1
00:00:05,520 --> 00:00:08,960
Let me know when you can see my screen

2
00:00:06,960 --> 00:00:10,880
Michaela

3
00:00:08,960 --> 00:00:14,719
it's visible

4
00:00:10,880 --> 00:00:14,719
all right that's all yours

5
00:00:15,200 --> 00:00:20,080
so um as Michaela mentioned i'm Travis

6
00:00:18,080 --> 00:00:22,640
Howerton with uh RegScale where i'm the

7
00:00:20,080 --> 00:00:25,199
co-founder and chief technology officer

8
00:00:22,640 --> 00:00:27,519
i run the r d division uh responsible

9
00:00:25,199 --> 00:00:29,279

for building the platform so definitely

10
00:00:27,519 --> 00:00:32,320
definitely on the technical side and

11
00:00:29,279 --> 00:00:34,239
we'll abide by the technical rules

12
00:00:32,320 --> 00:00:36,320
most of this is going to be a live demo

13
00:00:34,239 --> 00:00:39,040
just showing what we can technically do

14
00:00:36,320 --> 00:00:42,239
how we leverage OSCAL and how we

15
00:00:39,040 --> 00:00:42,239
leverage it at the

16
00:00:42,640 --> 00:00:49,600
full spectrum of the stack from catalogs

17
00:00:45,680 --> 00:00:52,239
to profiles ssps components sap stars

18
00:00:49,600 --> 00:00:55,520
poems um we've got we've basically built

19
00:00:52,239 --> 00:00:57,520
our entire platform around the OSCAL

20
00:00:55,520 --> 00:00:59,680
standard as a

21
00:00:57,520 --> 00:01:00,719
underlying foundation

22
00:00:59,680 --> 00:01:02,559
and so

23
00:01:00,719 --> 00:01:04,799
for the agenda today we're going to go

24
00:01:02,559 --> 00:01:06,720
through just a quick background on me so

25
00:01:04,799 --> 00:01:08,720
you know who's talking to you

26
00:01:06,720 --> 00:01:12,000

um we're going to talk about our OSCAL

27
00:01:08,720 --> 00:01:14,159
support and sort of what we do there um

28
00:01:12,000 --> 00:01:16,720
of note we everything we're going to

29
00:01:14,159 --> 00:01:17,920
talk about today on the oscal side are

30
00:01:16,720 --> 00:01:20,000
free tools

31
00:01:17,920 --> 00:01:22,240
they are not open source tools we are a

32
00:01:20,000 --> 00:01:23,680
commercial vendor but everything we're

33
00:01:22,240 --> 00:01:25,600
going to show you is going to be out of

34
00:01:23,680 --> 00:01:28,560
the free side we're not going to show

35
00:01:25,600 --> 00:01:29,680
you anything on the paid commercial side

36
00:01:28,560 --> 00:01:31,040
today

37
00:01:29,680 --> 00:01:32,000
we're going to show again that full

38
00:01:31,040 --> 00:01:33,600
stack

39
00:01:32,000 --> 00:01:35,759
and again this is all done out of our

40
00:01:33,600 --> 00:01:38,560
community edition which is

41
00:01:35,759 --> 00:01:39,759
downloadable off the web again for free

42
00:01:38,560 --> 00:01:41,759
we're going to show you some new

43
00:01:39,759 --> 00:01:43,920

features um that we haven't talked to

44
00:01:41,759 --> 00:01:46,159
the uh through the nist audience too

45
00:01:43,920 --> 00:01:48,000
about dynamic oscal content authoring

46
00:01:46,159 --> 00:01:49,520
and some of what we're doing there

47
00:01:48,000 --> 00:01:50,880
we've taken a lot of feedback from the

48
00:01:49,520 --> 00:01:52,960
community and tried to make that

49
00:01:50,880 --> 00:01:54,399
experience better

50
00:01:52,960 --> 00:01:57,200
we're talking about an interesting

51
00:01:54,399 --> 00:01:59,439
integration we did with vitg on threat

52
00:01:57,200 --> 00:02:02,320
based risk modeling and sort of control

53
00:01:59,439 --> 00:02:05,200
tailoring uh that we did with a partner

54
00:02:02,320 --> 00:02:07,680
um all based off OSCAL

55
00:02:05,200 --> 00:02:08,800
and then the last bit i'm really excited

56
00:02:07,680 --> 00:02:10,959
to show you is we're going to do some

57
00:02:08,800 --> 00:02:12,800
hands-on demo work with the rec scale

58
00:02:10,959 --> 00:02:15,760
command line interface

59
00:02:12,800 --> 00:02:18,319
so this is a new cli we did for bulk

60
00:02:15,760 --> 00:02:20,480

processing oscal data to get it in and

61
00:02:18,319 --> 00:02:22,560
out of rig scale or other platforms

62
00:02:20,480 --> 00:02:24,640
convert between formats

63
00:02:22,560 --> 00:02:26,720
do lots of things on the command line

64
00:02:24,640 --> 00:02:29,360
that just make life easier to script an

65
00:02:26,720 --> 00:02:31,680
automated scale

66
00:02:29,360 --> 00:02:32,640
going forward

67
00:02:31,680 --> 00:02:35,920
and so

68
00:02:32,640 --> 00:02:38,480
real quick my background 22 years in the

69
00:02:35,920 --> 00:02:40,239
us nuclear weapons program

70
00:02:38,480 --> 00:02:44,160
i started as a federal employee the

71
00:02:40,239 --> 00:02:45,760
federal cio at the y12 weapons plant

72
00:02:44,160 --> 00:02:47,760
kind of rose up through the ranks over

73
00:02:45,760 --> 00:02:49,440
the years and became the first chief

74
00:02:47,760 --> 00:02:52,000
technology officer of the u.s nuclear

75
00:02:49,440 --> 00:02:54,640
weapons program where i met anil

76
00:02:52,000 --> 00:02:57,840
my fellow co-founder i'm here so anil's

77
00:02:54,640 --> 00:02:59,599

on the call he's our ceo um lead sort of

78
00:02:57,840 --> 00:03:01,120
the sales and operations side of the

79
00:02:59,599 --> 00:03:03,360
business

80
00:03:01,120 --> 00:03:05,120
i'm a former deputy ceo at oak ridge

81
00:03:03,360 --> 00:03:07,200
national lab where he did lots of

82
00:03:05,120 --> 00:03:09,280
research

83
00:03:07,200 --> 00:03:11,440
so very familiar with sort of academic

84
00:03:09,280 --> 00:03:13,680
research support

85
00:03:11,440 --> 00:03:15,920
i got recruited by bechtel to lead the

86
00:03:13,680 --> 00:03:17,280
merger and watch while pantex

87
00:03:15,920 --> 00:03:20,480
and then

88
00:03:17,280 --> 00:03:22,560
uh left bechtel and have always had

89
00:03:20,480 --> 00:03:24,319
this uh feeling that sort of compliance

90
00:03:22,560 --> 00:03:26,959
was the equal and opposite force to

91
00:03:24,319 --> 00:03:29,200
digital transformation anytime we tried

92
00:03:26,959 --> 00:03:31,120
to do anything um

93
00:03:29,200 --> 00:03:33,280
sort of innovative in government sort of

94
00:03:31,120 --> 00:03:35,519

the compliance process always slowed us

95
00:03:33,280 --> 00:03:38,319
down and so that's when i joined denil

96
00:03:35,519 --> 00:03:40,159
as a cto i mean we've been building out

97
00:03:38,319 --> 00:03:42,640
reg scale which is a continuous

98
00:03:40,159 --> 00:03:45,280
compliance automation platform

99
00:03:42,640 --> 00:03:47,599
think of it like a next generation grc

100
00:03:45,280 --> 00:03:50,720
and it's built entirely from top to

101
00:03:47,599 --> 00:03:53,599
bottom on ausgal as the foundation

102
00:03:50,720 --> 00:03:54,720
and so we just closed our series around

103
00:03:53,599 --> 00:03:57,680
and what

104
00:03:54,720 --> 00:03:59,519
our our sort of difference is is we are

105
00:03:57,680 --> 00:04:02,560
an api driven

106
00:03:59,519 --> 00:04:04,640
tool that's based off compliance's code

107
00:04:02,560 --> 00:04:07,200
and sort of real-time self-updating

108
00:04:04,640 --> 00:04:10,000
paperwork so we want to use OSCAL to

109
00:04:07,200 --> 00:04:12,560
free people from paperwork make machines

110
00:04:10,000 --> 00:04:16,239
talk to machines self-update and deliver

111
00:04:12,560 --> 00:04:17,280

a great human and machine experience by

112
00:04:16,239 --> 00:04:18,479
leveraging

113
00:04:17,280 --> 00:04:20,079
OSCAL

114
00:04:18,479 --> 00:04:21,759
and so everything we're going to talk

115
00:04:20,079 --> 00:04:23,520
from here on is just sort of how we're

116
00:04:21,759 --> 00:04:24,479
doing some of those things and to give

117
00:04:23,520 --> 00:04:27,199
you some

118
00:04:24,479 --> 00:04:28,800
some demos of our capabilities in that

119
00:04:27,199 --> 00:04:32,720
space i'm going to jump between the

120
00:04:28,800 --> 00:04:35,280
powerpoint to different live feeds um

121
00:04:32,720 --> 00:04:39,199
and show you some actual oscal

122
00:04:35,280 --> 00:04:40,960
exporting and everything uh are working

123
00:04:39,199 --> 00:04:43,120
so

124
00:04:40,960 --> 00:04:45,360
early on we got to work with nests in

125
00:04:43,120 --> 00:04:47,919
the early days as an open source project

126
00:04:45,360 --> 00:04:50,880
we wanted to contribute to that um in

127
00:04:47,919 --> 00:04:51,600
our former services company c2 labs

128
00:04:50,880 --> 00:04:53,759

and

129
00:04:51,600 --> 00:04:56,800
as we saw it going we saw immediately

130
00:04:53,759 --> 00:04:59,840
the promise of this um for

131
00:04:56,800 --> 00:05:01,440
uh streamlining a lot of workflow around

132
00:04:59,840 --> 00:05:03,759
staying compliant

133
00:05:01,440 --> 00:05:05,919
but we saw some also some gaps in

134
00:05:03,759 --> 00:05:08,639
tooling that we thought might be an

135
00:05:05,919 --> 00:05:10,960
opportunity for us to do some r d and so

136
00:05:08,639 --> 00:05:13,600
one of those um is we thought that the

137
00:05:10,960 --> 00:05:15,199
ability to generate oscal

138
00:05:13,600 --> 00:05:17,600
people were going to need tools that

139
00:05:15,199 --> 00:05:21,199
made that easier to do so they're not

140
00:05:17,600 --> 00:05:23,199
sort of coding xml and json and yaml

141
00:05:21,199 --> 00:05:25,520
files by hand

142
00:05:23,199 --> 00:05:28,160
also nist was going to put out

143
00:05:25,520 --> 00:05:30,639
stuff around 853 and their stuff in

144
00:05:28,160 --> 00:05:33,600
moscow but there's a whole universe of

145
00:05:30,639 --> 00:05:35,919

other things um even in this like 800

146
00:05:33,600 --> 00:05:37,440
171 or csf

147
00:05:35,919 --> 00:05:41,039
in moscow

148
00:05:37,440 --> 00:05:42,800
the iso 27000 sock 2 kind of world hipaa

149
00:05:41,039 --> 00:05:44,639
pci

150
00:05:42,800 --> 00:05:47,360
cloud security alliance there's a bunch

151
00:05:44,639 --> 00:05:49,120
of folks who publish control frameworks

152
00:05:47,360 --> 00:05:50,800
and we thought we should make tools that

153
00:05:49,120 --> 00:05:53,520
help digitize those and we're going to

154
00:05:50,800 --> 00:05:55,199
show you some of that um we also

155
00:05:53,520 --> 00:05:57,680
realized early on that most of the

156
00:05:55,199 --> 00:06:00,240
people who leverage these kind of tools

157
00:05:57,680 --> 00:06:02,400
don't have engineering backgrounds to go

158
00:06:00,240 --> 00:06:03,840
convert it to machine readable by hand

159
00:06:02,400 --> 00:06:04,880
themselves

160
00:06:03,840 --> 00:06:07,600
so that

161
00:06:04,880 --> 00:06:09,039
they're going to need

162
00:06:07,600 --> 00:06:12,080

tools that could help

163
00:06:09,039 --> 00:06:13,600
and we also assume that like moving in

164
00:06:12,080 --> 00:06:14,720
this direction is not going to be

165
00:06:13,600 --> 00:06:16,240
something people are going to want to

166
00:06:14,720 --> 00:06:19,120
pay a premium for

167
00:06:16,240 --> 00:06:21,120
and so for that reason we published uh

168
00:06:19,120 --> 00:06:23,520
reg scale community edition

169
00:06:21,120 --> 00:06:25,280
all the OSCAL support is in it and so

170
00:06:23,520 --> 00:06:29,120
everything i'm going to show you can do

171
00:06:25,280 --> 00:06:30,720
for free in those tools and so

172
00:06:29,120 --> 00:06:34,160
when you look at rec scale this is an

173
00:06:30,720 --> 00:06:36,720
example of nist 800 171 we were able to

174
00:06:34,160 --> 00:06:38,560
digitize and bring that indirect scale

175
00:06:36,720 --> 00:06:39,440
i'm going to show you some actual how we

176
00:06:38,560 --> 00:06:41,039
do it

177
00:06:39,440 --> 00:06:42,720
um

178
00:06:41,039 --> 00:06:45,759
but the upside is we're able to one

179
00:06:42,720 --> 00:06:48,479

click button and output a catalog based

180
00:06:45,759 --> 00:06:49,919
off 800 171 in austin

181
00:06:48,479 --> 00:06:51,120
and so

182
00:06:49,919 --> 00:06:53,599
i'm going to show you the free tools

183
00:06:51,120 --> 00:06:55,680
we've got how we do it there's multiple

184
00:06:53,599 --> 00:06:58,319
ways to do this you can literally type

185
00:06:55,680 --> 00:07:00,880
by hand into a gui you can copy and

186
00:06:58,319 --> 00:07:03,199
paste things from word or pdf forever

187
00:07:00,880 --> 00:07:04,960
into the gui but most of the time what

188
00:07:03,199 --> 00:07:07,759
we do and what we work with partners to

189
00:07:04,960 --> 00:07:10,479
do is build scripts that leverage our

190
00:07:07,759 --> 00:07:12,160
apis the bulk process and upload the

191
00:07:10,479 --> 00:07:14,240
data

192
00:07:12,160 --> 00:07:18,240
and the important thing is we're not

193
00:07:14,240 --> 00:07:20,880
just using oscal for cyber in 853

194
00:07:18,240 --> 00:07:23,199
related use cases we can take any

195
00:07:20,880 --> 00:07:26,400
catalog of controls or requirements and

196
00:07:23,199 --> 00:07:29,360

then rip it and convert it over to

197
00:07:26,400 --> 00:07:30,880
an OSCAL published catalog version and

198
00:07:29,360 --> 00:07:32,479
so we can do that with things like

199
00:07:30,880 --> 00:07:34,400
nuclear safety requirements

200
00:07:32,479 --> 00:07:35,919
environmental requirements

201
00:07:34,400 --> 00:07:37,759
we're working with the air force on some

202
00:07:35,919 --> 00:07:40,319
management instructions they have and

203
00:07:37,759 --> 00:07:41,440
taking them from pdf into a digital

204
00:07:40,319 --> 00:07:43,680
catalog

205
00:07:41,440 --> 00:07:46,560
where we can export it as OSCAL

206
00:07:43,680 --> 00:07:48,319
so um this approach is pretty extensible

207
00:07:46,560 --> 00:07:51,039
and this team's done an awesome job with

208
00:07:48,319 --> 00:07:53,280
ascal making it to be sort of a robust

209
00:07:51,039 --> 00:07:55,039
extensible framework for modeling that

210
00:07:53,280 --> 00:07:58,479
sort of thing and so we're going to show

211
00:07:55,039 --> 00:08:00,879
you a couple examples of what we've done

212
00:07:58,479 --> 00:08:02,720
also we gave a talk back in the spring

213
00:08:00,879 --> 00:08:04,960

since then we've digitized 20 more

214
00:08:02,720 --> 00:08:06,240
catalogs so basically any customer who

215
00:08:04,960 --> 00:08:08,479
comes to us and says we have a

216
00:08:06,240 --> 00:08:09,919
requirement we publish all of those out

217
00:08:08,479 --> 00:08:12,639
on the web i'm going to show you where

218
00:08:09,919 --> 00:08:15,199
to find them in a minute um but we make

219
00:08:12,639 --> 00:08:17,599
them available in our proprietary format

220
00:08:15,199 --> 00:08:20,080
and the oscal format upon request so

221
00:08:17,599 --> 00:08:22,639
anything we've digitized for anyone

222
00:08:20,080 --> 00:08:24,240
is out on the web free for you to use

223
00:08:22,639 --> 00:08:26,240
download

224
00:08:24,240 --> 00:08:28,639
load into different tools that support

225
00:08:26,240 --> 00:08:30,560
OSCAL, but we try to get back to the

226
00:08:28,639 --> 00:08:33,519
ecosystem in that way

227
00:08:30,560 --> 00:08:35,919
um the next thing we do is we we were at

228
00:08:33,519 --> 00:08:37,919
the control level before at moscow we

229
00:08:35,919 --> 00:08:39,680
couldn't go to the objective level we

230
00:08:37,919 --> 00:08:40,640

didn't have full support for parameters

231
00:08:39,680 --> 00:08:42,560
yet

232
00:08:40,640 --> 00:08:44,800
now we do we're going to show you an

233
00:08:42,560 --> 00:08:47,760
actual job run in a minute but we

234
00:08:44,800 --> 00:08:49,279
process all the objectives parameters

235
00:08:47,760 --> 00:08:51,440
we have the ability to pre-load

236
00:08:49,279 --> 00:08:53,920
implementation options and so we can

237
00:08:51,440 --> 00:08:56,240
build a library of options for how you

238
00:08:53,920 --> 00:08:58,160
satisfy this control that you can select

239
00:08:56,240 --> 00:08:59,440
from that kind of eases the building of

240
00:08:58,160 --> 00:09:02,160
a plan

241
00:08:59,440 --> 00:09:04,480
for engineers we preload the tests and

242
00:09:02,160 --> 00:09:07,680
we've got a dod customer we've we've

243
00:09:04,480 --> 00:09:09,279
also added support for cci's which is

244
00:09:07,680 --> 00:09:10,959
important for integration with stig

245
00:09:09,279 --> 00:09:12,080
scanners and so

246
00:09:10,959 --> 00:09:14,480
um

247
00:09:12,080 --> 00:09:16,880

lots of in-depth support on the oscal

248
00:09:14,480 --> 00:09:18,880
side that continues to be something we

249
00:09:16,880 --> 00:09:20,480
invest in and make richer

250
00:09:18,880 --> 00:09:22,800
and so here's where i'm going to start

251
00:09:20,480 --> 00:09:25,360
jumping around a little bit and so i'm

252
00:09:22,800 --> 00:09:28,959
going to start by going to our

253
00:09:25,360 --> 00:09:28,959
catalogs that we have available

254
00:09:33,360 --> 00:09:39,440
you can see rickscale.com regulations we

255
00:09:36,560 --> 00:09:41,040
have over 80 today you can search for

256
00:09:39,440 --> 00:09:41,760
you know anything that's of interest to

257
00:09:41,040 --> 00:09:43,279
you

258
00:09:41,760 --> 00:09:45,440
uh maybe you want to look at the cloud

259
00:09:43,279 --> 00:09:47,360
security alliance cloud controls matrix

260
00:09:45,440 --> 00:09:48,720
we have that one published and recently

261
00:09:47,360 --> 00:09:51,200
digitized it

262
00:09:48,720 --> 00:09:53,600
but there's 80 of these from commercial

263
00:09:51,200 --> 00:09:55,519
to international to government standards

264
00:09:53,600 --> 00:09:58,080

if they put out a control framework and

265
00:09:55,519 --> 00:10:00,640
we're aware of it we've published it one

266
00:09:58,080 --> 00:10:03,360
click to download it and uh get started

267
00:10:00,640 --> 00:10:05,600
with it in rich scale community edition

268
00:10:03,360 --> 00:10:09,360
all of these are available upon request

269
00:10:05,600 --> 00:10:09,360
in oscao format as well

270
00:10:09,519 --> 00:10:13,200
um so the next thing is let's show you

271
00:10:11,519 --> 00:10:14,880
one of these that we loaded

272
00:10:13,200 --> 00:10:17,360
actually show you what that looks like

273
00:10:14,880 --> 00:10:20,079
in direct scale so we took our command

274
00:10:17,360 --> 00:10:24,560
line interface and we loaded

275
00:10:20,079 --> 00:10:27,680
uh 853 in this case this is rev4 catalog

276
00:10:24,560 --> 00:10:30,720
it's got all 922 controls in it you can

277
00:10:27,680 --> 00:10:33,519
see the front matter and metadata here

278
00:10:30,720 --> 00:10:35,040
each of the controls we can go into

279
00:10:33,519 --> 00:10:36,720
um

280
00:10:35,040 --> 00:10:39,200
and you'll see here

281
00:10:36,720 --> 00:10:42,480

here's the control it's family all this

282
00:10:39,200 --> 00:10:44,959
comes directly from the nest catalog we

283
00:10:42,480 --> 00:10:47,920
bring in some of the OSCAL marking

284
00:10:44,959 --> 00:10:49,600
as well um we bring in all of the

285
00:10:47,920 --> 00:10:51,200
objectives that are defined for that

286
00:10:49,600 --> 00:10:54,000
control

287
00:10:51,200 --> 00:10:57,519
any parameters and we also have brought

288
00:10:54,000 --> 00:10:59,839
in 853 a from the published version of

289
00:10:57,519 --> 00:11:02,480
that so if there are any tests or

290
00:10:59,839 --> 00:11:05,040
auditing that goes with that control we

291
00:11:02,480 --> 00:11:06,160
bring in those defaults as well and then

292
00:11:05,040 --> 00:11:08,640
in other

293
00:11:06,160 --> 00:11:10,480
instances we brought in the cci's and

294
00:11:08,640 --> 00:11:12,959
mapped them as well so we have sort of a

295
00:11:10,480 --> 00:11:14,480
unified view of that control and then we

296
00:11:12,959 --> 00:11:16,320
use it for different things i'm going to

297
00:11:14,480 --> 00:11:19,839
demo later

298
00:11:16,320 --> 00:11:22,079

another example of this that's not nist

299
00:11:19,839 --> 00:11:24,000
is we've done some work recently in

300
00:11:22,079 --> 00:11:27,040
financial services sector

301
00:11:24,000 --> 00:11:28,880
where the cyber risk institute or cri

302
00:11:27,040 --> 00:11:29,839
publishes a set of catalogs that are

303
00:11:28,880 --> 00:11:31,839
tiered

304
00:11:29,839 --> 00:11:35,040
and so in this case we brought in their

305
00:11:31,839 --> 00:11:39,279
tier 4 catalog which is the lowest

306
00:11:35,040 --> 00:11:42,640
tier you can see their 137 controls

307
00:11:39,279 --> 00:11:45,680
but if i export that click the export i

308
00:11:42,640 --> 00:11:48,399
can now in a single click go grab the

309
00:11:45,680 --> 00:11:50,160
oscal file you will see that cri has

310
00:11:48,399 --> 00:11:53,360
some naming conventions that then this

311
00:11:50,160 --> 00:11:54,880
validator doesn't like um but it does go

312
00:11:53,360 --> 00:11:56,399
ahead and create it and so when i

313
00:11:54,880 --> 00:11:59,920
download that

314
00:11:56,399 --> 00:11:59,920
i can open it

315
00:12:02,399 --> 00:12:08,399

and you will see here's the oscal for

316
00:12:04,639 --> 00:12:10,560
the cri catalog and so it in a single

317
00:12:08,399 --> 00:12:13,120
click of a button once it's in here it

318
00:12:10,560 --> 00:12:15,360
can always export as OSCAL that does not

319
00:12:13,120 --> 00:12:17,440
mean it will fully meet every validation

320
00:12:15,360 --> 00:12:19,920
rule but you will see all the controls

321
00:12:17,440 --> 00:12:22,639
the front matter the properties

322
00:12:19,920 --> 00:12:24,880
um are all organized in accordance with

323
00:12:22,639 --> 00:12:26,240
the oscal standard and available in a

324
00:12:24,880 --> 00:12:27,440
click of a button

325
00:12:26,240 --> 00:12:30,079
um so

326
00:12:27,440 --> 00:12:33,680
we try to make it easy to bring those in

327
00:12:30,079 --> 00:12:36,079
um when we bring them in it is always

328
00:12:33,680 --> 00:12:38,399
almost always done through our apis that

329
00:12:36,079 --> 00:12:40,880
you see here so everything i'll show you

330
00:12:38,399 --> 00:12:43,600
in the gui today we have an application

331
00:12:40,880 --> 00:12:46,079
programming interface or api for

332
00:12:43,600 --> 00:12:50,399

um that'll allow you to script and bulk

333
00:12:46,079 --> 00:12:53,680
upload artifacts um indirect scale using

334
00:12:50,399 --> 00:12:55,279
osteo and so um this one was done using

335
00:12:53,680 --> 00:12:57,360
that methodology and i'm going to walk

336
00:12:55,279 --> 00:13:00,000
through a little bit more of that here

337
00:12:57,360 --> 00:13:02,639
shortly um but that's just an example

338
00:13:00,000 --> 00:13:05,440
and a real world demo of how we actually

339
00:13:02,639 --> 00:13:07,600
loaded 853 i'm going to run that routine

340
00:13:05,440 --> 00:13:09,440
for you later in this demo

341
00:13:07,600 --> 00:13:11,120
but also shows you other ones that we've

342
00:13:09,440 --> 00:13:12,880
digitized

343
00:13:11,120 --> 00:13:13,920
and to give you an example i want to run

344
00:13:12,880 --> 00:13:15,920
you through

345
00:13:13,920 --> 00:13:18,320
an example of what some code looks like

346
00:13:15,920 --> 00:13:20,399
so you say like how much work is it to

347
00:13:18,320 --> 00:13:23,120
actually digitize

348
00:13:20,399 --> 00:13:25,200
a regulation and pipe it through

349
00:13:23,120 --> 00:13:27,120

we can take a look again

350
00:13:25,200 --> 00:13:29,360
at some example code

351
00:13:27,120 --> 00:13:31,760
so here i'm going to go into this is a

352
00:13:29,360 --> 00:13:34,240
internal script library we have

353
00:13:31,760 --> 00:13:36,160
but this is for cis version eight

354
00:13:34,240 --> 00:13:38,399
control library and so we did

355
00:13:36,160 --> 00:13:41,600
implementation group three

356
00:13:38,399 --> 00:13:43,920
um so you log into this program you

357
00:13:41,600 --> 00:13:46,000
provide it your creds it logs in direct

358
00:13:43,920 --> 00:13:48,560
scale it gets a token

359
00:13:46,000 --> 00:13:50,560
it uses that token then to go

360
00:13:48,560 --> 00:13:53,519
um create a catalog

361
00:13:50,560 --> 00:13:55,360
in this case we just hard code the uh

362
00:13:53,519 --> 00:13:57,360
catalog information that we pulled from

363
00:13:55,360 --> 00:14:00,720
the website from them

364
00:13:57,360 --> 00:14:03,360
in their case they actually post um an

365
00:14:00,720 --> 00:14:05,600
excel file that has all their controls

366
00:14:03,360 --> 00:14:08,880

we have another routine that just runs a

367
00:14:05,600 --> 00:14:11,760
panda takes it out of excel dumps it

368
00:14:08,880 --> 00:14:14,240
into a json object

369
00:14:11,760 --> 00:14:17,199
we then go iterate over that so we load

370
00:14:14,240 --> 00:14:19,519
it into memory we iterate over it

371
00:14:17,199 --> 00:14:22,639
for each one of those we do a basic

372
00:14:19,519 --> 00:14:23,600
mapping job so we map their data into

373
00:14:22,639 --> 00:14:24,959
our

374
00:14:23,600 --> 00:14:27,839
schema

375
00:14:24,959 --> 00:14:30,240
we loop over that for all the controls

376
00:14:27,839 --> 00:14:32,560
we then post that back to our api to

377
00:14:30,240 --> 00:14:34,720
create a new security control underneath

378
00:14:32,560 --> 00:14:36,399
that catalog we just created

379
00:14:34,720 --> 00:14:39,040
we continue to do that until we've

380
00:14:36,399 --> 00:14:40,399
created all the controls in the catalog

381
00:14:39,040 --> 00:14:43,519
we get a total

382
00:14:40,399 --> 00:14:45,519
then we go query the catalog pull down

383
00:14:43,519 --> 00:14:47,519

all the controls

384
00:14:45,519 --> 00:14:49,760
loop through those make sure that all

385
00:14:47,519 --> 00:14:51,279
the titles and metadata match make sure

386
00:14:49,760 --> 00:14:54,079
the number matches that we didn't have

387
00:14:51,279 --> 00:14:56,320
any errors in terms of uploading and if

388
00:14:54,079 --> 00:14:57,279
all went well we we prompt a success

389
00:14:56,320 --> 00:14:59,519
message

390
00:14:57,279 --> 00:15:00,560
but you've got a little under 200 lines

391
00:14:59,519 --> 00:15:02,639
of code

392
00:15:00,560 --> 00:15:05,440
really the only difference in this code

393
00:15:02,639 --> 00:15:07,040
for any catalog is this mapping tool and

394
00:15:05,440 --> 00:15:08,880
maybe a little bit of the work in the

395
00:15:07,040 --> 00:15:10,560
pre-processing tool

396
00:15:08,880 --> 00:15:12,639
to get it done on the front side but

397
00:15:10,560 --> 00:15:14,079
we've built a software factory for

398
00:15:12,639 --> 00:15:16,800
getting this done

399
00:15:14,079 --> 00:15:19,040
once it's now loaded in cis

400
00:15:16,800 --> 00:15:22,079

we can hop over to reg scale

401
00:15:19,040 --> 00:15:26,160
ccif so here's the cis catalog version

402
00:15:22,079 --> 00:15:27,040
4. it's all loaded with its 141 controls

403
00:15:26,160 --> 00:15:28,880
um

404
00:15:27,040 --> 00:15:32,000
and then the exact same thing i can go

405
00:15:28,880 --> 00:15:34,720
export it with a click of a button take

406
00:15:32,000 --> 00:15:37,279
that cis version eight export it as

407
00:15:34,720 --> 00:15:38,720
auscal and now you have an OSCAL version

408
00:15:37,279 --> 00:15:40,880
available for

409
00:15:38,720 --> 00:15:41,680
that standard space for anything you

410
00:15:40,880 --> 00:15:44,320
want to

411
00:15:41,680 --> 00:15:46,320
use it for so

412
00:15:44,320 --> 00:15:48,959
nice repeatable process we've done this

413
00:15:46,320 --> 00:15:51,759
for 80 plus catalogs today

414
00:15:48,959 --> 00:15:53,680
um and we continue to digitize anything

415
00:15:51,759 --> 00:15:54,880
customers stakeholders in the community

416
00:15:53,680 --> 00:15:56,639
access to

417
00:15:54,880 --> 00:15:59,360

and then we publish it again on the rec

418
00:15:56,639 --> 00:16:02,800
scale site for free use

419
00:15:59,360 --> 00:16:02,800
for anyone who'd like to use it

420
00:16:03,759 --> 00:16:08,959
so that's a little bit of sort of the

421
00:16:07,040 --> 00:16:11,440
front matter results

422
00:16:08,959 --> 00:16:13,680
scripting getting catalogs in once

423
00:16:11,440 --> 00:16:16,240
you've got a catalog you can create

424
00:16:13,680 --> 00:16:18,959
profiles in regscale and then profiles

425
00:16:16,240 --> 00:16:22,639
can be used to create security plans

426
00:16:18,959 --> 00:16:24,720
components um etc um here you see sort

427
00:16:22,639 --> 00:16:26,880
of a screenshot on the left

428
00:16:24,720 --> 00:16:29,839
what what we wanted to deliver was a

429
00:16:26,880 --> 00:16:32,959
nice human experience for real world

430
00:16:29,839 --> 00:16:34,000
creating SSPs that also then dumps to

431
00:16:32,959 --> 00:16:36,639
OSCAL

432
00:16:34,000 --> 00:16:39,040
and so what we did here is when you

433
00:16:36,639 --> 00:16:41,600
create an SSP it's sort of a turbo tax

434
00:16:39,040 --> 00:16:44,800

lock experience next next finish

435
00:16:41,600 --> 00:16:46,720
um select your profile pops out

436
00:16:44,800 --> 00:16:48,800
and you get your controls it will

437
00:16:46,720 --> 00:16:50,639
preload the objectives and the

438
00:16:48,800 --> 00:16:52,720
parameters and the tests from that

439
00:16:50,639 --> 00:16:53,920
catalog we just created

440
00:16:52,720 --> 00:16:56,320
um

441
00:16:53,920 --> 00:16:58,800
we could not do that before so now all

442
00:16:56,320 --> 00:17:02,000
that loads into the gui at the control

443
00:16:58,800 --> 00:17:04,559
level the upside is you can now edit and

444
00:17:02,000 --> 00:17:06,959
in real time it's going to update your

445
00:17:04,559 --> 00:17:07,919
policy statement that comes from nist

446
00:17:06,959 --> 00:17:10,559
here

447
00:17:07,919 --> 00:17:13,039
and fill it out real time for you and

448
00:17:10,559 --> 00:17:15,760
then when you export that ssp all that

449
00:17:13,039 --> 00:17:17,679
data is going to dump out as valid oscal

450
00:17:15,760 --> 00:17:20,640
that you could submit your ssp your

451
00:17:17,679 --> 00:17:23,199

component on to an ao or to whoever is

452
00:17:20,640 --> 00:17:25,919
going to authorize it check it

453
00:17:23,199 --> 00:17:28,720
on the other side maybe a fedramp office

454
00:17:25,919 --> 00:17:30,320
for example and so

455
00:17:28,720 --> 00:17:32,559
i'm going to show you how you do this so

456
00:17:30,320 --> 00:17:34,880
we're going to go directly into an

457
00:17:32,559 --> 00:17:37,840
actual uh control

458
00:17:34,880 --> 00:17:40,720
if we go to the next tab

459
00:17:37,840 --> 00:17:43,039
here we're on ac2 so you see here's the

460
00:17:40,720 --> 00:17:44,720
language from ac2 along with its

461
00:17:43,039 --> 00:17:47,600
objectives picked ac2 because it's

462
00:17:44,720 --> 00:17:49,039
pretty robust in terms of the parameters

463
00:17:47,600 --> 00:17:51,440
and other things

464
00:17:49,039 --> 00:17:54,000
so if i go to a parameter

465
00:17:51,440 --> 00:17:56,640
i see we have ec2

466
00:17:54,000 --> 00:17:58,400
this is organization defined personnel

467
00:17:56,640 --> 00:18:03,440
enroll so i'm just going to say this

468
00:17:58,400 --> 00:18:03,440

apply to the ceo and cto of rec scale

469
00:18:04,160 --> 00:18:06,720
bring

470
00:18:05,360 --> 00:18:09,039
save it

471
00:18:06,720 --> 00:18:12,240
now when i go back to my requirement

472
00:18:09,039 --> 00:18:14,960
that parameter is no longer red default

473
00:18:12,240 --> 00:18:17,280
it swaps it in real time with whatever

474
00:18:14,960 --> 00:18:20,000
parameter we put in so you're able to

475
00:18:17,280 --> 00:18:22,080
just swap parameters dynamically updates

476
00:18:20,000 --> 00:18:24,720
the policy in real time

477
00:18:22,080 --> 00:18:28,080
similarly we can go through objectives

478
00:18:24,720 --> 00:18:30,640
and so we have an objective 2 here it

479
00:18:28,080 --> 00:18:33,280
says identify and select organization

480
00:18:30,640 --> 00:18:36,080
defined account types

481
00:18:33,280 --> 00:18:37,679
to support business functions so

482
00:18:36,080 --> 00:18:39,280
here let's say we're going to create a

483
00:18:37,679 --> 00:18:40,840
new option and we're going to say we're

484
00:18:39,280 --> 00:18:42,480
going to

485
00:18:40,840 --> 00:18:43,760

restrict

486
00:18:42,480 --> 00:18:45,679
to

487
00:18:43,760 --> 00:18:47,280
brexscale

488
00:18:45,679 --> 00:18:49,520
admin group

489
00:18:47,280 --> 00:18:49,520
and

490
00:18:50,400 --> 00:18:53,120
nad

491
00:18:51,679 --> 00:18:54,720
we're going to tag that as fully

492
00:18:53,120 --> 00:18:56,400
implemented

493
00:18:54,720 --> 00:18:58,880
save it

494
00:18:56,400 --> 00:19:01,120
i can add any notes i can attach

495
00:18:58,880 --> 00:19:03,120
screenshots or anything else but i'm

496
00:19:01,120 --> 00:19:05,440
going to complete that assessment and

497
00:19:03,120 --> 00:19:07,120
now that shows fully implemented so now

498
00:19:05,440 --> 00:19:09,280
objective one and two

499
00:19:07,120 --> 00:19:11,840
the oscal would export showing that

500
00:19:09,280 --> 00:19:14,240
objective is met how it was met

501
00:19:11,840 --> 00:19:15,760
as you do more and more options those

502
00:19:14,240 --> 00:19:18,559

become available to the next person

503
00:19:15,760 --> 00:19:21,840
who's filling out an ssp in your org so

504
00:19:18,559 --> 00:19:23,919
you create sort of a library by uh

505
00:19:21,840 --> 00:19:26,000
objective of what's valid in your

506
00:19:23,919 --> 00:19:28,160
organization and again it will

507
00:19:26,000 --> 00:19:29,600
dynamically update so now you see these

508
00:19:28,160 --> 00:19:32,240
go from a

509
00:19:29,600 --> 00:19:35,440
not yet assessed um

510
00:19:32,240 --> 00:19:38,080
blue box to green so you're able to see

511
00:19:35,440 --> 00:19:39,520
dynamically where you're at in real time

512
00:19:38,080 --> 00:19:41,520
visually

513
00:19:39,520 --> 00:19:43,760
and completing your control

514
00:19:41,520 --> 00:19:45,520
by objective and you can move quickly

515
00:19:43,760 --> 00:19:48,720
from control to control

516
00:19:45,520 --> 00:19:50,000
search controls and just quickly go go

517
00:19:48,720 --> 00:19:51,600
through

518
00:19:50,000 --> 00:19:55,039
populate it all and then when you're

519
00:19:51,600 --> 00:19:56,880

done you go back to the ssp

520
00:19:55,039 --> 00:19:59,840
click export

521
00:19:56,880 --> 00:20:03,280
and again you'll get an oscal file um

522
00:19:59,840 --> 00:20:05,440
that you can download and uh uh

523
00:20:03,280 --> 00:20:06,880
will be valid and updated in real time

524
00:20:05,440 --> 00:20:08,400
with that information

525
00:20:06,880 --> 00:20:10,400
so we're trying to make the human

526
00:20:08,400 --> 00:20:12,799
experience of interacting with austral

527
00:20:10,400 --> 00:20:14,640
parameters and other things easier while

528
00:20:12,799 --> 00:20:17,120
also fully supporting a machine

529
00:20:14,640 --> 00:20:20,240
experience where it now outputs that

530
00:20:17,120 --> 00:20:21,679
file in real time for whatever use case

531
00:20:20,240 --> 00:20:24,640
you need to run scripts or anything

532
00:20:21,679 --> 00:20:24,640
you're going to do to check it

533
00:20:25,120 --> 00:20:27,679
um

534
00:20:25,680 --> 00:20:29,840
[Music]

535
00:20:27,679 --> 00:20:32,000
the other thing we can do there um

536
00:20:29,840 --> 00:20:33,840

hopping back

537
00:20:32,000 --> 00:20:36,720
is we're trying to support sort of two

538
00:20:33,840 --> 00:20:41,600
use cases i mentioned the uh machine use

539
00:20:36,720 --> 00:20:43,360
case but also the uh human use case so

540
00:20:41,600 --> 00:20:45,440
the other thing we've done is once

541
00:20:43,360 --> 00:20:47,679
you've built out an ssp you need to put

542
00:20:45,440 --> 00:20:49,200
it in continuous monitoring assess it

543
00:20:47,679 --> 00:20:51,200
over time

544
00:20:49,200 --> 00:20:53,039
make sure you're doing the right things

545
00:20:51,200 --> 00:20:56,400
i mentioned we would load we would have

546
00:20:53,039 --> 00:20:58,880
the default tests these come from 853a

547
00:20:56,400 --> 00:21:01,520
so when i click it it's pulling in those

548
00:20:58,880 --> 00:21:04,240
default tests that come from moscow

549
00:21:01,520 --> 00:21:06,960
i can now go do a lightning assessment

550
00:21:04,240 --> 00:21:08,960
so it tells me exactly what i should do

551
00:21:06,960 --> 00:21:11,360
as the best practice coming from this to

552
00:21:08,960 --> 00:21:13,679
assess this control i can put past

553
00:21:11,360 --> 00:21:16,320

partial paths whatever it may be this

554
00:21:13,679 --> 00:21:17,760
will feed your sap sar that goes with it

555
00:21:16,320 --> 00:21:19,760
as you assess

556
00:21:17,760 --> 00:21:21,200
the controls and you can output your sap

557
00:21:19,760 --> 00:21:23,760
sorry as well

558
00:21:21,200 --> 00:21:25,760
so this is sort of the human-based way

559
00:21:23,760 --> 00:21:27,440
of doing that you write up your notes

560
00:21:25,760 --> 00:21:29,039
that go with it you attach your audit

561
00:21:27,440 --> 00:21:30,480
files or evidence

562
00:21:29,039 --> 00:21:33,600
and you complete your lightning

563
00:21:30,480 --> 00:21:36,320
assessment and it's done

564
00:21:33,600 --> 00:21:38,640
once it's done it'll re-update the page

565
00:21:36,320 --> 00:21:41,760
the other thing that we're doing um

566
00:21:38,640 --> 00:21:44,720
that's interesting is we see oscal as a

567
00:21:41,760 --> 00:21:46,240
base for machine to machine automation

568
00:21:44,720 --> 00:21:48,080
so the other way we're doing this is

569
00:21:46,240 --> 00:21:50,880
we're integrating with commercial tools

570
00:21:48,080 --> 00:21:53,039

and our apis so maybe it's a cloud

571
00:21:50,880 --> 00:21:54,240
security posture management tool like

572
00:21:53,039 --> 00:21:56,880
wiz or

573
00:21:54,240 --> 00:21:59,840
uh prisma maybe it's a scanner like

574
00:21:56,880 --> 00:22:01,280
tenable or qualis or a stick tool like

575
00:21:59,840 --> 00:22:04,240
steel cloud

576
00:22:01,280 --> 00:22:07,440
if they can output things in oscao on

577
00:22:04,240 --> 00:22:10,400
their side um or any other structured

578
00:22:07,440 --> 00:22:12,080
format we bring that in as a poem

579
00:22:10,400 --> 00:22:14,400
showing that it was identified so in

580
00:22:12,080 --> 00:22:16,320
this case is a machine scan

581
00:22:14,400 --> 00:22:18,400
it found a vulnerability this is

582
00:22:16,320 --> 00:22:20,080
actually in the azure infrastructure

583
00:22:18,400 --> 00:22:21,360
layer which was interesting so we never

584
00:22:20,080 --> 00:22:24,000
would have seen it

585
00:22:21,360 --> 00:22:27,039
um and then it was fixed by microsoft

586
00:22:24,000 --> 00:22:29,679
inside of 30 days and i can go in and i

587
00:22:27,039 --> 00:22:32,000

can see the scanner that found it

588
00:22:29,679 --> 00:22:35,120
and so if i click that link it opens up

589
00:22:32,000 --> 00:22:36,880
the actual wiz ticket of here's the tool

590
00:22:35,120 --> 00:22:39,280
that went out and found this problem

591
00:22:36,880 --> 00:22:41,280
with it identified it you can see here

592
00:22:39,280 --> 00:22:43,440
they map to these frameworks we're using

593
00:22:41,280 --> 00:22:44,960
that to update it to the proper control

594
00:22:43,440 --> 00:22:47,840
and rank scale

595
00:22:44,960 --> 00:22:49,840
as part of that sap sar assessment layer

596
00:22:47,840 --> 00:22:52,240
and then we can even look at the jira or

597
00:22:49,840 --> 00:22:54,480
servicenow ticket to go remediate it and

598
00:22:52,240 --> 00:22:57,520
we keep it all tied off so that your

599
00:22:54,480 --> 00:23:00,320
poem register when you dump your poems

600
00:22:57,520 --> 00:23:03,039
out um has all the information around

601
00:23:00,320 --> 00:23:06,159
what found it um who fixed it on the

602
00:23:03,039 --> 00:23:09,120
itil side it keeps that up to date with

603
00:23:06,159 --> 00:23:11,200
automation again with austral as an

604
00:23:09,120 --> 00:23:12,400

underlying standard for uh how we're

605
00:23:11,200 --> 00:23:14,960
doing things

606
00:23:12,400 --> 00:23:17,360
and so we keep all that ticked and todd

607
00:23:14,960 --> 00:23:19,440
and just make it ideally a lot easier

608
00:23:17,360 --> 00:23:22,480
for people to to maintain their security

609
00:23:19,440 --> 00:23:26,880
posture with automation using compliance

610
00:23:22,480 --> 00:23:26,880
as code again based on uh

611
00:23:29,679 --> 00:23:35,280
some work in progress so some upcoming

612
00:23:32,320 --> 00:23:36,799
work so we see um

613
00:23:35,280 --> 00:23:39,200
continuing to support more and more

614
00:23:36,799 --> 00:23:42,000
formats so we're doing some emass and

615
00:23:39,200 --> 00:23:44,640
fedramp exports today those are in excel

616
00:23:42,000 --> 00:23:48,080
files is what they accept as they move

617
00:23:44,640 --> 00:23:48,880
to moscow we'll dump austral uh poems

618
00:23:48,080 --> 00:23:51,840
out

619
00:23:48,880 --> 00:23:55,039
as well uh we're adding more scanners so

620
00:23:51,840 --> 00:23:56,400
the more industry starts to uh add OSCAL

621
00:23:55,039 --> 00:23:58,640

support to their tools the more

622
00:23:56,400 --> 00:24:01,360
standards you can plug into a tool that

623
00:23:58,640 --> 00:24:03,120
supports moscow so this is again

624
00:24:01,360 --> 00:24:05,200
not anything we're trying to sell here

625
00:24:03,120 --> 00:24:08,000
just an example of how you can use

626
00:24:05,200 --> 00:24:09,840
auscal for extreme automation

627
00:24:08,000 --> 00:24:11,760
to take a lot of pain out of how people

628
00:24:09,840 --> 00:24:14,400
do things today and then we're adding

629
00:24:11,760 --> 00:24:16,400
more itil support we've got servicenow

630
00:24:14,400 --> 00:24:18,799
in jira we're going to add salesforce

631
00:24:16,400 --> 00:24:22,000
service cloud soon and we'll continue to

632
00:24:18,799 --> 00:24:24,720
add more scanners more itil tools

633
00:24:22,000 --> 00:24:27,440
and just more automation again with that

634
00:24:24,720 --> 00:24:29,919
oscal base as a way of outputting

635
00:24:27,440 --> 00:24:29,919
results

636
00:24:30,720 --> 00:24:34,640
the next one i'm just going to touch on

637
00:24:32,559 --> 00:24:37,279
um not demo in the interest of time so i

638
00:24:34,640 --> 00:24:40,080

can show you some of the cli stuff today

639
00:24:37,279 --> 00:24:41,120
um but we have a partner at vitg it's

640
00:24:40,080 --> 00:24:43,760
volpe

641
00:24:41,120 --> 00:24:45,279
it group they do work in the fedramp

642
00:24:43,760 --> 00:24:47,360
office and they've been doing a lot of

643
00:24:45,279 --> 00:24:50,159
threat based risk modeling using some of

644
00:24:47,360 --> 00:24:52,080
that dhs gov car model

645
00:24:50,159 --> 00:24:54,960
what we proved out with them is we could

646
00:24:52,080 --> 00:24:56,400
send them an ssp in moscow they could

647
00:24:54,960 --> 00:24:58,840
then run it through their risk

648
00:24:56,400 --> 00:25:01,760
thresholds and tailor

649
00:24:58,840 --> 00:25:03,760
um basically what control should be in

650
00:25:01,760 --> 00:25:05,360
or out of scope based on risk or where

651
00:25:03,760 --> 00:25:08,400
you're falling short

652
00:25:05,360 --> 00:25:10,159
by threat groups based on risk and the

653
00:25:08,400 --> 00:25:12,320
goal is to make it sort of faster and

654
00:25:10,159 --> 00:25:14,720
cheaper to get a fed ramp by doing some

655
00:25:12,320 --> 00:25:17,120

risk tailoring um

656
00:25:14,720 --> 00:25:19,279
here's just a screenshot of the results

657
00:25:17,120 --> 00:25:21,039
where we pass them a moderate system we

658
00:25:19,279 --> 00:25:23,919
built in rec scale

659
00:25:21,039 --> 00:25:26,960
we call an api on their side we pass

660
00:25:23,919 --> 00:25:29,520
them and ask an actual oscal file

661
00:25:26,960 --> 00:25:32,240
they then do stuff with it they run it

662
00:25:29,520 --> 00:25:33,919
through their checks it spins for 30 45

663
00:25:32,240 --> 00:25:36,159
seconds it comes back

664
00:25:33,919 --> 00:25:38,559
and based off the threshold we set in

665
00:25:36,159 --> 00:25:41,760
rec scale it will tell us what we can

666
00:25:38,559 --> 00:25:44,559
tailor out or where we are short

667
00:25:41,760 --> 00:25:46,000
in sort of being compliant against that

668
00:25:44,559 --> 00:25:48,240
risk threshold

669
00:25:46,000 --> 00:25:50,799
so again another example of machine to

670
00:25:48,240 --> 00:25:52,000
machine integration and automation using

671
00:25:50,799 --> 00:25:54,159
moscow

672
00:25:52,000 --> 00:25:55,919

there was no commonality between our

673
00:25:54,159 --> 00:25:58,480
tools in any way we hadn't worked

674
00:25:55,919 --> 00:25:59,440
together at all before this and we were

675
00:25:58,480 --> 00:26:01,840
able to

676
00:25:59,440 --> 00:26:05,200
pass back and forth and do a high value

677
00:26:01,840 --> 00:26:07,600
integration using oscal as a standard

678
00:26:05,200 --> 00:26:07,600
indian

679
00:26:08,799 --> 00:26:14,400
the next part is something we tease when

680
00:26:11,279 --> 00:26:16,720
we're at the spring workshop um but we

681
00:26:14,400 --> 00:26:18,559
are seeing use cases

682
00:26:16,720 --> 00:26:20,720
when you get into sort of this extreme

683
00:26:18,559 --> 00:26:21,679
automation you start to end up in a

684
00:26:20,720 --> 00:26:23,520
space where you're going to be

685
00:26:21,679 --> 00:26:26,080
processing a lot of data

686
00:26:23,520 --> 00:26:29,039
and so the web as a format to us didn't

687
00:26:26,080 --> 00:26:31,120
feel like a scalable way to do that um

688
00:26:29,039 --> 00:26:32,720
inside of reg scale so what we built is

689
00:26:31,120 --> 00:26:35,520

the reg scale

690
00:26:32,720 --> 00:26:38,480
cli or command line interface

691
00:26:35,520 --> 00:26:41,760
um what's important to know about it is

692
00:26:38,480 --> 00:26:44,240
it's basically it's a python library

693
00:26:41,760 --> 00:26:45,840
that uses a yaml file to tell it what to

694
00:26:44,240 --> 00:26:48,080
do and how to do it

695
00:26:45,840 --> 00:26:49,840
and then you execute your commands which

696
00:26:48,080 --> 00:26:52,480
express your intent and then it just

697
00:26:49,840 --> 00:26:55,840
bulk processes things um we're going to

698
00:26:52,480 --> 00:26:58,320
show you some of the free oscal stuff um

699
00:26:55,840 --> 00:27:00,559
that we do here um this is actually

700
00:26:58,320 --> 00:27:03,679
available on pophops you could do pip

701
00:27:00,559 --> 00:27:06,559
install rakescale cli and pull this down

702
00:27:03,679 --> 00:27:08,880
today and start playing with it um so it

703
00:27:06,559 --> 00:27:11,520
is available for free

704
00:27:08,880 --> 00:27:13,520
out there we've also got documentation

705
00:27:11,520 --> 00:27:16,000
on the rec scale website about how to

706
00:27:13,520 --> 00:27:17,200

use it example code some of the stuff

707
00:27:16,000 --> 00:27:19,200
i'm going to show you in work in

708
00:27:17,200 --> 00:27:21,200
progress we've got to get better docs up

709
00:27:19,200 --> 00:27:22,880
there but i'm going to show you that it

710
00:27:21,200 --> 00:27:24,559
does actually work and we're actually

711
00:27:22,880 --> 00:27:26,240
building documentation this week and

712
00:27:24,559 --> 00:27:28,880
next week to go with it

713
00:27:26,240 --> 00:27:30,559
but the goal is to be able to bulk

714
00:27:28,880 --> 00:27:34,000
process stuff

715
00:27:30,559 --> 00:27:36,080
in a modern cloud-native way using OSCAL

716
00:27:34,000 --> 00:27:38,000
and so we've taken that command line

717
00:27:36,080 --> 00:27:40,720
interface the next step is we

718
00:27:38,000 --> 00:27:42,559
containerized it so we've also published

719
00:27:40,720 --> 00:27:44,159
there's a link there on docker hub i'll

720
00:27:42,559 --> 00:27:45,679
share these slides with michaela for

721
00:27:44,159 --> 00:27:47,600
anyone who wants some in the future so

722
00:27:45,679 --> 00:27:49,679
you have links

723
00:27:47,600 --> 00:27:52,399

but we've also published the reg scale

724
00:27:49,679 --> 00:27:54,000
cli as a docker image and what that'll

725
00:27:52,399 --> 00:27:55,520
allow you to do is let's say you're in

726
00:27:54,000 --> 00:27:58,080
kubernetes you're using some cloud

727
00:27:55,520 --> 00:28:00,880
kubernetes service you can spin up a

728
00:27:58,080 --> 00:28:03,520
container it can run its jobs

729
00:28:00,880 --> 00:28:05,039
tear itself down do those asynchronously

730
00:28:03,520 --> 00:28:07,279
however you want to schedule and

731
00:28:05,039 --> 00:28:09,440
orchestrate it but it will allow you to

732
00:28:07,279 --> 00:28:11,840
be ephemeral where you spin it up you do

733
00:28:09,440 --> 00:28:13,360
whatever job you need you tear it down

734
00:28:11,840 --> 00:28:16,080
you use what capacity you have in the

735
00:28:13,360 --> 00:28:18,640
cluster but you don't need to size it

736
00:28:16,080 --> 00:28:21,039
for like peak load on a vm or anything

737
00:28:18,640 --> 00:28:22,960
like that so we are continuing to do

738
00:28:21,039 --> 00:28:23,840
more and more work that makes it easy to

739
00:28:22,960 --> 00:28:25,919
do

740
00:28:23,840 --> 00:28:28,080

bulk processing and what we call extreme

741
00:28:25,919 --> 00:28:29,440
automation with OSCAL because some of

742
00:28:28,080 --> 00:28:30,960
these integrations we're doing with

743
00:28:29,440 --> 00:28:33,360
scanner tools are going to pull a lot of

744
00:28:30,960 --> 00:28:35,360
data and that job may need to run

745
00:28:33,360 --> 00:28:37,279
for an hour or maybe you can spin it up

746
00:28:35,360 --> 00:28:38,159
in ten containers and get it done in six

747
00:28:37,279 --> 00:28:41,840
minutes

748
00:28:38,159 --> 00:28:44,080
right um so it's one of those um

749
00:28:41,840 --> 00:28:46,000
abilities to sort of bulk process stuff

750
00:28:44,080 --> 00:28:47,600
from the command line

751
00:28:46,000 --> 00:28:49,600
so on the right

752
00:28:47,600 --> 00:28:51,600
i'm actually going to run this um it

753
00:28:49,600 --> 00:28:53,200
probably won't have time to complete or

754
00:28:51,600 --> 00:28:54,480
we'll just barely complete by the time

755
00:28:53,200 --> 00:28:56,799
we finish here

756
00:28:54,480 --> 00:28:59,760
but um the command i ran there is

757
00:28:56,799 --> 00:29:01,360

basically it's rig scale oscal catalog

758
00:28:59,760 --> 00:29:03,279
and i'm giving it one of the nist

759
00:29:01,360 --> 00:29:06,399
catalogs so this is published in the

760
00:29:03,279 --> 00:29:09,279
OSCAL content github directory we took

761
00:29:06,399 --> 00:29:11,200
the 853 rev4 catalog

762
00:29:09,279 --> 00:29:14,159
you can see it goes out creates the

763
00:29:11,200 --> 00:29:15,200
catalog it finds the number of families

764
00:29:14,159 --> 00:29:17,360
um

765
00:29:15,200 --> 00:29:19,440
the number of controls all the

766
00:29:17,360 --> 00:29:21,360
parameters that are identified

767
00:29:19,440 --> 00:29:23,760
the parts which are the objectives and

768
00:29:21,360 --> 00:29:25,880
statements inside of it guidance and

769
00:29:23,760 --> 00:29:27,919
then any assessment procedures from the

770
00:29:25,880 --> 00:29:30,080
853a side

771
00:29:27,919 --> 00:29:32,399
loads all that and then just goes and

772
00:29:30,080 --> 00:29:35,440
creates all the objects in real time and

773
00:29:32,399 --> 00:29:38,559
rexscale using our apis and orchestrates

774
00:29:35,440 --> 00:29:41,600

all that for the cli so with one command

775
00:29:38,559 --> 00:29:42,559
you can create thousands of objects in

776
00:29:41,600 --> 00:29:45,760
minutes

777
00:29:42,559 --> 00:29:48,080
inside of rexscale using auscal via this

778
00:29:45,760 --> 00:29:50,080
command line interface and so

779
00:29:48,080 --> 00:29:51,279
i'm actually going to show you one here

780
00:29:50,080 --> 00:29:54,279
um so

781
00:29:51,279 --> 00:29:54,279
oops

782
00:29:54,720 --> 00:29:56,960
that

783
00:29:59,600 --> 00:30:03,840
what i'm going to do is open

784
00:30:01,840 --> 00:30:05,520
hopefully this works well and murphy's

785
00:30:03,840 --> 00:30:07,679
law doesn't pop up

786
00:30:05,520 --> 00:30:09,279
but we're going to do a demo here so i'm

787
00:30:07,679 --> 00:30:12,240
in uh

788
00:30:09,279 --> 00:30:15,120
a windows desktop here uh install

789
00:30:12,240 --> 00:30:17,840
windows windows subsystem for linux

790
00:30:15,120 --> 00:30:20,559
um and i've got a ubuntu image that i'm

791
00:30:17,840 --> 00:30:25,039

running so i just did an ls you can see

792
00:30:20,559 --> 00:30:26,720
i've got the 853 rev4 json file there

793
00:30:25,039 --> 00:30:29,120
um

794
00:30:26,720 --> 00:30:30,640
that nist publishes i've got my yaml

795
00:30:29,120 --> 00:30:32,559
file that i'm not going to show that's

796
00:30:30,640 --> 00:30:34,640
got keys and other things

797
00:30:32,559 --> 00:30:36,559
um that lets it run

798
00:30:34,640 --> 00:30:38,799
and i'm going to run a command

799
00:30:36,559 --> 00:30:40,880
and what this is saying is go to use the

800
00:30:38,799 --> 00:30:42,000
rec scale cli which is just pip

801
00:30:40,880 --> 00:30:45,440
installed

802
00:30:42,000 --> 00:30:48,640
um use the oscill library we want to

803
00:30:45,440 --> 00:30:50,240
create a catalog based off this file it

804
00:30:48,640 --> 00:30:52,720
will do some validation to make sure

805
00:30:50,240 --> 00:30:54,720
it's a valid file it will open it and

806
00:30:52,720 --> 00:30:56,159
then it's going to run and then

807
00:30:54,720 --> 00:30:59,760
the screen is going to look like the

808
00:30:56,159 --> 00:31:01,440

matrix and so when i click it

809
00:30:59,760 --> 00:31:03,120
you can see

810
00:31:01,440 --> 00:31:05,360
it did that same thing you saw in the

811
00:31:03,120 --> 00:31:07,760
screenshot but what it's doing now is

812
00:31:05,360 --> 00:31:10,399
it's running through in real time every

813
00:31:07,760 --> 00:31:12,799
control you can see the success 200 is

814
00:31:10,399 --> 00:31:14,799
coming back for each of those controls

815
00:31:12,799 --> 00:31:17,519
it's posting them against our api

816
00:31:14,799 --> 00:31:19,120
creating the control with the controls

817
00:31:17,519 --> 00:31:21,760
are created it will create all the

818
00:31:19,120 --> 00:31:23,519
objectives parameters tests that go with

819
00:31:21,760 --> 00:31:25,679
it and it's going to create somewhere

820
00:31:23,519 --> 00:31:28,640
near 10 000 records by the time this is

821
00:31:25,679 --> 00:31:30,240
done it may take a few minutes to run um

822
00:31:28,640 --> 00:31:32,559
on this vm

823
00:31:30,240 --> 00:31:34,960
but all of that is running basically

824
00:31:32,559 --> 00:31:37,120
real time through the cli so i'm going

825
00:31:34,960 --> 00:31:39,039

to let that continue to run

826
00:31:37,120 --> 00:31:41,120
um that's going to end up creating a

827
00:31:39,039 --> 00:31:43,279
profile that i'll show you hopefully by

828
00:31:41,120 --> 00:31:44,399
the end of this presentation it should

829
00:31:43,279 --> 00:31:47,039
finish

830
00:31:44,399 --> 00:31:51,120
but in the whoops interim i'm going to

831
00:31:47,039 --> 00:31:51,120
go pull up another u by 2 vm

832
00:31:52,640 --> 00:31:56,720
and

833
00:31:53,919 --> 00:31:59,679
here i need to get another

834
00:31:56,720 --> 00:32:02,880
another one so once that catalogs loaded

835
00:31:59,679 --> 00:32:05,279
we also have a cli for uh profiles let

836
00:32:02,880 --> 00:32:06,159
me make sure i'm in the right folder

837
00:32:05,279 --> 00:32:08,240
um

838
00:32:06,159 --> 00:32:09,840
here i'm going to go

839
00:32:08,240 --> 00:32:14,000
create that so here

840
00:32:09,840 --> 00:32:16,159
same reg scale cli same oscal library

841
00:32:14,000 --> 00:32:17,200
this time we're using the profile sub

842
00:32:16,159 --> 00:32:19,039

command

843
00:32:17,200 --> 00:32:20,559
i'm giving it a title of what i want to

844
00:32:19,039 --> 00:32:22,640
name my profile

845
00:32:20,559 --> 00:32:24,159
i'm giving it a categorization so in

846
00:32:22,640 --> 00:32:26,960
this case it's low

847
00:32:24,159 --> 00:32:29,120
i'm giving it the idea the catalog that

848
00:32:26,960 --> 00:32:30,320
we created using that

849
00:32:29,120 --> 00:32:33,679
and then

850
00:32:30,320 --> 00:32:36,159
the actual profile that's defined in the

851
00:32:33,679 --> 00:32:38,240
NIST OSCAL repo which in this case we've

852
00:32:36,159 --> 00:32:39,360
named low profile this one runs much

853
00:32:38,240 --> 00:32:43,000
faster

854
00:32:39,360 --> 00:32:43,000
so if i click it

855
00:33:01,440 --> 00:33:04,080
something wrong

856
00:33:05,279 --> 00:33:10,720
did something wrong there but that one

857
00:33:06,799 --> 00:33:15,519
should create the uh OSCAL profile um

858
00:33:10,720 --> 00:33:17,440
based off of that um on the other side

859
00:33:15,519 --> 00:33:18,720

this is still running here creating out

860
00:33:17,440 --> 00:33:24,679
that catalog

861
00:33:18,720 --> 00:33:24,679
and what you will get at the end is a uh

862
00:33:26,880 --> 00:33:30,840
new catalog built

863
00:33:38,720 --> 00:33:43,519
you'll see this is building you have

864
00:33:40,480 --> 00:33:47,279
that new catalog number 62 you see we

865
00:33:43,519 --> 00:33:49,279
just created it's got its 922 controls

866
00:33:47,279 --> 00:33:51,279
it's still going through as you can see

867
00:33:49,279 --> 00:33:53,039
here and creating its

868
00:33:51,279 --> 00:33:55,519
assessment layer

869
00:33:53,039 --> 00:33:57,519
while we run but everything will be bulk

870
00:33:55,519 --> 00:34:00,399
loaded into the gui so one click to get

871
00:33:57,519 --> 00:34:01,760
it out one command to get it in can go

872
00:34:00,399 --> 00:34:04,240
back and forth

873
00:34:01,760 --> 00:34:06,559
laterally and can run at basically any

874
00:34:04,240 --> 00:34:07,919
scale provided you have capacity in your

875
00:34:06,559 --> 00:34:09,599
cluster to run it

876
00:34:07,919 --> 00:34:11,359

um

877
00:34:09,599 --> 00:34:13,520
and make that work and so i'm going to

878
00:34:11,359 --> 00:34:15,200
let that keep running while i come back

879
00:34:13,520 --> 00:34:16,800
over here

880
00:34:15,200 --> 00:34:19,679
i think i've got one or two more things

881
00:34:16,800 --> 00:34:21,040
to show and then we can hop into a q a

882
00:34:19,679 --> 00:34:22,399
with the group

883
00:34:21,040 --> 00:34:25,040
the other thing that we're running

884
00:34:22,399 --> 00:34:27,119
that's kind of interesting um

885
00:34:25,040 --> 00:34:29,520
we everything we do is json because

886
00:34:27,119 --> 00:34:31,599
we're we're a heavily urest based shop

887
00:34:29,520 --> 00:34:33,599
in architecture in rec scale

888
00:34:31,599 --> 00:34:35,919
but other people are not maybe you

889
00:34:33,599 --> 00:34:37,679
prefer yaml um you need tools that use

890
00:34:35,919 --> 00:34:40,399
ammo or xml

891
00:34:37,679 --> 00:34:42,079
um we've also built inside that

892
00:34:40,399 --> 00:34:43,919
container that i mentioned a set of

893
00:34:42,079 --> 00:34:46,240

tools for converters

894
00:34:43,919 --> 00:34:48,480
uh we've taken these from this we've

895
00:34:46,240 --> 00:34:50,159
pre-packaged and configured all that in

896
00:34:48,480 --> 00:34:51,359
the container so there's no setup on

897
00:34:50,159 --> 00:34:53,119
your side

898
00:34:51,359 --> 00:34:54,399
um so you'll see here you just do a

899
00:34:53,119 --> 00:34:56,879
docker run

900
00:34:54,399 --> 00:34:58,960
you mount wherever your files are so in

901
00:34:56,879 --> 00:35:02,560
this case it's in my uc

902
00:34:58,960 --> 00:35:04,480
drive users howie cli folder i'm

903
00:35:02,560 --> 00:35:06,480
mounting that to the data folder in the

904
00:35:04,480 --> 00:35:07,599
container i'm setting up an interactive

905
00:35:06,480 --> 00:35:10,400
terminal

906
00:35:07,599 --> 00:35:12,560
i'm pulling the latest version the cli

907
00:35:10,400 --> 00:35:13,680
container with an alpine interactive

908
00:35:12,560 --> 00:35:16,000
shell

909
00:35:13,680 --> 00:35:18,320
and what happens is if we come back over

910
00:35:16,000 --> 00:35:18,320

here

911
00:35:25,520 --> 00:35:28,880
happy with me

912
00:35:27,520 --> 00:35:31,440
yep

913
00:35:28,880 --> 00:35:33,440
over here you see i now have a running

914
00:35:31,440 --> 00:35:34,400
uh docker container

915
00:35:33,440 --> 00:35:36,079
um

916
00:35:34,400 --> 00:35:38,240
um uh

917
00:35:36,079 --> 00:35:39,839
got my data volume mounted and i'm

918
00:35:38,240 --> 00:35:42,400
inside of it with the interactive

919
00:35:39,839 --> 00:35:44,480
terminal where i can now run reg scale

920
00:35:42,400 --> 00:35:46,079
the upside of doing it this way is you

921
00:35:44,480 --> 00:35:48,480
don't have to worry about having the

922
00:35:46,079 --> 00:35:50,960
right python versions the right java

923
00:35:48,480 --> 00:35:52,480
versions having saxon installed and

924
00:35:50,960 --> 00:35:54,800
configured the right way there's a bunch

925
00:35:52,480 --> 00:35:58,240
of setup that goes into making those

926
00:35:54,800 --> 00:36:00,640
things work um we just have it all

927
00:35:58,240 --> 00:36:03,119

properly pre-configured in the container

928
00:36:00,640 --> 00:36:05,599
and now you can run some examples and so

929
00:36:03,119 --> 00:36:07,280
in this case i want to run first

930
00:36:05,599 --> 00:36:10,000
i'm going to take rec scale i'm going to

931
00:36:07,280 --> 00:36:11,119
convert it from json to yaml

932
00:36:10,000 --> 00:36:13,520
um

933
00:36:11,119 --> 00:36:16,560
what i've done here is we take our uh

934
00:36:13,520 --> 00:36:19,440
OSCAL whoops

935
00:36:16,560 --> 00:36:22,160
our OSCAL SSP example so this is a

936
00:36:19,440 --> 00:36:24,640
actual SSP that i exported out of RgScale

937
00:36:22,160 --> 00:36:26,560
as a json file and i'm going to

938
00:36:24,640 --> 00:36:29,119
convert it to yaml

939
00:36:26,560 --> 00:36:31,440
going to run for a second

940
00:36:29,119 --> 00:36:34,440
and now if i go back to visual studio

941
00:36:31,440 --> 00:36:34,440
code

942
00:36:41,200 --> 00:36:47,599
come over here and see here's my example

943
00:36:43,599 --> 00:36:51,760
ssp so this is a cmmc level two so it's

944
00:36:47,599 --> 00:36:54,640

got 110 controls from nist 800 171

945
00:36:51,760 --> 00:36:56,640
all in moscow when i ran that command it

946
00:36:54,640 --> 00:37:00,960
outputted this yaml file

947
00:36:56,640 --> 00:37:00,960
so in this case you can see the uh

948
00:37:02,720 --> 00:37:08,640
threat matter for the ssp all came over

949
00:37:05,599 --> 00:37:10,400
all our metadata around version history

950
00:37:08,640 --> 00:37:12,560
you get to the front matter for the

951
00:37:10,400 --> 00:37:14,480
implementation then you get into your

952
00:37:12,560 --> 00:37:16,640
controls and it pulls in all the same

953
00:37:14,480 --> 00:37:18,160
data that was in the json but dumps it

954
00:37:16,640 --> 00:37:20,240
out

955
00:37:18,160 --> 00:37:22,000
to a yaml file

956
00:37:20,240 --> 00:37:24,320
the next thing and last thing i'll show

957
00:37:22,000 --> 00:37:26,720
you here in the can presentation is

958
00:37:24,320 --> 00:37:28,800
we're going to do the same thing um but

959
00:37:26,720 --> 00:37:30,640
instead of converting to yaml we are

960
00:37:28,800 --> 00:37:32,480
going to convert it

961
00:37:30,640 --> 00:37:33,920

um to xml

962
00:37:32,480 --> 00:37:37,760
so in this case i'm going to run reg

963
00:37:33,920 --> 00:37:40,960
scale oscal convert catalog json to xml

964
00:37:37,760 --> 00:37:44,160
give it the json file same ssp

965
00:37:40,960 --> 00:37:44,160
output that as xml

966
00:37:44,640 --> 00:37:46,880
run

967
00:37:47,200 --> 00:37:53,040
fires up saxon in the converter when i

968
00:37:50,320 --> 00:37:55,200
come back i can look at the xml

969
00:37:53,040 --> 00:37:56,640
and here's my xml file with the same

970
00:37:55,200 --> 00:37:59,280
data in it

971
00:37:56,640 --> 00:38:00,960
again this is a work in progress on this

972
00:37:59,280 --> 00:38:02,880
part and so

973
00:38:00,960 --> 00:38:04,560
it is technically working with a few

974
00:38:02,880 --> 00:38:07,599
bugs here and there and so we've got to

975
00:38:04,560 --> 00:38:10,240
iron out the last mile bugs and the beta

976
00:38:07,599 --> 00:38:12,320
and get the documentation better um but

977
00:38:10,240 --> 00:38:14,640
all of that is now up and running and

978
00:38:12,320 --> 00:38:15,680

commands to sort of do the basics are

979
00:38:14,640 --> 00:38:19,760
here

980
00:38:15,680 --> 00:38:21,200
um last slide for me is um i'm the one

981
00:38:19,760 --> 00:38:22,480
giving the presentation but i'm

982
00:38:21,200 --> 00:38:25,119
definitely not the one who did all the

983
00:38:22,480 --> 00:38:28,000
work and so juliette easily came to us

984
00:38:25,119 --> 00:38:30,880
from the university of tennessee um data

985
00:38:28,000 --> 00:38:33,760
scientist um has now become a full stack

986
00:38:30,880 --> 00:38:35,920
developer a lot of the OSCAL export

987
00:38:33,760 --> 00:38:38,079
stuff single button click you saw was

988
00:38:35,920 --> 00:38:39,839
her work and so you can learn more about

989
00:38:38,079 --> 00:38:42,880
Juliet and her linkedin

990
00:38:39,839 --> 00:38:44,560
Super talented young developer and then

991
00:38:42,880 --> 00:38:47,839
we were fortunate to just get Brian

992
00:38:44,560 --> 00:38:50,079
eaton who took my happy Python code and

993
00:38:47,839 --> 00:38:52,720
brought it to another level and so brian

994
00:38:50,079 --> 00:38:55,760
was doing sort of large-scale work at

995
00:38:52,720 --> 00:38:58,640

oak ridge national lab with python

996
00:38:55,760 --> 00:39:02,160
he's our senior data engineer and brian

997
00:38:58,640 --> 00:39:05,040
has continued to build out the OSCAL CLI

998
00:39:02,160 --> 00:39:06,640
um uh and make it more robust all those

999
00:39:05,040 --> 00:39:08,240
tools are now out there in the community

1000
00:39:06,640 --> 00:39:09,920
for free to use

1001
00:39:08,240 --> 00:39:12,079
and if you want to learn more you can

1002
00:39:09,920 --> 00:39:14,400
email me anytime there's a bunch of

1003
00:39:12,079 --> 00:39:16,160
handy links below to our website

1004
00:39:14,400 --> 00:39:17,839
some blogs describing how this stuff

1005
00:39:16,160 --> 00:39:19,680
works or docs

1006
00:39:17,839 --> 00:39:21,119
where you pip install it where you can

1007
00:39:19,680 --> 00:39:23,599
pull the container

1008
00:39:21,119 --> 00:39:24,960
for the cli are all out there on the web

1009
00:39:23,599 --> 00:39:27,280
so with that

1010
00:39:24,960 --> 00:39:30,240
i thought i'd leave 10-15 minutes at the

1011
00:39:27,280 --> 00:39:34,280
end for any questions comments

1012
00:39:30,240 --> 00:39:34,280

from from the community