UntrustIDE

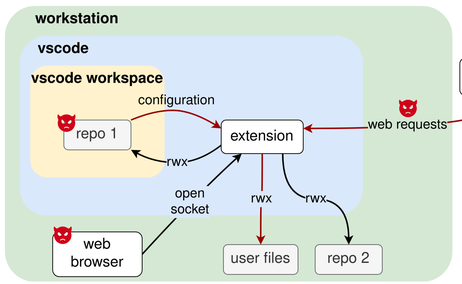
Exploiting Weaknesses in VS Code Extensions

Elizabeth Lin, Igibek Koishybayev,
Trevor Dunlap, William Enck,
and Alexandros Kapravelos
North Carolina State University

Visual Studio Code is the most popular IDE for developers. It's marketplace offers more than 50 thousand extensions that allow users to add functionality to the IDE.
In this study, we explore vulnerabilities in VS Code extensions through taint analysis.

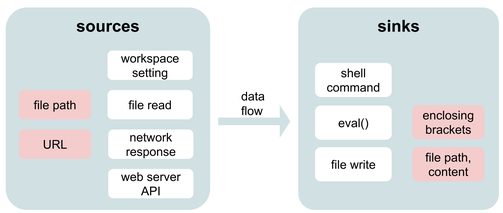
Threat Model

In our threat model, VS Code users and VS Code extension developers are trusted. The goal of the adversary is to achieve code execution.



Analysis

The goal of this study is to systematically identify code injection and file integrity vulnerabilities in VS Code extensions. We build CodeQL queries to identify dangerous data flows that could lead to attacks.




```
from
Configuration cfg, DataFlow::PathNode source,
    DataFlow::PathNode sink, StringOps::
    Concatenation concat_string
where
cfg.hasFlowPath(source, sink) and concat_string =
    sink.getNode() and
( concat_string.getFirstLeaf().toString() = "\"(\""
    or concat_string.getFirstLeaf().toString() =
    "'('" )
and
( concat_string.getLastLeaf().toString() = "\")\""
    or concat_string.getLastLeaf().toString() = "')
    '" )
```

NC STATE UNIVERSITY

Do you use VS Code?

We show how **VS Code extensions** could be exploited by a third party to achieve **code injection**.

Exploitation of a VS Code extension can be a single line of code in a git repo you download.

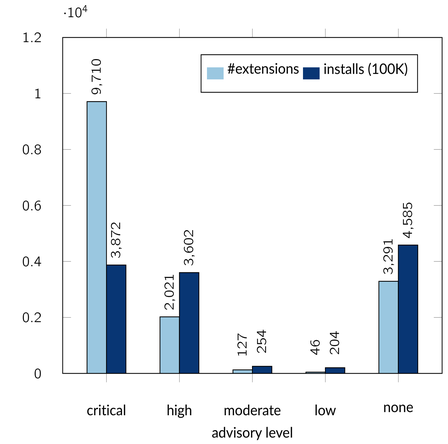
```
"git": { "path": "a-shell-script.sh" }
```

We verified vulnerabilities with proof of concepts for **21 extensions**, amounting to more than **6 million installations**.

Scan to access our repository with the queries at https://github.com/s3c2/UntrustIDE

Distinguished Paper Award @ NDSS Symposium

We show the impact of the Node.js ecosystem on VS Code extensions.
We discovered 13,655 VS Code extensions where each one has more than 100 npm transitive dependencies.
Furthermore, **9,710 extensions** depend on vulnerable npm packages with a **critical-level advisory.**



Our tool identified **716** dangerous data flows.
Data read from **VS Code workspace settings** and **files** are the most common source of code injection vulnerabilities in VS Code extensions.

Source	Sink	Number of extensions with calls to both the source and the sink	Filtered Flows	PoCs
workspace settings	shell	2213	389	7
	eval	192	12	6
	file write	1847	24	0
file read	shell	1718	75	4
	eval	397	34	4
	file write	2847	150	0
network response	shell	174	0	0
	eval	122	1	0
	file write	259	25	0
web server	shell	151	3	0
	eval	64	0	0
	file write	146	3	1

Disclosure

For extensions with verified PoCs, we notified extension developers of the vulnerabilities. We also contacted the GitHub security team of results of our research.

Acknowledgements

This material is based upon work supported by the National Science Foundation Grant No. 2207008. Any opinions expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

S3C2