Type-Specific Languages to Fight Injection Attacks
Darya Kurilova, Cyrus Omar, Ligia Nistor, Benjamin Chung, Alex Potanin, and Jonathan Aldrich

1 let webpage : HTML = parse_html(
2 "<html>
3  "<body>
4   "<h1>Results for " + keyword + "</h1>"
5   "<ul id="results">" + to_string(query_results(db, parse_sql(
6      "SELECT title, snippet FROM products WHERE title = " + keyword + "'")) ) + 
7   "</ul></body></html>"

Safe version (library-based):
1 let webpage : HTML = HTMLElement(Dict.empty(), [BodyElement(Dict.empty(),
2   [H1Element(Dict.empty(), [TextNode("Results for " + keyword)]),
3   ULElement(Dict.add(Dict.empty(),{"id", "results"}), query_results(db,
4   SelectStmt(["title", "snippet"], "products",
5   [WhereClause(Equals("title", StringLit(keyword)))])))]))

Cross-site scripting (XSS) attack: Provide "<script>malicious_code</script>"
SQL injection attack: Provide "'; DROP TABLE products --"
Type-Specific Languages to Fight Injection Attacks

Darya Kurilova, Cyrus Omar, Ligia Nistor, Benjamin Chung, Alex Potanin, and Jonathan Aldrich

• **Wyvern** – new web programming language

• **Wyvern approach**: *use a library that extends the language*
  – As convenient as strings
  – As safe as conventional libraries

• Mechanism: **Type-Specific Languages (TSLs)**
  – Associates parsing logic with types
  – Provides modular extensibility
Type-Specific Languages to Fight Injection Attacks

Darya Kurilova, Cyrus Omar, Ligia Nistor, Benjamin Chung, Alex Potanin, and Jonathan Aldrich

In Wyvern, developer can write:

1. \texttt{let keyword : String = user_input}
2. \texttt{let webpage : HTML = ~}
3. \texttt{"<html>\n    <body>\n      <h1>Results for \{HTML.TextNode(keyword)\}</h1>
      <ul id="results">\n        \{query_results(db, ~)\}
        SELECT title, snippet FROM products WHERE title = \{keyword\}\n      </ul>\n    </body>\n</html>"}

- **XSS is prevented!**
  - The expression is of HTML type
  - Otherwise compilation error

- **SQL injection attack is prevented!**
  - Prepared SQL statement

- Shifts responsibility to typechecker and TSL designer
- Succinct to write
- Reduces developer’s cognitive burden