WebSheets: A New Privacy-Centric Framework for Web Applications

Scott D. Stoller
Stony Brook University
Stony Brook, New York, USA
stoller@cs.stonybrook.edu

ABSTRACT

Spreadsheets are enormously popular because they enable non-programmers to create applications that manipulate tabular data. The core functionality of many web applications is to display and manipulate tabular data, typically stored in databases. These observations inspired the design of WebSheets, a no-code/low-code web application development framework that provides novel support for security and privacy. The key innovation of WebSheets is that fine-grained, data-driven security policies, as well as application logic, are expressed in the spreadsheet paradigm. This empowers data owners, who are often non-programmers, to directly implement their desired security policies.

Each *data table* in WebSheets is paired with a *permission table*, which is editable only by the data table's owner. Formulas in a permission table define who can read and write cells in the associated data table. These formulas can easily express role-based, attribute-based and relationship-based access control policies as well as delegation. WebSheets guarantees that these policies are enforced during the entire lifetime of every data item, as it flows through calculations within an application and even when it is passed between applications. While providing global privacy guarantees similar to information flow control systems, WebSheets enables end users to work with the more familiar access control policies.

Any user wishing to safeguard their data should store them in tables they own, thereby requiring all web applications to access their data by referencing their tables. This ensures that all applications will respect their access policies in the associated permission tables. By automatically filtering out inaccessible rows and columns, WebSheets presents user-customized views that are the key feature of many web applications.

Additional key features of WebSheets include: ***secure and scalable distributed evaluation techniques*** that confine WebSheets computations using OS-based access control and sandboxing mechanisms to enforce the principle of least privilege; ***secure integration*** with external systems, including web servers, databases, web browsers, user interfaces, and external modules. The benefits of distributed, least-privilege evaluation extend to modules written in any language; ***policy analysis***, including novel techniques to help users understand policies and debug policy errors, and to improve policies over time, either to correct problems or respond to changes in use; and ***expressive formula language*** that features first-class tables, seamless integration of access control and input validation, and support for declassification.

Web application vulnerabilities have been the dominant cause of data breaches in recent years. As defenses against lower-level vulnerabilities have come to be widely deployed, attackers are targeting higher-level errors. WebSheets addresses the following three common types of higher-level errors.

Omitted or incorrectly coded security policies. Key stakeholders in data privacy are typically non-programmers that need to first communicate their security requirements to developers that then implement them. Developers may misunderstand the desired policies or implement simpler, relaxed policies as a result of pressure to deliver required functionality on time. In WebSheets, data owners can directly express desired fine-grained security policies using formulas.

Incorrect placement of security checks. Today, policies are enforced mainly by ad-hoc placement of security checks throughout a web application's code. This lack of separation of concerns makes it hard to check whether important security policies are correctly implemented and soundly enforced by complete mediation. In WebSheets, security policies are separated from other application logic and enforced automatically on all data paths.

Vulnerabilities that create unintended dataflows. Command and data injection vulnerabilities provide avenues for attackers to create new data flows, allowing data breaches to occur. The underlying problem is that web applications generally execute with a superset of the privileges available to all end users. In contrast, WebSheets by default executes with the privilege of the requesting user. Hence, data inaccessible to that user won't be leaked or corrupted, despite vulnerabilities in the application code or the WebSheets evaluation engine.

WebSheets is related to commercial no-code and low-code web application development frameworks for creating mobile apps and web apps centered around interacting with tabular data stored in databases or spreadsheets, such as Google AppSheet and Glide Apps, but they lack WebSheets's key features listed above.

This is joint work with R. Sekar. Preliminary work on WebSheets is described in [1, 2].

SACMAT '23, June 7–9, 2023, Trento, Italy

ACM ISBN 979-8-4007-0173-3/23/06.
https://doi.org/10.1145/3589608.3593816

CCS CONCEPTS

• **Security and privacy → Web application security**; *Access control*; *Information flow control*; *Usability in security and privacy*.

KEYWORDS

web application security, access control, information flow control, spreadsheets

ACM Reference Format:
Scott D. Stoller. 2023. WebSheets: A New Privacy-Centric Framework for Web Applications. In *Proceedings of the 28th ACM Symposium on Access Control Models and Technologies (SACMAT '23), June 7–9, 2023, Trento, Italy.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3589608.3593816

BIOGRAPHY

Scott D. Stoller is a Professor in the Computer Science Department at Stony Brook University. His primary research areas are computer security, cyber-physical systems, distributed systems, and programming languages. He received his Bachelor's degree in Physics, summa cum laude, from Princeton University and his Ph.D. degree in Computer Science from Cornell University. He received an NSF CAREER Award, an ONR Young Investigator Award, the NASA Turning Goals Into Reality Award for Engineering Innovation (as a member of the Java PathFinder team), and three Best Paper Awards. He is the author or co-author of over 140 refereed research publications and has been the PI or co-PI on over $24M of research grants.

ACKNOWLEDGMENTS

This work is supported in part by NSF award CNS-2153056.

REFERENCES

[1] Riccardo Pelizzi. 2016. *Securing Web Applications.* Ph. D. Dissertation. Stony Brook University, Stony Brook, NY, U.S.A.
[2] Riccardo Pelizzi and R Sekar. 2015. WebSheets: Web Applications for Non-Programmers. In *2015 New Security Paradigms Workshop (NSPW).* ACM, 137–147. https://doi.org/10.1145/2841113.2841124