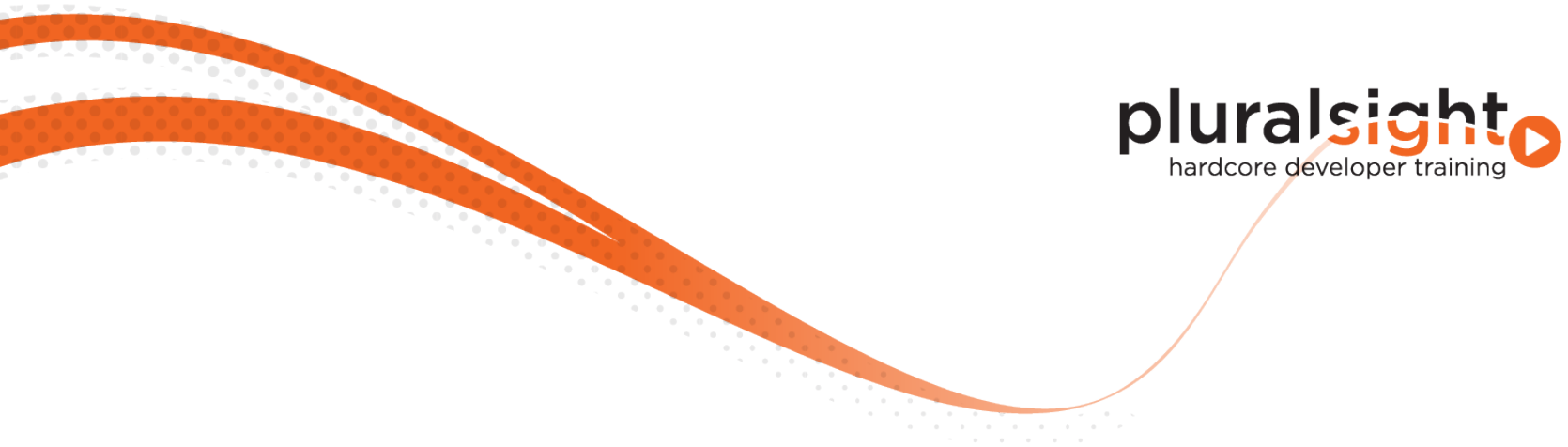
Injection

Troy Hunt
troyhunt.com

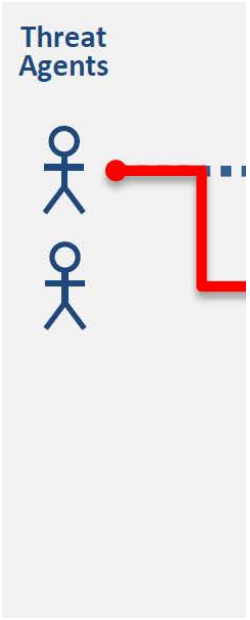


pluralsight
hardcore developer training

Outline

- How OWASP views the risk
- Performing an attack
- Understanding SQL injection and untrusted data
- Implementing the principle of least privilege in the database
- Parameterising inline SQL
- Using stored procedures to reduce injection risk
- Understanding and implementing a whitelist
- How ORMs protect against SQL injection
- Some remaining risks and attack tools

Understanding application security risks



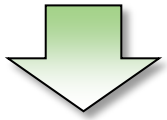
OWASP overview and risk rating

Threat Agents
—
Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.

Understanding SQL injection

Trusted

http://www.mysite.com/Widget?Id=1



```
SELECT * FROM Widget WHERE ID = 1
```

Untrusted

What constitutes untrusted data?

- **The integrity is not verifiable**
- **The intent may be malicious**
- **The data may include payloads such as:**
  - SQL injection
  - Cross site scripting
  - Binaries containing malware

Common sources of untrusted data

- **From the user**
  - In the URL via a query string or route
  - Posted via a form
- **From the browser**
  - In cookies
  - In the request headers
- **From any number of other locations**
  - External services
  - *Your own database!*

Implementing a whitelist

- **All untrusted data should *always* be validated against a whitelist of known good values**
- **A whitelist describes what data we know to be *safe***
  - It's a very *explicit* control
  - It strictly filters out anything we don't trust
- **A blacklist describes what data we know to be *dangerous***
  - It's a very *implicit* control
  - It assumes anything not on the list will always be safe

Approaches to whitelisting

- **Type conversion**
  - Integer, date, GUID, etc.
- **Use a regular expression**
  - Email address, phone number, name (but be careful)
- **List of known good values**
  - Countries, products, colours etc.

Summary

- **Apply the principle of least privilege to the database account**
  - What does it really need to be able to do?
- **Always parameterise untrusted data**
  - Never just concatenate query and data
- **Stored procedures also offer protection via parameterisation**
  - Just remember you can still build an injection risk into them
- **Always validate untrusted data against a whitelist**
  - Remember type conversion, regexes and known good values
- **Use ORMs and their native ability to parameterise**
  - They're also a great time saver
- **Remember the ease of exploitation by automated tools**
  - Injection consequences can be severe and be easily exploited