Implementing Basic Web Application Security

Nick Russo

NETWORK ENGINEER

@nickrusso42518   www.njrusmc.net

Agenda

Common secret storage techniques

Understanding OAuth 2.0

Exploring common OWASP threats

Implementing CSRF support:

- CRM app
- System tests

Common Options for Handling Secrets

Interactive prompts

Environment variables

OS-specific keyrings

Encrypted vault files

How We Used to Do It

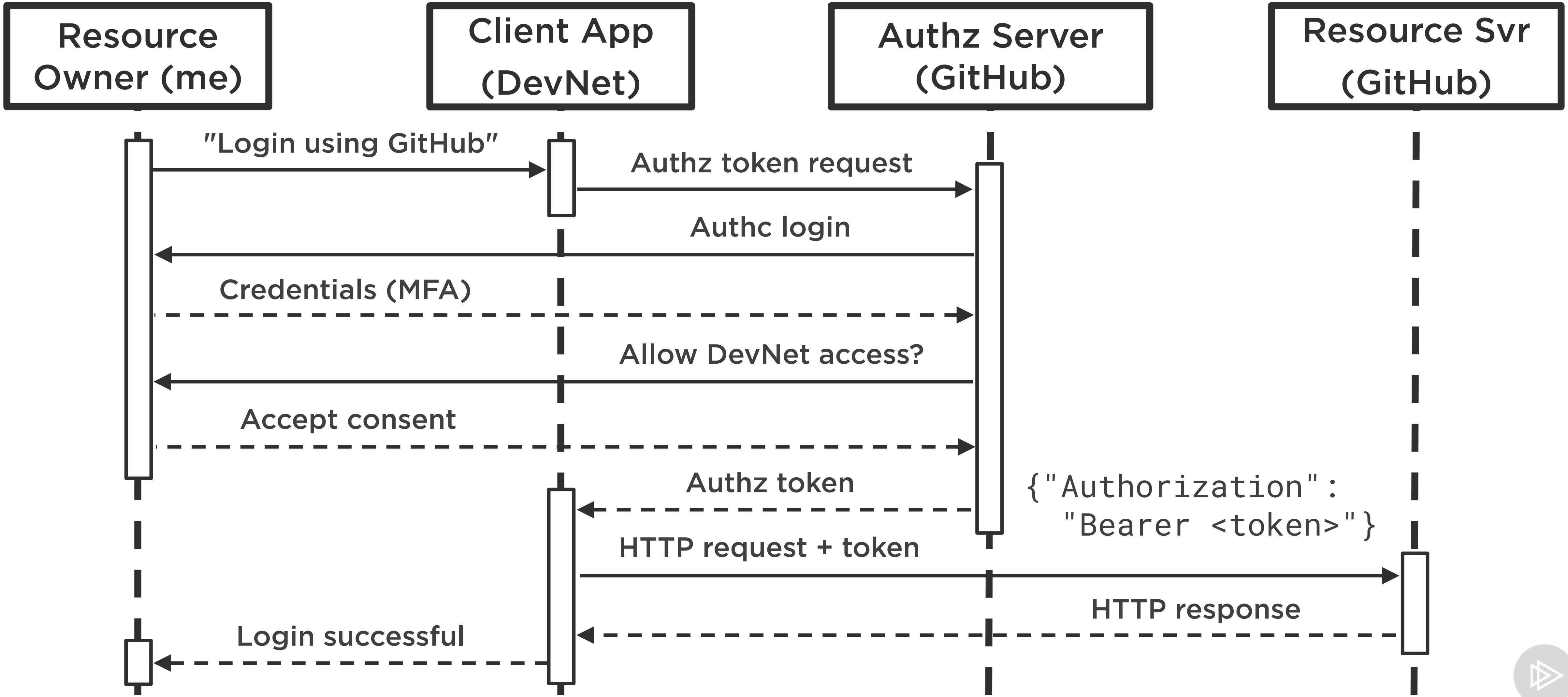
- Credential sharing
- Cookies
- API keys

Authentication vs. Authorization

Authentication
(authc)

Authorization
(authz)

OAuth 2.0 High-level Operation

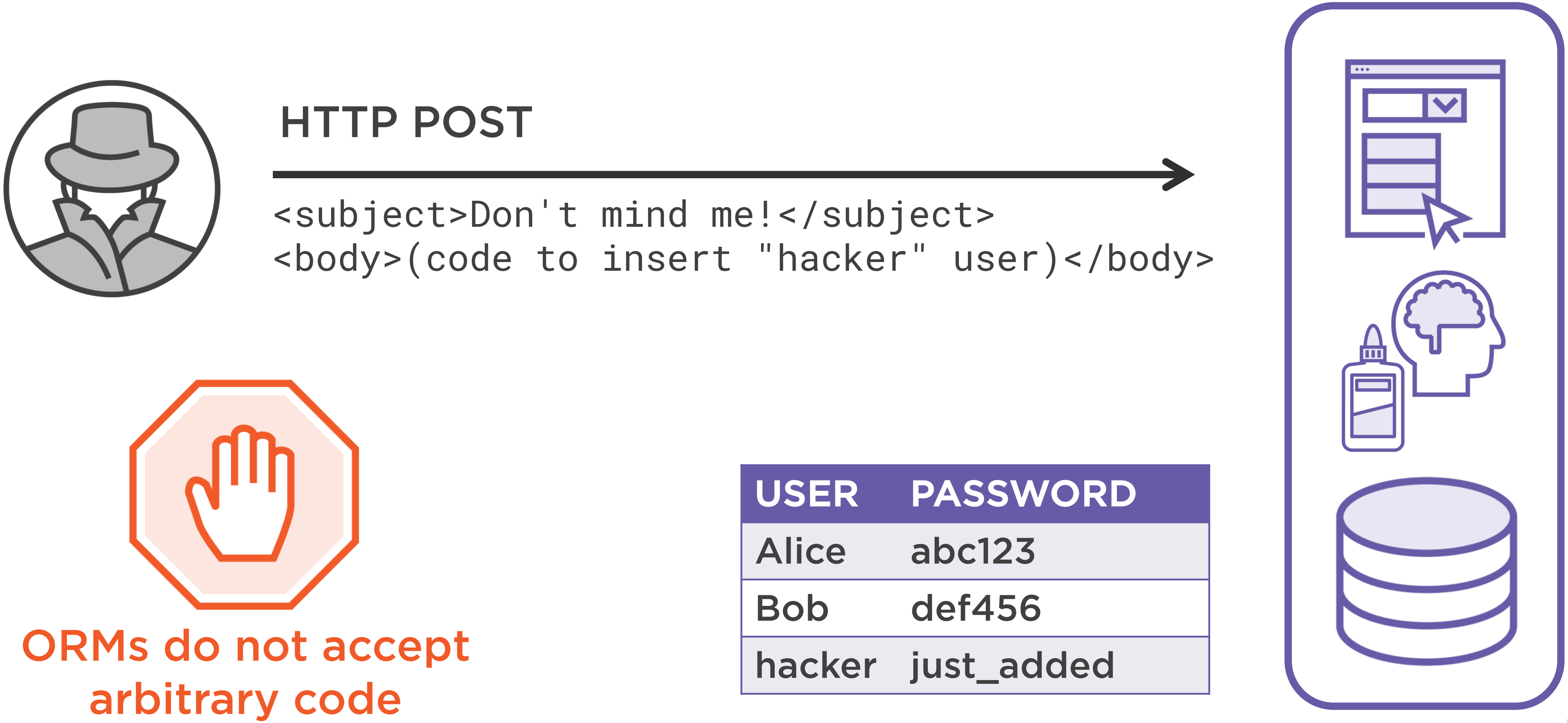


```
{"Authorization":
  "Bearer <token>"}
```

Demo

OAuth 2.0 in real life; Github and Devnet

Injection Attacks



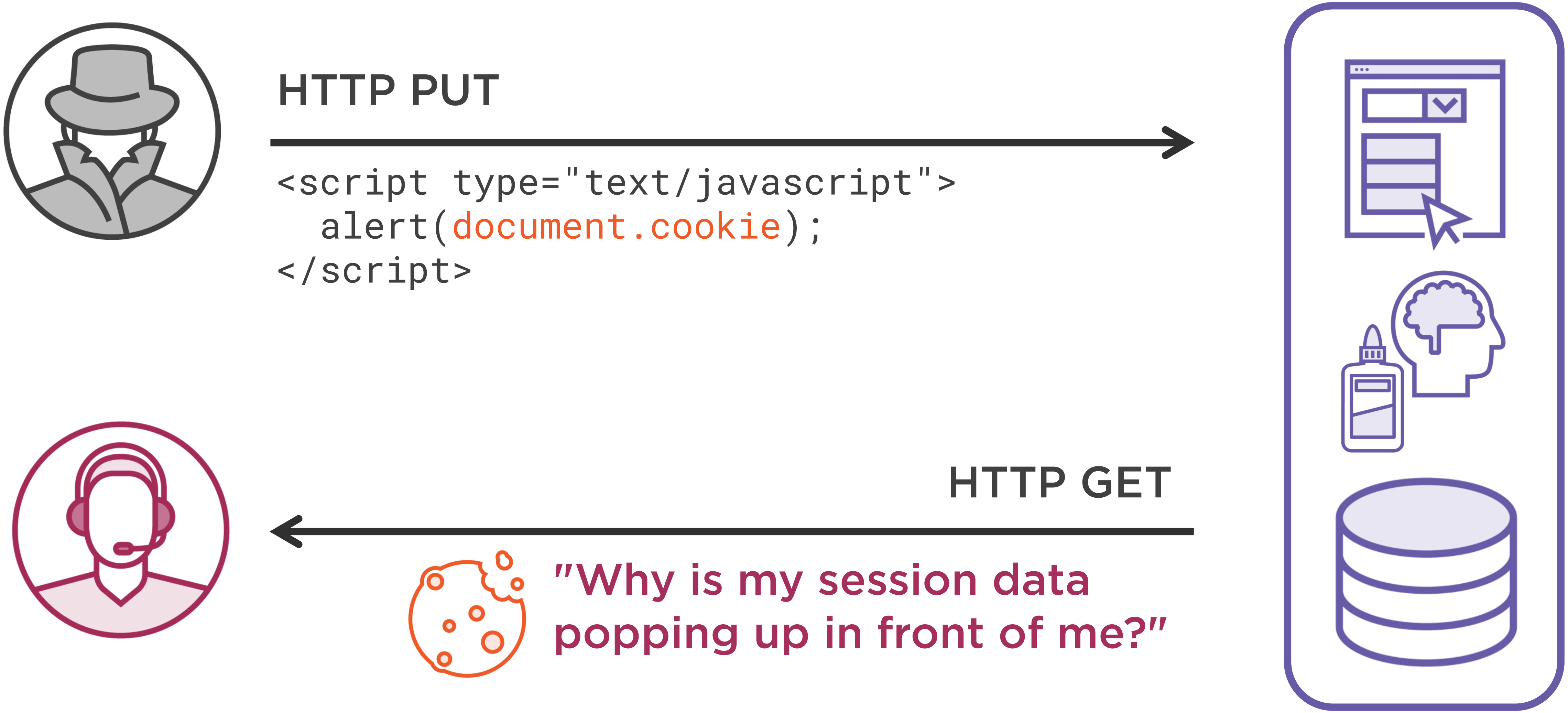
HTTP POST

```
<subject>Don't mind me!</subject>
<body>(code to insert "hacker" user)</body>
```

ORMs do not accept arbitrary code

USER	PASSWORD
Alice	abc123
Bob	def456
hacker	just_added

Cross-site Scripting (XSS)



HTTP PUT

```
<script type="text/javascript">
  alert(document.cookie);
</script>
```

HTTP GET

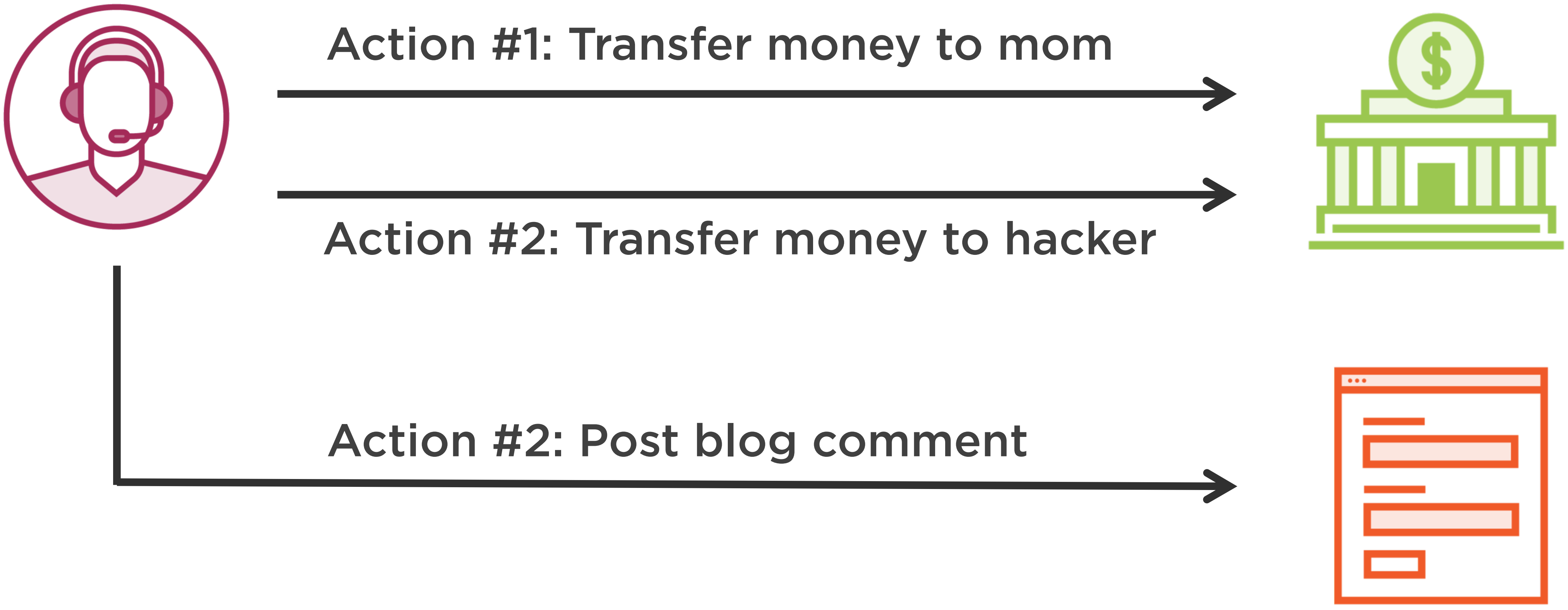
"Why is my session data popping up in front of me?"

Basic XSS Prevention

Escape, validate, and sanitize user input

Encode and identify output responses

Cross-site Request Forgery (CSRF)



Action #1: Transfer money to mom

Action #2: Transfer money to hacker

Action #2: Post blog comment

Demo

Implementing CSRF in the CRM app

Demo

Updating the CRM app system tests

Challenge: Update the System Tests!

requests.session()

pip install bs4

Summary

Secret management and OAuth 2.0

Common OWASP threats

Implementing real-life CSRF