Role Modelse Role-based Debloating for Web Applications

Babak AminAzad, Nick Nikiforakis

What is software debloating?

- Software debloating consists of identifying and removing unnecessary code.
- Software packages and third-party dependencies are the main contributors to code-bloat.
- By removing unused code, we remove the vulnerabilities associated with that code.
- Through a user study, we modeled the behavior of users, and then removed unused modules from the source code.
- Our system named *DBLTR*, can shrink the size of web applications by **40-79%**.





In Step 1, we provide the debloated web applications and user-role mappings to DBLTR's content delivery module. User requests (Step 2) are processed by *DBLTR's* reverse-proxy (Step 3). After extracting the identity of the user (Steps 4-6), *DBLTR* internally routes the requests to custom debloated web applications (Step 7).

Design of *DBLTR*

- We conduct a user study with 60 web developers and administrators to understand how experienced users interact with web applications.
- We cluster users with similar behavior together under the same "Role" using source code features (e.g., usage of similar packages, files, functions, classes, etc.)
- We generate tailored web applications for each Role.
- *DBLTR* detects successful login requests and extracts authentication cookies.
- Finally, we transparently route user traffic to the designated web applications via a reverse-proxy.

Security benefits of DBLTR

DBLTR outperforms the state-of-the-art in web application debloating. Previous debloating schemes produce one debloated web application for all users.

Role-based debloating enables *DBLTR* to: - Remove **30% more lines of code** from the source code of web applications. - Produce applications exposed to 80% fewer

- historic CVEs.
- removed features.





Computer Science

- By clustering users with similar behavior together, we reduce the likelihood of users running into