

Towards Socio-Technical Topology-Aware Adaptive Threat Detection in Software Supply Chains

Thomas Welsh, Kristófer Finnsson,
Brynjólfur Stefánsson, Helmut Neukirchen

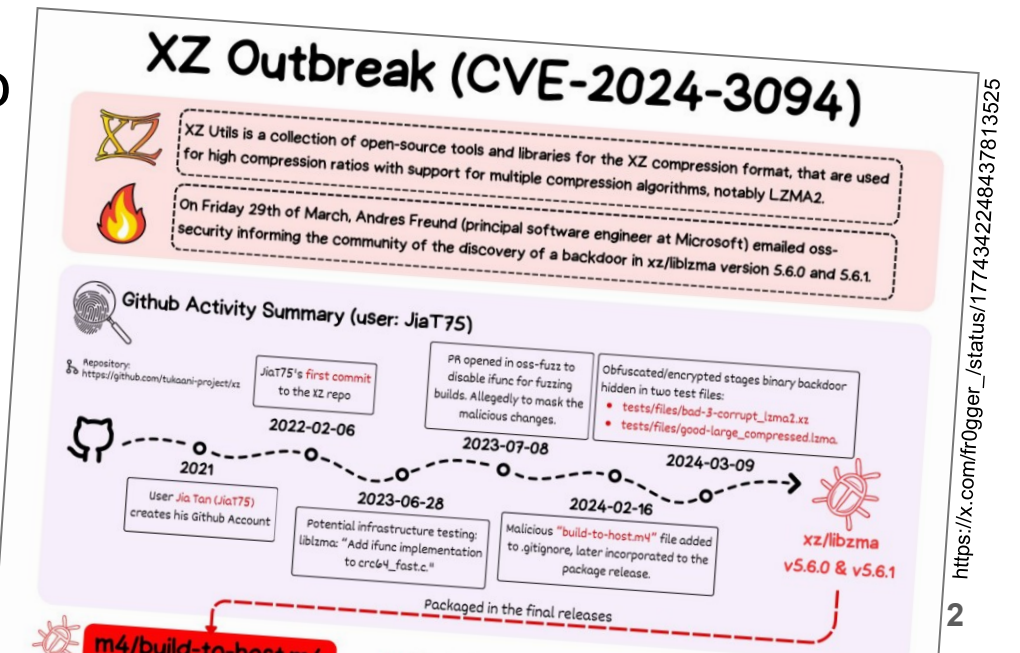
Prof. Dr. Helmut Neukirchen
Department of Computer Science
helmut@hi.is · <https://uni.hi.is/helmut>

**FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL
ENGINEERING AND COMPUTER SCIENCE**

Motivation:

The XZ Utils software supply chain attack

- XZ Utils: free and open-source software for lossless data compression.
 - Relevant software supply chain:
 - XZ Utils → XZ Utils used by **systemd** library → Linux distributions patch **OpenSSH** to use systemd notification mechanism.
 - Backdoor added in XZ Utils affecting OpenSSH:
 - A specific private SSH key could be used to **execute code remotely on any Linux machine running an OpenSSH server.**
- ⇒ Would allow to **take over large portion of the world's IT infrastructure.**



XZ Utils social engineering attack:

Details

Repository at: <https://github.com/tukaani-project/xz>

- Social engineering attack, long running from 2021-'24:
 - Original maintainer: [Lasse Collin \(username: Larhzu\)](#),
 - Malicious actors: [Jia Tan \(JiaT75\)](#) + several sock puppet accounts.
- **Legitimate Contributions (LC):** 2021-2022
 - Small, useful contributions to [establish credibility](#).
- **Escalation of Control (EC):** 2022-2024
 - Sock puppet accounts criticising Lasse Collin for slow progress, suggesting new maintainer.
 - [Lasse Collin agrees to make Jia Tan maintainer](#).
 - Technical groundwork led for the attack: changes to make later backdoor harder to detect.
- **Backdoor Deployment (BD):** Feb-Mar 2024
 - Jia Tan introduces well obfuscated backdoor.
- **Exposure and Removal (ER):** Mar 2024
 - Andres Freund notices SSH logins take 0.8s instead of 0.3s, finds backdoor and reports it.
 - [Backdoor gets removed again](#).



The XZ Utils software supply chain attack:

Could it have been detected using a socio-technical approach?

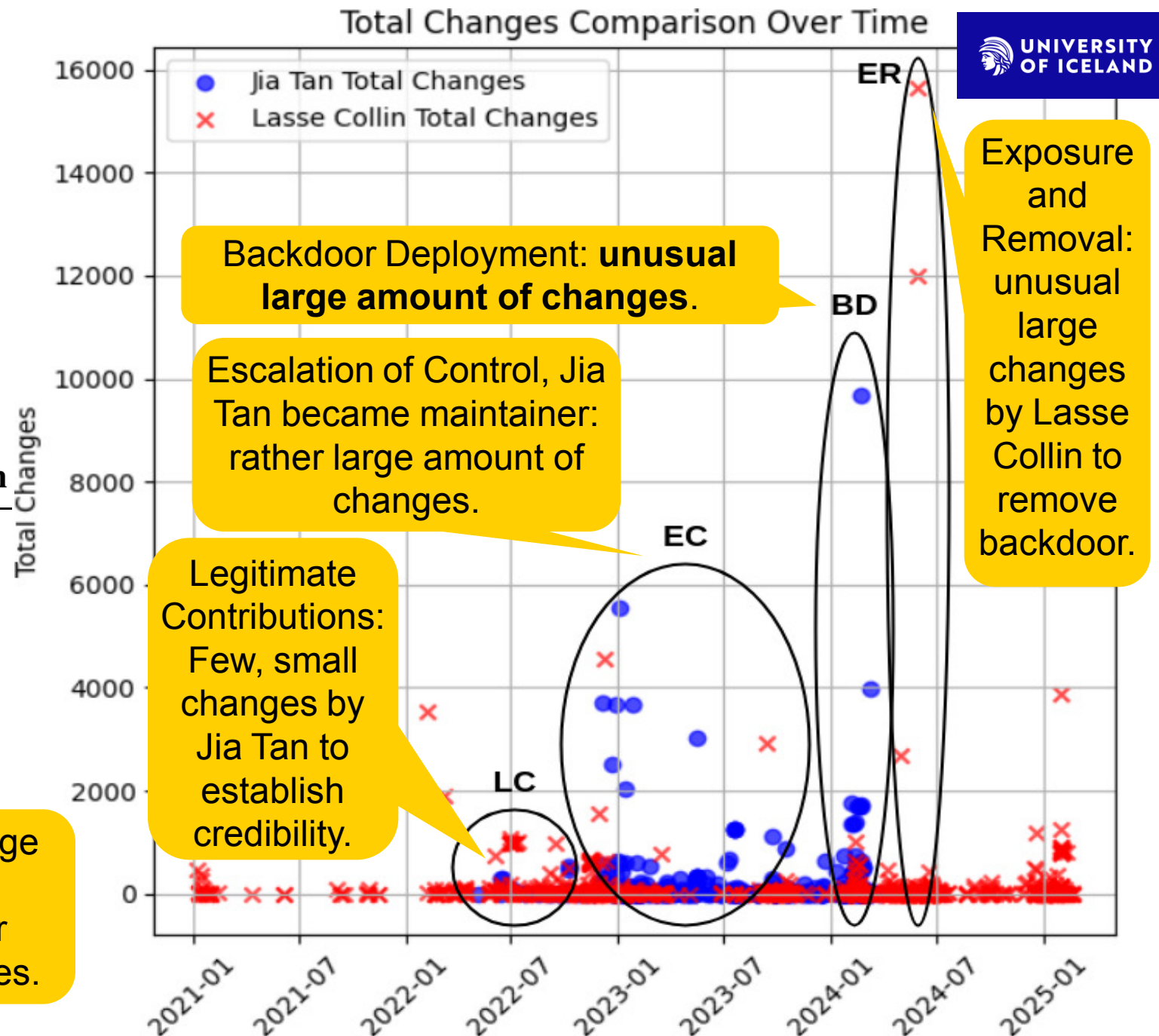
- Question: Can changes in developer behaviour be used to indicate software is trending towards an insecure state?
- Our position: automated monitoring of software repositories.
 - Use socio-technical threat indicators,
 - Create a topology out of these indicators to model the socio-technical relations,
 - As these are only indicators: vulnerability testing needed to confirm.
 - But: Vulnerability testing is resource-intensive (e.g. manual investigation).
 - ⇒ Use an adaptive approach to perform vulnerability testing only on relevant components.
- Example based on real XZ Utils data: see remaining slides.

Changes in files

- # changes per author and file:
 - “Change”=lines added/deleted
 - E.g. from `git log` command,
 - modified line counts as:
 - 1 deleted, 1 added.

Statistic	Jia Tan	Lasse Collin
Total Files Changed	697	1973
Average Additions	89.42	28.26
Average Deletions	42.10	18.31
Average Total Changes	131.53	46.56
Std Dev Additions	396.20	146.41
Std Dev Deletions	163.45	147.61
Std Dev Total Changes	492.14	249.01

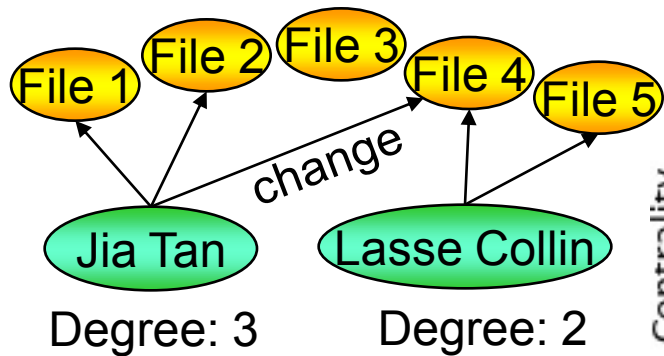
Malicious actor **Jia Tan**: Rather large average changes per file, but on fewer files.
Original maintainer **Lasse Collin**: Smaller changes in average per file, but on many files.



Centrality of “author-changing-file” graph

■ Graph:

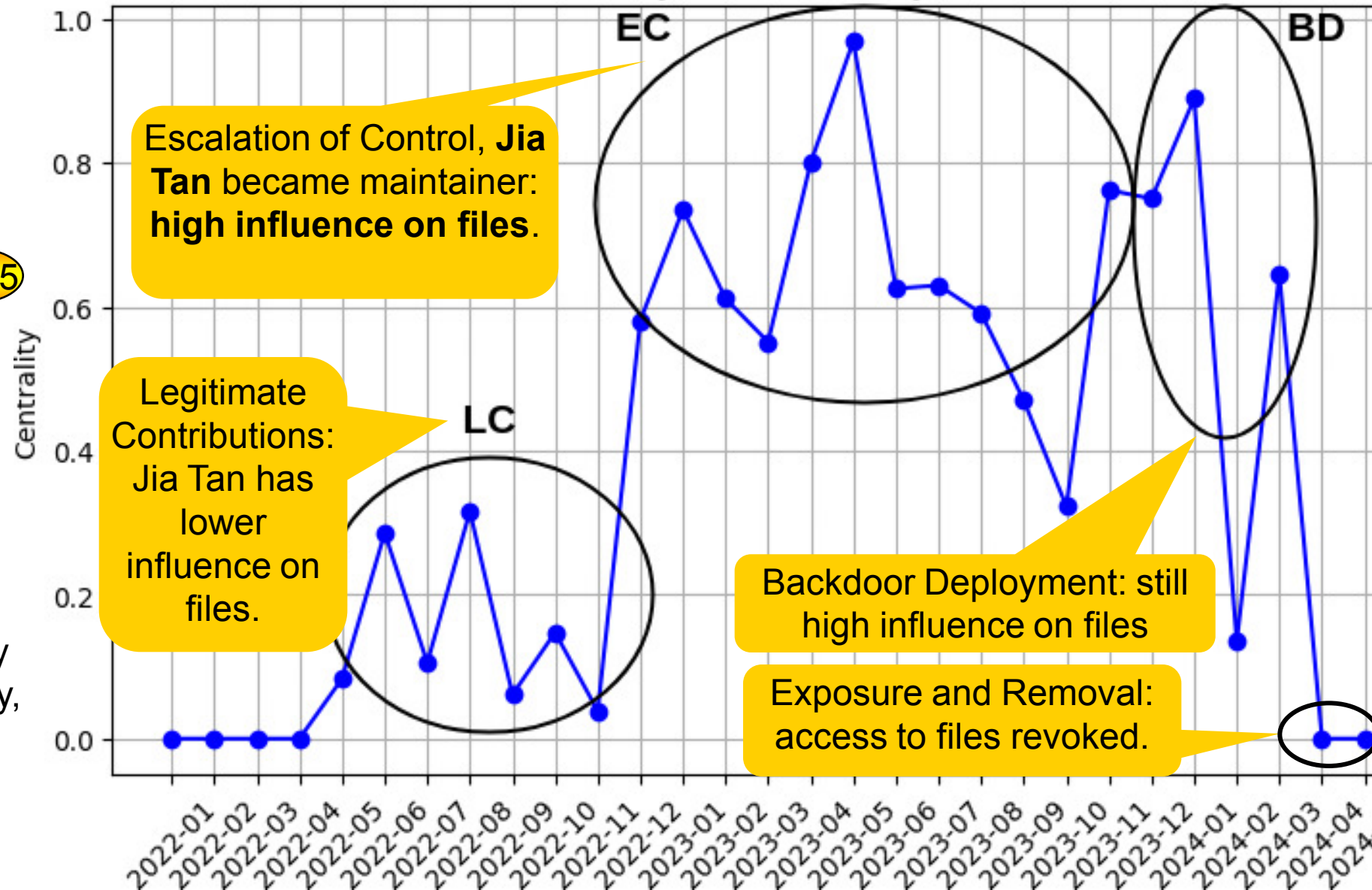
- Author changes a file:
⇒ edge from author node to file node.



■ Degree centrality:

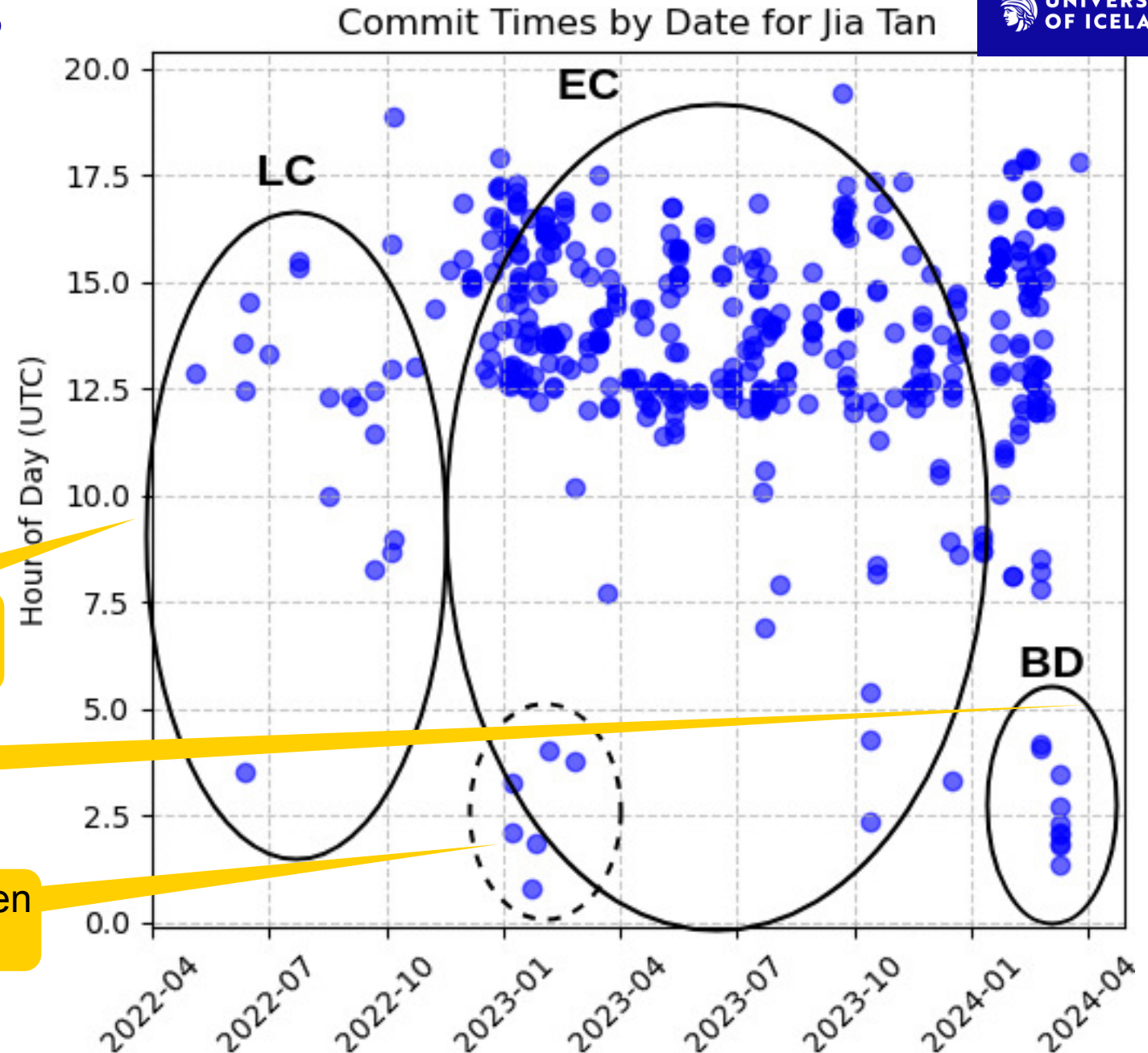
- # of edges of a node.
- Author changing many files has high centrality, i.e. high influence on repository.

Centrality Over Time for Jia Tan



Time of day of commits

- Atypical times of changes:
 - Jia Tan might be even two persons in different time zones or with different sleep patterns.



Most changes:
7:00-18:00 UTC

**Backdoor Deployment at
atypical time 1:00-4:00 UTC**

Other example of atypical times of changes, even
in parallel to typical times of changes.

- Communication**

from
her.
s:
ed as
:
:

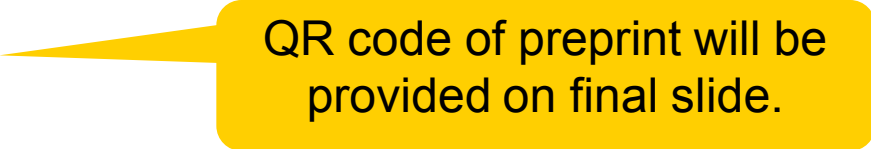
Jia Tan

Very positive sentiment in communication with sock puppet accounts.

Lasse Collin

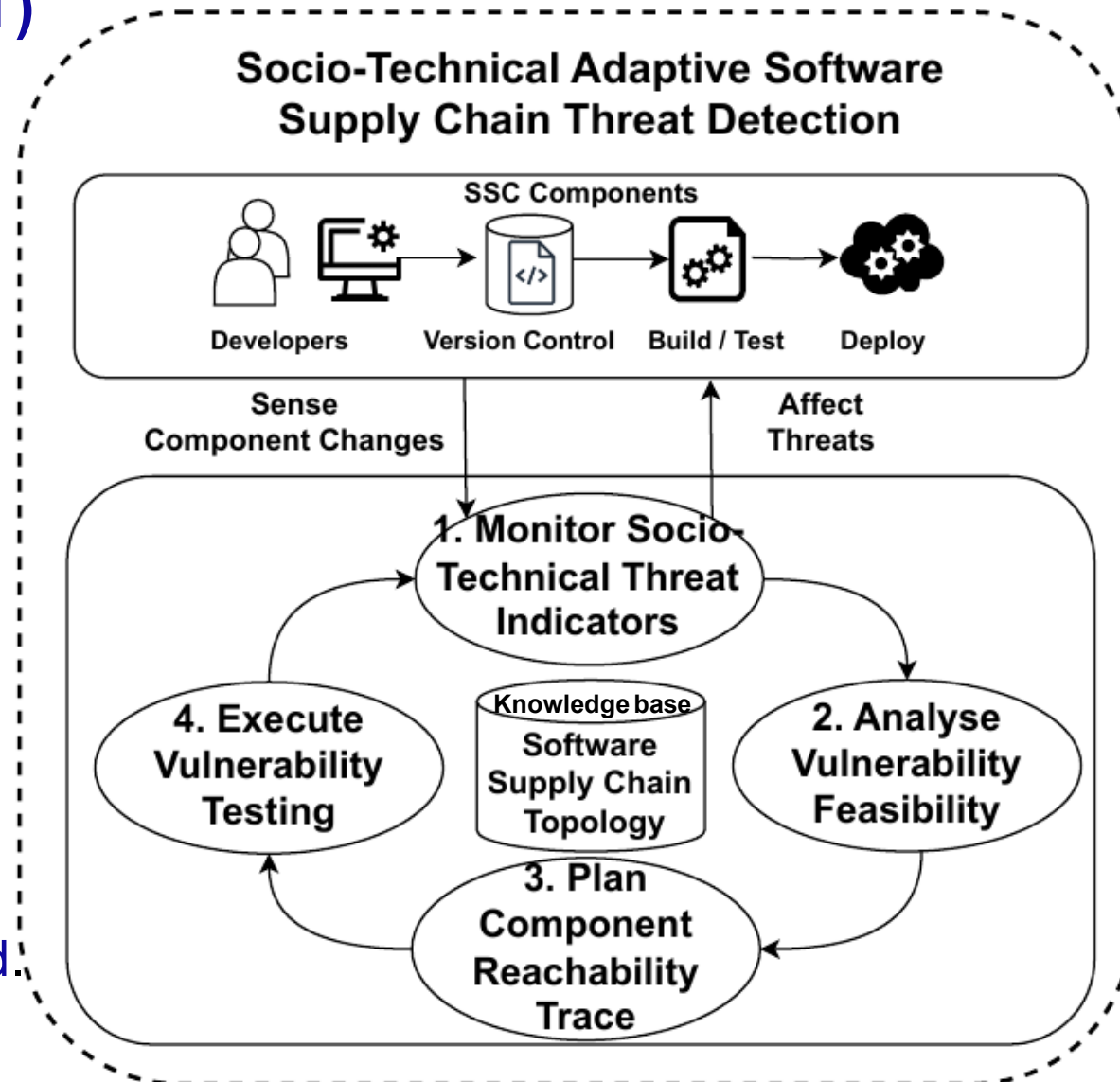
Not as positive sentiment as in communication with other (sock puppet) authors.

Technical Topology-Aware Adaptive Threat Detection in Software Supply Chains

- Looking at one indicator alone is not sufficient:
 - Would lead to false positives in case of legitimate change of maintainer.
 - Need to be combined.
- Build a structural model that represents the topology of the **socio-technical relations** of the respective software project.
 - See paper for details.  QR code of preprint will be provided on final slide.
 - If changes of indicators over time violate some threshold:
 - ⇒ Further analysis needed.

Adaptive Threat Detection based on Socio-Technical Topology (STT)

- As the STT contains only indicators:
vulnerability testing needed to confirm threat.
⇒ Do resource-intensive testing only where needed:
MAPE-K framework for **adaptive** software.
Quin, Weyns, Gheibi: "Decentralized self-adaptive systems: A mapping study,"
Int. Symposium Softw. Eng. Adaptive Self-Managing Systems (SEAMS). IEEE, 2021.
- Monitor:** Mine software repositories, filter for suspect components based on topology changes exceeding a threshold.
- Analyse:** use on filtered components **automated tools for static software security analysis** (i.e. without executing SW=less resource intensive).
- Plan:** for possible vulnerabilities found via static analysis: **calculate which other software along the software supply chain (SSC) would be affected**.
- Execute:** do **vulnerability testing on all affected software along supply chain** (e.g. dynamic testing).



- **Technical approaches** to software supply chain security exist, e.g.:
 - Software Bill of Materials (SBOM) to keep track of dependencies and then monitor Common Vulnerabilities and Exposures (CVEs) and report them for along the supply chain.
 - But works only retroactively *after* a CVE has been published.
- **Existing work does not account for social factors.**
 - Hope: suggested **socio-technical approach** can **detect attacks *pro-actively*!**

- Limitations:
 - Validation of approach.
 - XZ Utils attack is one of the few known attacks that we can use that to tune thresholds.
 - Would need data from more attacks, but: such data is sparse.
 - What are good socio-technical indicators?
 - Running our approach on *all* relevant software where source code is available:
 - Despite adaptive approach, would need a lot of resources.
- Position paper:
 - Implemented: gathering basic socio-technical indicators from GitHub repos.
 - Collected data from XZ Utils repository: <https://github.com/tukaani-project/xz>
 - Built part of the Socio-Technical Topology and its visualisation.
 - Adaptive MAPE-K loop still needs to be implemented!

Thank you for your attention · any questions?



**UNIVERSITY
OF ICELAND**



Thomas Welsh · Kristófer Finnsson · Brynjólfur Stefánsson
Helmut Neukirchen · helmut@hi.is · <https://uni.hi.is/helmut>

Preprint DOI