**ITCR Training and Outreach Working Group Meeting**

**Thursday, October 11, 2018**

**4:00 - 5:00 PM Eastern Time**

**Agenda**

1. Ideas/suggestions for future meetings
2. Roundtable discussion: Security implications for open source software

*Background*:

Medical schools are increasingly focusing on security, with potentially new restrictions coming down the line on what software will be allowed on any machine running on a secure network. This meeting will try to address how security groups are considering the use of open source software by research teams, and what developers will need to consider for the continued development of their software in terms of addressing security concerns, especially as it relates to potential adoption/installation obstacles at other sites.

The discussion will be led by Brett Miller.

*Brief Bio****:***

Brett Miller joined the University of Michigan in 2002. He is currently the Security Strategist for Michigan Medicine, where he concentrates on secure software development and advanced threats. He has also been active in security incident response, penetration testing, risk assessment, secure architectures, and policy/procedure development, among other areas.

Brett has had a varied career. His previous experience includes developing software for automation, business, collaboration, education, finance, health care, and networking. He is a past co-chair of the Association of American Medical Colleges Security Working Group.

He received his BS in electrical engineering from Purdue University, an MBA from Eastern Michigan University, and a PhD in Technology with a concentration in Information Assurance from Eastern Michigan University. His research interests include using machine learning for the detection of attacks and the discovery of vulnerabilities. He holds a CISM certification.

*Questions for Brett Miller (feel free to add your own):*

* How important is it these days to make sure that software is continually kept up to date with the most recent software packages that open source software might rely on?
* Do you foresee a time when research software will have to be ‘approved’ before being installed on any machine?
* What kind of security threats might exist on a network even if the software itself does not deal with protected health information?
  + I see that increased security measures would be needed soon to protect against potential threats including (but not limited to) data poisoning (intentional, or not), model backdoors, training data extraction from shared deep learning models, and membership attacks. Especially since efforts of sharing models is becoming more prominent. Do you see a framework on this respect that could be adopted by tools that are potential “threatened”?
* What kind of assurances will security teams require when assessing the safety of software that a research team wants to install. For example, will any proof of security reviews be required?
* Open-source research software is likely to be unsupported or abandoned over time. However, institutions continue to run old software, particularly for existing pipelines in longitudinal studies. Do you recommend particular tools (container as sandbox environment, SELinux-type MLS, etc) for managing the increasing risk over time of using unsupported, unreviewed software?
* How to respond to questions regarding OAG report about vulnerabilities in Pentagon weapon systems? (I’m concerned this is going to have fallout like “FOSS is unsafe!”)

|  |
| --- |
| **Webex Information:**  JOIN WEBEX MEETING  <https://cbiit.webex.com/cbiit/j.php?MTID=m90b62cf4b0ffc9b0c9e13b53425cc602>  Meeting number (access code): 733 094 302    JOIN BY PHONE  1-650-479-3207 Call-in toll number (US/Canada)    Global call-in numbers:  <https://cbiit.webex.com/cbiit/globalcallin.php?serviceType=MC&ED=574640482&tollFree=0> |

**Prior Meeting Notes:** [NCIP Hub](https://nciphub.org/groups/itcr/training_and_outreach_working_group_teleconference)

**Feedback about meeting or suggestions for future meetings:** [Anonymous Qualtrics Survey](https://umichumhs.qualtrics.com/jfe/form/SV_cIkVstbpgnbs4xT)

**Outreach Activities Since Last Meeting** (please fill in with new rows)

|  |  |  |
| --- | --- | --- |
| Name | Tool | Description of Activity |
| David Hanauer | EMERSE | Software demo for MSKCC  Software demo for U of Minnesota  Software demo for the CorVita Foundation |
| Joseph Perl | TOPAS | Multiple presentations at Radiation Research Society annual meeting.  Online Introductory Conferences for New Users (given 6 times in last 2 weeks for total of approx 60 new users) |
| Mary Goldman | UCSC Xena | CBIIT Speaker  Poster at TCGA legacy conference |
| Bradley Broom | NGCHM | Poster and interactive demo of NGCHMs at TCGA legacy conference |
| Michael Reich | GenePattern Notebook | 1-day workshop, UCSD  1-day workshop, NCI, Bethesda  Poster, [CSBC/PSON](https://csbconsortium.org/) annual meeting |
| Christos Davatzikos | CaPTk | Software Demo/Presentation and poster at the Tutorial on Tools Allowing Clinical Translation of Image COmputing ALgorithms (TACTICAL), in conjunction with the Medical Image Computing and Computer Assisted Interventions (MICCAI) conference |
| Dan Marcus | XNAT | Software Demo and poster at the Tutorial on Tools Allowing Clinical Translation of Image Computing ALgorithms (TACTICAL), in conjunction with the Medical Image Computing and Computer Assisted Interventions (MICCAI) conference |
| Stephen Jett | ITCR & IDC | Software Presentation and poster at the Tutorial on Tools Allowing Clinical Translation of Image Computing ALgorithms (TACTICAL), in conjunction with the Medical Image Computing and Computer Assisted Interventions (MICCAI) conference |
| Ken Chang (Jayashree Kalpathy Cramer) | DeepNeuro | Software Presentation and poster at the Tutorial on Tools Allowing Clinical Translation of Image Computing ALgorithms (TACTICAL), in conjunction with the Medical Image Computing and Computer Assisted Interventions (MICCAI) conference |
| Tahsin Kurc (Joel Saltz) | QuIP | Software Presentation and poster at the Tutorial on Tools Allowing Clinical Translation of Image Computing ALgorithms (TACTICAL), in conjunction with the Medical Image Computing and Computer Assisted Interventions (MICCAI) conference |
| Nate Braman (Anant Madabhushi, Anne Martel, Metin Gurcan) | PathIIP | Software Presentation at the Tutorial on Tools Allowing Clinical Translation of Image Computing ALgorithms (TACTICAL), in conjunction with the Medical Image Computing and Computer Assisted Interventions (MICCAI) conference |
| Spyridon Bakas | ITCR-funded tools | Organization of the Tutorial on Tools Allowing Clinical Translation of Image Computing ALgorithms (TACTICAL), in conjunction with the Medical Image Computing and Computer Assisted Interventions (MICCAI) conference. |

**Attendees, please sign in here:**

David Hanauer, U of Michigan

Mervi Heiskanen, NCI

Michael Ryan, JHU / MD Anderson

Chris Wakefield, MD Anderson

Joseph Perl, Stanford (TOPAS collaboration)

Bradley Broom, MD Anderson

Mary Goldman, UC Santa Cruz

Tali Mazor, DFCI

Mark Bergman, UPenn

Sarthak Pati, UPenn

Michael Reich, UCSD

Brett Miller, U of Michigan

Spyridon Bakas, UPenn

Hayley Dingerdissen, GW

**Minutes (everyone feel free to contribute):**

ITCR Annual Meeting, May 28-31, 2019. Salt Lake City/Park City, Utah

Incomplete notes (by David Hanauer):

Brett Miller:

Has spent a lot of time in software developement.

He focuses a lot on software itself.

Likes to look at software from the inside out (code itself).

But also looks from outside in (how could people break the application).

When they do a review they engage with the developers no the project, gets an overall idea for the architecture on the project. Gets a copy of the software. Will use as many automatic scanning tools (static application testing). Looks for constructs likely to be vulnerable. Runs scanners that look for "entropy" (information density). Can find passwords and secret keys that maybe shouldn't be in the source at al.

They also try to look at all the way they can get in the software (the attack surface). How can they compromise the software? In parallel they are using tools to defeat authentication.

They write up a summary of their findings, rank the things that should be addressed (low, moderate, highs) and they try to address the high and moderate issues.

It is very resource intensive.

So they look at the application itself to prioritize PHI containing data.

But even for apps that have public data, but if someone could attack that application it could also also make other systems vulnerable.

This is part of an overall review process. If things have met all the requirements it should be ready for production.

U of Michigan does lots of development (research and pure IT development).

But often doesn't matter if it is research information or clinical data, if the information gets out it's still a problem.

For all system, people must do their due-diligence to make sure they are not broken into.

"Supply chain risk". Open source software can be good, but it can also contain vulnerabilities. Some trusted open source software has been compromised, code has been back-doored. And software can be used in thousands of places. Too difficult to vet every single line of code.

Picking up updates immediately is good. Sometimes a new vulnerability can be attacked within hours. But if software is compromised, that can be a problem too (a double-edged sword).

(Bradely): They create next generation clustered heat maps, and they would like to send them to other users via email attachments. It is a standard zip file, and inside they give the name .bin since they are binary objects, and their email filter will remove anything that contains a .bin file.

They would also look to share a copy of their viewer, and it is an HTML file with embedded javascript.

So they have a clash between trying to make it easy to distribute software and security people filtering things out.

They do have to do reviews. it is a very long list of questions. There is a long list of checks to get it up and running. It seems to be a one-size fits all. They are standing up a web site tool, so checking for SQL injection, etc.

If the server is sitting in an isolated area of the network it wouldn't do damage if compromised. That could decrease burden a bit.

They just move it to a cloud server so they don't have to deal with institutional security.

(Brett): Even if it has moved to the cloud, network pipes back to the main site can also cause issues. May also need BAAs if it has regulated data.

Also a concern about datasets getting modified on the cloud, making for invalid data. So could use a checksum, like SHA-256, but need to make sure checksum is not in same place

(Brett): Most email inspection software. If it is an encrypted zip it may not allow it. Other systems will let a zip file go through if it cannot open the zip file.

But it is also possible someone could insert bad code before it gets there.

They often share files by Box, which can be "painful"

Joseph from Stanford:

They have a tool where users can write their own extensions. They have wanted to create a repository, because of concern about liability where users can put things, because what if someone puts something malicious there. The concern has limited them in what they can share.

They have had a thought if this is an area where maybe NCI could help to provide a trusted place to share things. The government may have some type of indemnification against those things.

(Brett): Some groups do continuous integration where code is built daily and you can pull packages in near-real-time. One way to avoid picking up a trivial update and getting a security issue. If something is just adding a minor enhancement, may want to hold off, but may accelerating something that has a security patch. Consider a risk vs reward tradeoff.

Places are going for application "white-listing" which means it may have to be vetted.

Some places are saying that software can't be downloaded to a server, but must be downloaded to desktop first.