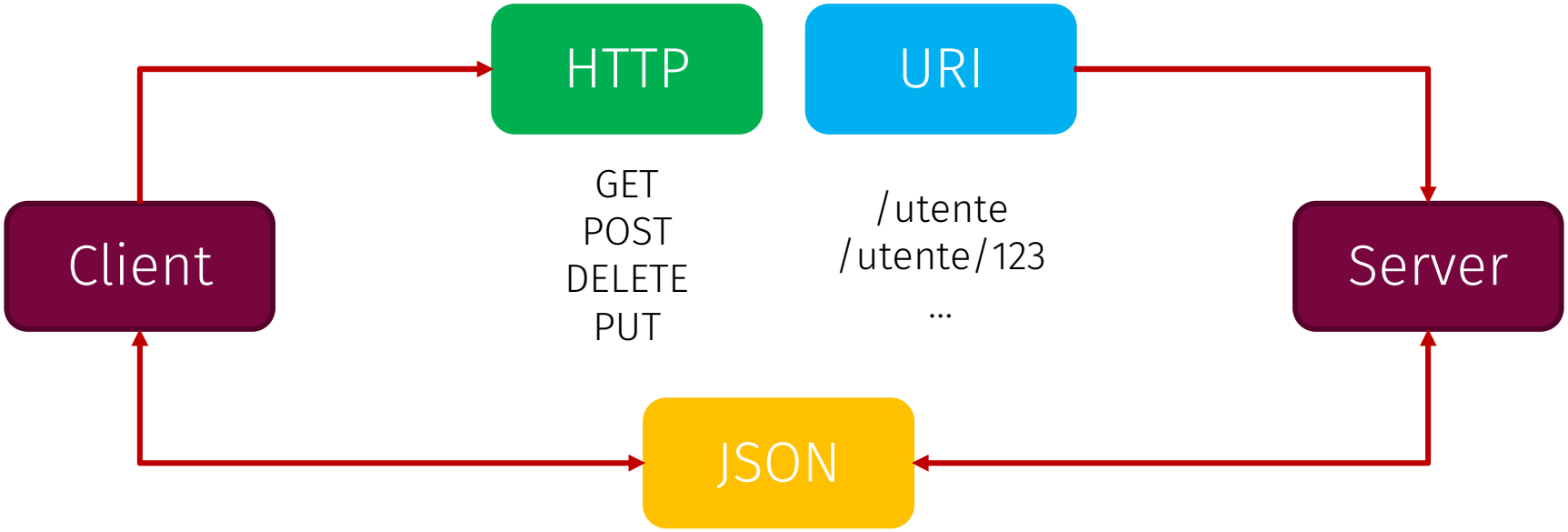
INIEZIONE AUTOMATIZZATA DI VULNERABILITÀ MASS ASSIGNMENT IN REST API

REST API




```
{
   "email": email,
   "password": password,
   "nome": nome,
   "cognome": cognome
}
```

Mass assignment

- Funzionalità di mappatura automatica dei parametri passati ad una REST API
- Nelle REST API, le proprietà di una risorsa possono essere leggibili (R) o anche scrivibili (W)
- A causa di errori di programmazione una proprietà leggibile può essere comunque modificata
- La vulnerabilità al mass assignment si manifesta quando una proprietà leggibile viene scritta o modificata con successo

UTENTE
email (W): *string*
password (W): *string*
admin (R): *boolean*

Problema affrontato

- Il problema affrontato nella tesi è lo sviluppo di uno strumento di iniezione di vulnerabilità al mass assignment nelle REST API

- 3 fasi:
  - Identificazione dei linguaggi di programmazione più popolari e dei framework più utilizzati
  - Implementazione del tool di iniezione
  - Validazione dello strumento prodotto

Motivazioni

Sostegno alla validazione sistematica svolta tramite tool di rilevamento automatico di mass assignment

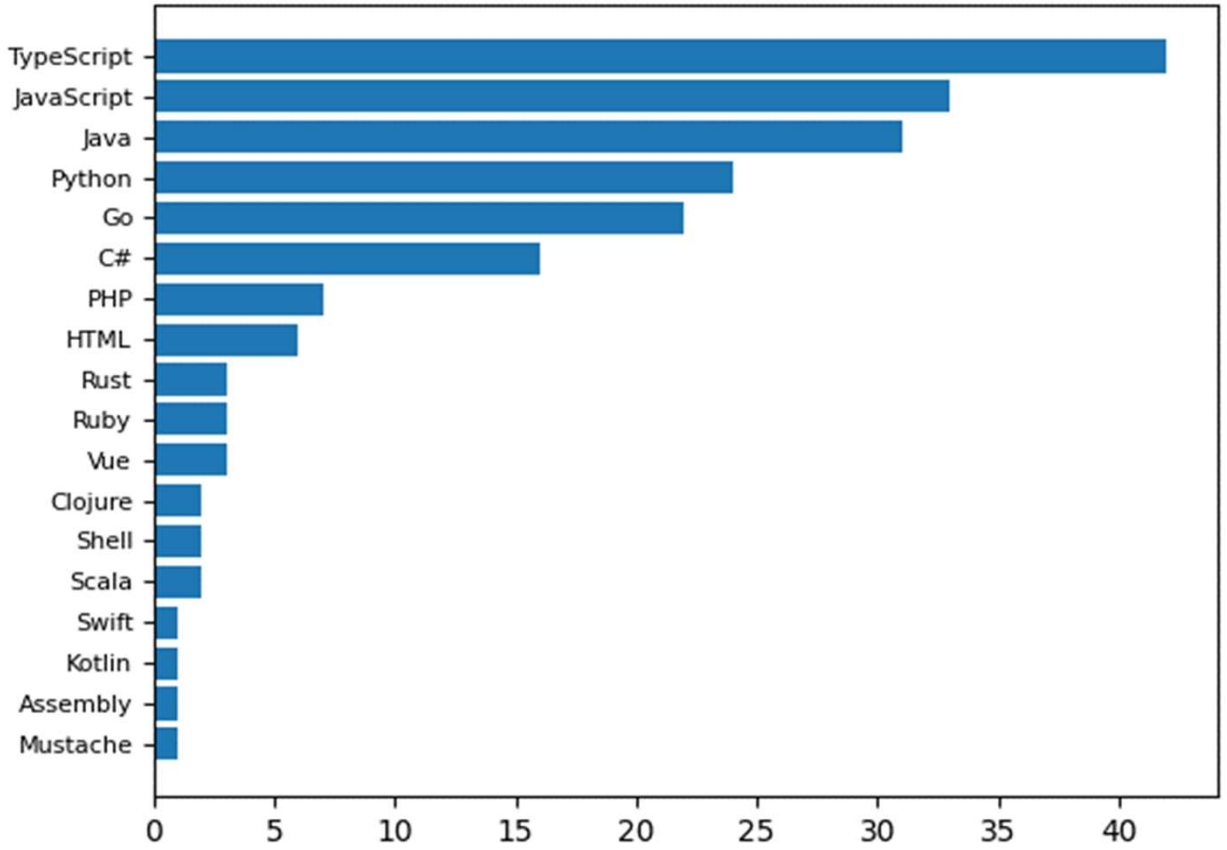
Garantire la conformità di un’applicazione rispetto alle pratiche di sicurezza da adottare per mass assignment

Migliorare la comprensione della vulnerabilità al mass assignment e contribuire alla ricerca nell’ambito della sicurezza delle REST API

Indagine sullo stato dell’arte delle REST API

1

Linguaggi di programmazione più popolari



Framework di programmazione più utilizzati

Linguaggio	Framework	Occorrenze
Typescript	nestjs	6
	tsed	1
JavaScript	express	8
	fastify	4
	hapi	4
	koa	2
Java	spring	9
	jaxrs	2
	resteasy	1
	jersey	1

Iniezione di vulnerabilità

2

Processo di iniezione automatizzata


```
...
@HttpCode(HttpStatus.OK)
@Post('signup') signUp(@Body() signUpDto: any ) {
    return this.authService.signUp(signUpDto);
}
...
```

Validazione sperimentale

3

Quesiti di ricerca

- **QR1:** Sono stati effettivamente iniettati tutti i punti potenzialmente vulnerabili?
- **QR2:** Il codice modificato è effettivamente vulnerabile a mass assignment?
- **QR3:** Il codice modificato rimane corretto da un punto di vista funzionale?

Setup sperimentale

- Scelti 9 casi di studio, 3 per ogni linguaggio e rispettivo framework di programmazione
- Definite 3 metriche:
  - Correttezza dell'identificazione dei pattern
  - Correttezza dell'iniezione di vulnerabilità
  - Validità delle API post iniezione
- Scelto un sottoinsieme di file per ogni caso di studio, analizzati prima e dopo l'esecuzione del tool insieme a stampe di debug preimpostate

Risultati

- Nei casi in cui erano presenti punti potenzialmente iniettabili, lo strumento ha identificato e modificato con successo la quasi totalità dei casi, 41 casi identificati su 45 potenziali

- La vulnerabilità di mass assignment è stata correttamente introdotta da tutte le iniezioni effettuate

- Tutte le API esaminate si sono rivelate funzionanti anche utilizzando il codice prodotto dallo strumento

Risposte ai quesiti di ricerca

- **Risposta a QR1:** Lo strumento mostra una buona correttezza nell'identificare i punti potenzialmente vulnerabili, con una percentuale totale di identificazione corretta del 91%

- **Risposta a QR2:** Lo strumento presenta un'ottima correttezza nell'iniezione di vulnerabilità al mass assignment, mostrando una percentuale del 100% nei casi utilizzati per la sperimentazione

- **Risposta a QR3:** Lo strumento non altera la validità del codice da un punto di vista funzionale, presentando una percentuale di validità del 100% per i codici analizzati nei casi di studio scelti

Considerazioni

- Lo strumento implementato fornisce un punto di partenza per trattare la vulnerabilità al mass assignment anche in altri linguaggi e framework di programmazione
- Può essere utilizzato per verificare la sicurezza di un'applicazione che implementa REST API contro attacchi via mass assignment
- Fornisce supporto alla validazione tramite tool di rilevamento automatico di vulnerabilità al mass assignment

Relatore

Prof. *Mariano Ceccato*

Correlatore

Dott. *Davide Corradini*

Contro-Relatore

Prof. *Michele Pasqua*

Laureando

Robert Octavian Timofte
VR471628

ANNO ACCADEMICO 2022/2023