Practical security and data access guidelines for the Canvas API @ UBC

Version 1.1 (17 October 2018)
Written by:
  Craig Thompson, Research Analyst, UBC Learning Analytics Project
Reviewed by:
  Derek White, Senior Manager, Learning, Applications and Analytics
  Tammy Yasrobi, Senior Manager, Learning Applications

Purpose

This document is intended to provide general information to the UBC Canvas API Community (CAPICO) regarding security best practices and UBC data access practices with respect to the Canvas API. This document does not represent official UBC policy, and does not supersede any existing or future UBC policy, or guidance, including but not limited to:

- Policy #104: Acceptable Use and Security of UBC Electronic Information and Systems
- UBC Information Security Standards
- UBC Canvas Terms of Use (available from within Canvas)

If any part of this document is unclear, please submit questions to lt.hub@ubc.ca

FIPPA

UBC is subject to the Freedom of Information and Protection of Privacy Act (FIPPA). All UBC employees are responsible for ensuring FIPPA compliance. If you have any doubts about the FIPPA compliance of your work, contact Paul Hancock (Legal Counsel, Information and Privacy): paul.hancock@ubc.ca
  - University Counsel Access and Privacy
Non-employee members of the UBC community are not subject to FIPPA, however when working with personal information it is considered best practice to maintain the privacy principles contained therein.

Canvas API Tokens

Don’t share your token. Your Canvas token is essentially equal to your username and password. Anything you can do in Canvas can also be done by someone else if they have your token. Likewise, Don’t ask others to share their token with you.
  - UBC Information Security Standard #02
Don’t create a web app that requires a user to manually enter their token. You may have crafted a perfectly benign app, but doing so sets up a standard for other users to expect that this is a “normal” paradigm for interacting with the Canvas API or using applications built on top of the Canvas API. Supporting this will make phishing attacks much more successful. This approach is also a violation of Canvas’ terms of service.

If you have shared your token (intentionally or accidentally): delete the token from your Canvas account. This will invalidate the token, and it will no longer be useable by anyone. If you think your token (or another person’s token) is being misused, report it immediately to security@ubc.ca

- UBC Information Security Standard #04

Several examples of recommended approaches to handling the user’s token are here.

- If you would like a more “seamless” user experience, and don’t want the user to have to manually generate an API token and download/run your scripts, you should use Oauth2 and/or LTI. If you would like assistance with setting up either of these options, please submit a request to lt.hub@ubc.ca
  - An example using oauth2 is under development, and will be available at the above repo soon. Oauth2 authentication requires a developer key. You can request one from the LTHub.
  - Likewise, an example using LTI will be available soon as well. LTI integration with Canvas at UBC requires a review of the tool, and some support from our Canvas Admins to configure the tool. Please contact lt.hub@ubc.ca for assistance.

Course access practice

Data access policies at UBC are presently under review and subject to change. The following should be interpreted as interim guidance until a more formal statement can be made.

Some Canvas users have greater technical access to information than policy allows for. For example, Canvas administrators have full ability to access and modify all content within Canvas. This level of access is needed to effectively oversee operations of Canvas. Likewise, ISS and LTR roles grant significant technical access to Canvas.

At present, guidance indicates that course instructors are the owners of their courses. Any access to a course (via the API or the web application) should be done either by:

- Someone registered in the course for an academic purpose (such as a student, TA, or instructor)
- Someone with express consent from the instructor to access the course for a specific purpose.

Permission from any single course instructor (someone who has the “Teacher” role for the course in Canvas) is sufficient in cases where a course has multiple instructors.
With permission of the instructor, an API user is free to access a course within the *technical* limits of their Canvas role for the purposes approved by the instructor.

There may be cases where it is not possible, or practical, to seek explicit consent from the instructor for every course that you need to access. For example, if you need to create a report for your Faculty on the number of courses using various features of Canvas. In such cases, an application must be made to request permission to do so. This application process is being created by the Learning Data Committee. The process is currently in a draft state and more information will be available soon.

### API Rate limiting

Instructure is *supposed* to limit the rate at which API requests can be made. This is intended to prevent one user from abusing the system and effectively blocking others from accessing Canvas. More details here: [https://canvas.instructure.com/doc/api/file.throttling.html](https://canvas.instructure.com/doc/api/file.throttling.html) and [https://community.canvaslms.com/docs/DOC-8381-api-rate-limiting](https://community.canvaslms.com/docs/DOC-8381-api-rate-limiting)

However, there have been reports at UBC that this system is not 100% effective and that some users are able to impact the quality of service provided to others.

In light of this:

- **Whenever possible, scripts should be developed and tested against the “test” Canvas installation: ubc.test.instructure.com.** This installation is for testing only, and is refreshed to mirror the production installation every three weeks.
  - Note: students do not have access to this test environment, so students will need to use the production Canvas environment: canvas.ubc.ca
  - If you are a student, you are encouraged to be in touch with other members of CAPICO, your faculty support unit (if there is one in your faculty), or the LTHub for support in designing, developing, and testing your code.

- Exercise caution in the API requests you run.
- Limit scope whenever possible.
- Don’t run many requests in parallel without good reason.
- If your account is being used to generate an excessive number of requests, your access may be temporarily revoked so as not to impact others at UBC.
- **Consider having your code reviewed by the LTHub, especially if you are new to working with the Canvas API or need to run a particularly expensive (large) set of API requests.** Submit a script (or link to a repository) to lt.hub@ubc.ca and ask for a Canvas API code review. The LTHub can help with coding standards and efficiencies as well identifying any red flags from a policy perspective.