Techniques to Identify Bugs in Authorization

Frank Capobianco, Giuseppe Petracca, Nirupama Talele, Christian Skalka, Gang Tan, and Trent Jaeger

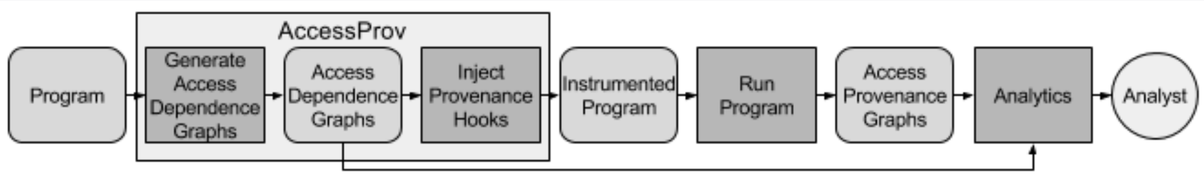
Overview

- Access control and authorization mechanisms are implemented **manually** in practice.
  - Lack of knowledge, negligence, or malicious intent can lead to bugs and vulnerabilities (bypass, backdoors, etc.).
- Correctness of access control enforcement depends on runtime factors, such as the access control policy and adversary controlled inputs.
- Combination of **static and dynamic analysis** are necessary to vet access control and authorization mechanisms within programs.

Problems

- Security sensitive operations may be performed **without authorization** entirely.
- Multiple distinct operations my be authorized with the **same permission set**.
- Authorization may **dominate multiple operations**.
- Data relevant to authorization may be **tampered with before, during, or after authorization** possibly altering how authorization is performed.

Detecting Access Control Errors



- Given program and set of known program input locations.
- Generate **access dependence graphs**.
  - Generate program dependence graph to capture all information flows.
  - Leverage taint analysis to **identify security sensitive operations**.
- **Inject provenance hooks** for each security sensitive operation.
- Generate **access provenance graphs** using additional runtime data.
- Use statically generated access dependence graphs to identify matching access provenance graphs in runtime data and present them to an analyst.

Detecting Data Tampering

- Given program and known locations of user credentials (username, password).
- Leverage taint analysis to **identify candidate authorization** code within program by intersecting taint labels.
- Generate **program chops** to understand the relationship of user input to authorization and security sensitive operations.
  - Pre-authorization chops.
  - Intra-authorization chops.
  - Post-authorization chops.
- Analyze computed chops for **instruction sequences that dictate data tampering**.
- Present tampering instructions to analyst for further investigation.

Evaluation

- Evaluated provenance tracking technique on **OpenMRS's test suite**.
  - Found 29 cases where authorization was not present.
  - Found a single case where permissions were not consistent with similar authorization elsewhere in the program.
  - Found a case where a single authorization hook dominated several security sensitive operations, where additional permissions should have been checked.
- Hook injection only induced a 2.1% performance overhead when running the test suite.

Partial Authorization:
- Subject "Admin123" is authorized to getUser() from database.
- Uses object to get login credentials of User without additional authorization.
- Gathers secret question from users credentials.

Consistency:
- Subject "Admin123" is authorized to perform two distinct operations to "edit" and "delete" a person from database.
- Same permission set is used for both operations, which is inconsistent to similar operations elsewhere in the program.

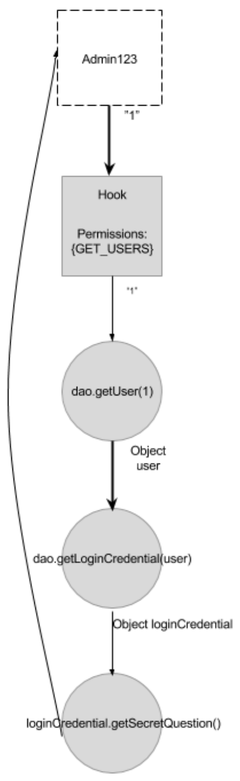


Figure 1. Partial Authorization

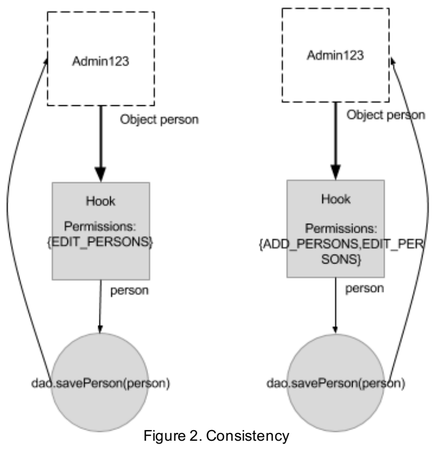


Figure 2. Consistency

```
v2 = a1 + 56;
v3 = *(int **)(a1 + 56);
*v3 = v4 ^ *((_DWORD *)&v92 + (((v4 - 1882412364) & (unsigned int)(v4 - 1882412363)) >>
31) - 24);
…
if ( sub_1A770(v3, v6) ) // Performs authentication for modified username
{
  ...
}
…
v7 = 1; // Set return value for authentication (1 = success)
return v7;
```

"KUp3"

- Analysis of maliciously **modified version of vsftpd**.
  - Taint analysis identified 2 functions that perform authorization.
  - Generated intra-authorization programs identified **7 LLVM instructions** corresponding to a single **source line of code** related to data tampering.
  - Bit manipulation instructions check whether the first 4 characters of the username match "**KU3p**". If they do, the username is changed to "**root**".

Publications

- Capobianco, F., Skalka, C., & Jaeger, T. (2017). AccessProv: Tracking the Provenance of Access Control Decisions. In *Proceedings of the 9th USENIX Conference on Theory and Practice of Provenance.* USENIX Association. (In Submission)
- Muthukumaran, D., Talele, N., Jaeger, T., & Tan, G. (2015, March). Producing Hook Placements to Enforce Expected Access Control Policies. In *International Symposium on Engineering Secure Software and Systems* (pp. 178-195). Springer International Publishing.
- Petracca, G., Capobianco, F., Skalka, C., & Jaeger, T. (2017). On Risk in Access Control Enforcement. *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies.* ACM.

Supported by National Science Foundation, DARPA, and Lincoln Lab