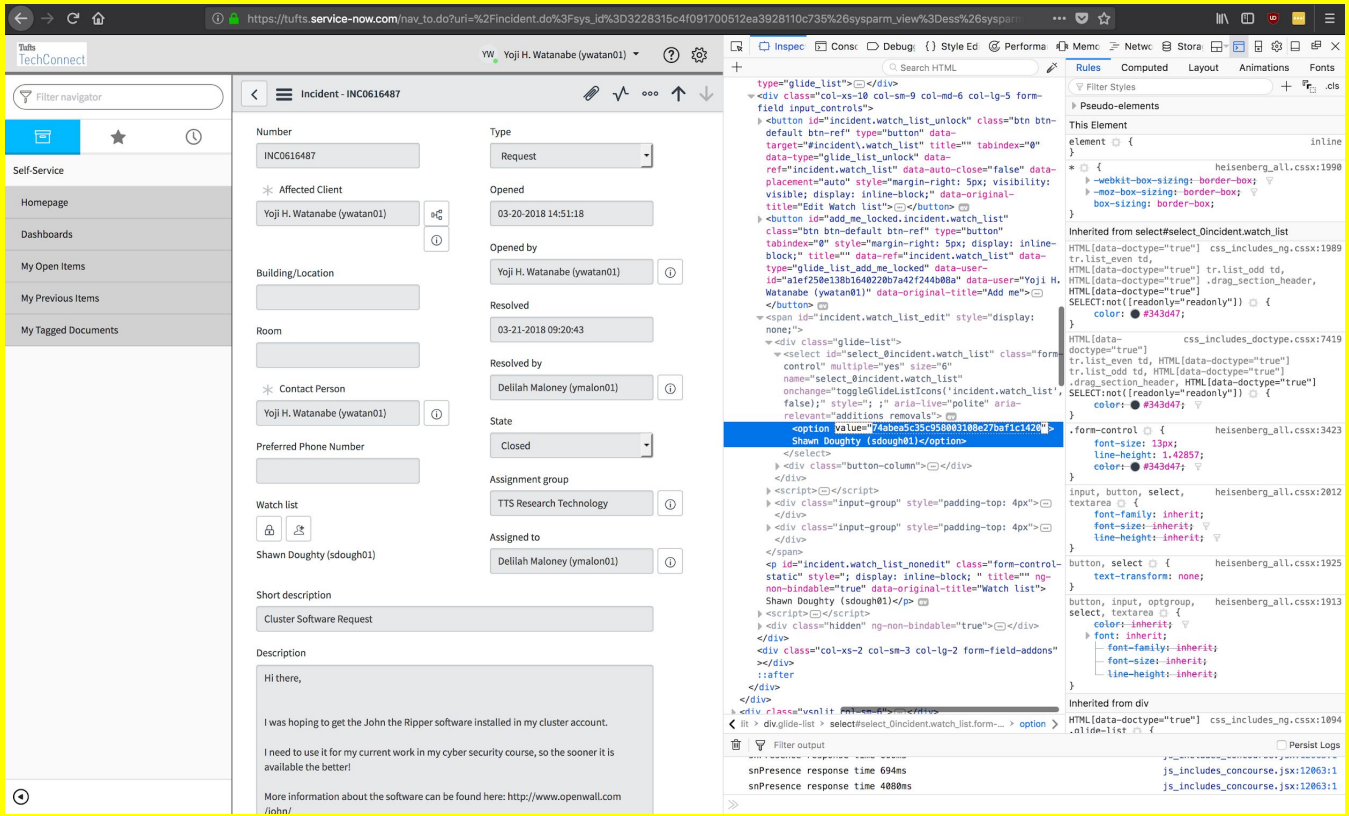
Surely, bringing these things to Tufts' attention would get them fixed

When ServiceNow® 'fixed' the issues

- No more SQLi
  - No access to user list
  - No access to incident list
- Still open to XPath Injection
- Still has access control vulnerabilities
- What kind of data do we still have access to?

- How can I get people's information in order to find their profile page?

ServiceNow® isn't safe: Round 2



Reconstructing Profile URLs

Profile URL:
https://tufts.service-now.com/nav_to.do?uri=%2Fsys_user.do%3Fsys_id%3D8eabe69c35c958003108e27baf1c144d%26sysparm_view%3Dess

Deconstructing URL						
USER	nav_to.do?	uri=%2Fsys_user.do	%3Fsys_id	%3D8eabe69c35c958003108e27baf1c144d	%26sysparm_view	%3Dess
URL decoded	nav_to.do?	uri=/sys_user.do	?sys_id	=8eabe69c35c958003108e27baf1c144d	&sysparm_view	=ess
Notes				[USER ID] !!!! Very important		

Can we then reconstruct a URL to gain access to a profile?

https://tufts.service-now.com/nav_to.do?uri=%2Fsys_user.do%3Fsys_id%3D8eabe69c35c958003108e27baf1c144d%26sysparm_view%3Dess

Tufts TechConnect

YW Yoji H. Watanabe (ywatan01)

All applications

Self-Service

- Homepage
- Dashboards
- My Open Items
- My Previous Items
- My Tagged Documents

User - [redacted]

First name	L[redacted]	Email	[redacted]
Last name	[redacted]	Notification	Email
Business phone	[redacted]	Date format	
Mobile phone		Time zone	
Title	Dir [redacted]		

Delegates | Service Groups

Delegates New Go to Starts Search

User = [redacted]

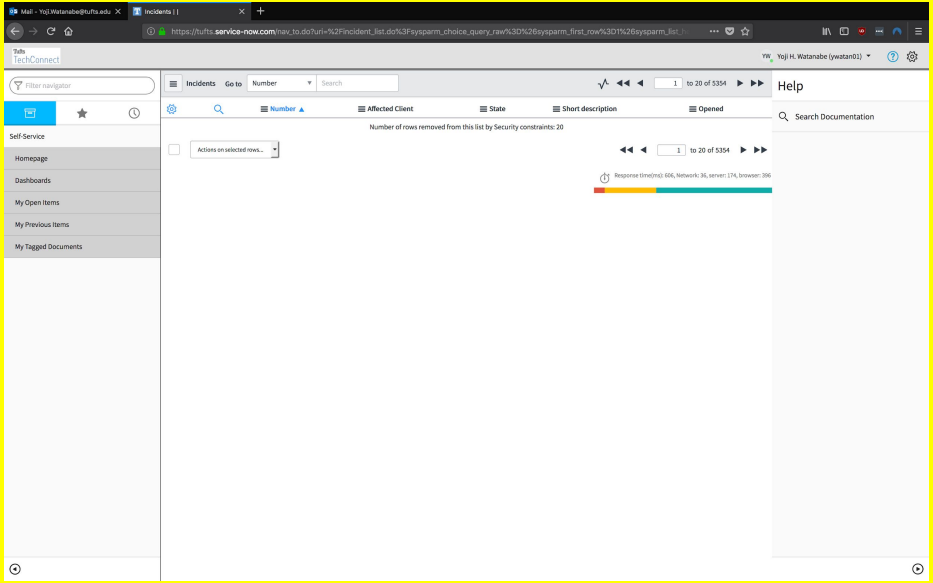
Starts	Ends	Delegate	Approvals	Assignments	CC notifications	Meeting invitations

No records to display

Response time(ms): 922, Network: 12, server: 322, browser: 588

Where are they now?

- Updated access control
  - No longer able to make unverified queries to incident or user database
  - No disclosure of critical information
- Sanitized user input
  - No longer able to perform code injection attacks



"Number of rows removed by this list by Security constraints: 20"

Reviewing vulnerabilities

SQLi

(CWE-89: Improper Neutralization of Special Elements used in an SQL Command)

- Access to incident list

XML Injection

(CWE-91: XML Injection)

- Access to incident and user list
- Search through user and incident list
- Direct to user profiles

Information Disclosure

(CWE-200: Information Exposure)

- Disclosed information to reconstruct profile URLs
- Disclosed information to access incident list
- Disclosed information about database

Access Control

(CWE-284: Improper Access Control)

- Access to user information
- Access to other incidents

Lessons learned

- 70% of the process was communicating
- It is shocking how vulnerable enterprise software can be, how inefficient they can be at fixing it
  - Initial report outlined four vulnerabilities
  - One was fixed
  - Over thirty days to fix all vulnerabilities
- It's scary

Questions?