

# OWASP TOP10

Check (mate) to the vulnerabilities



Some reflections about OWASP TOP 10 and chess game

# Agenda

- Threats in software application security
- OWASP top 10
- Definition
- Example
- Solutions
- Chess comparisons
- OWASP in the software life cycle

# Threats in chess

“A threat is simply a move that **attacks** one of your pieces to get some **advantage** if you don't find an **accurate defense**”

**Sheperd check mate**



# Threats in chess

"The threat is stronger than the execution" Aaron Nimzowitch (Latvia 1886, chess grandmaster).

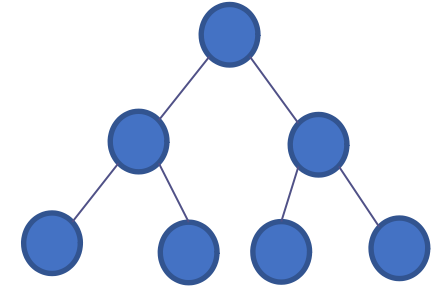
We need to keep the **tension** and do our best all the time to **avoid vulnerable points** in our system.

Every move should be done to avoid risky points.



# Threats in software

- Can we **guess/predict threats** for our app?
- As in a chess game we need to predict the next move of the **other player (hacker?)**. A move that can affect our software systems.
- **Threat modeling** is like brainstorming of possible moves a hacker can do against our system.
- There are different methodologies to follow: STRIDE, CAPEC, Attack Tree, **OWASP TOP 10 ...**



# OWASP TOP 10

Insufficient  
Logging

Injection

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

Sensitive data  
exposure

Insecure  
Deserialization

XML External Entity

Cross Site  
Scripting

Broken Access  
Control

Security  
Misconfiguration

# OWASP TOP 10

Injection

Insufficient  
Logging

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

Sensitive data  
exposure

Insecure  
Deserialization

XML External Entity

Cross Site  
Scripting

Broken Access  
Control

Security  
Misconfiguration

# Injection - Definition

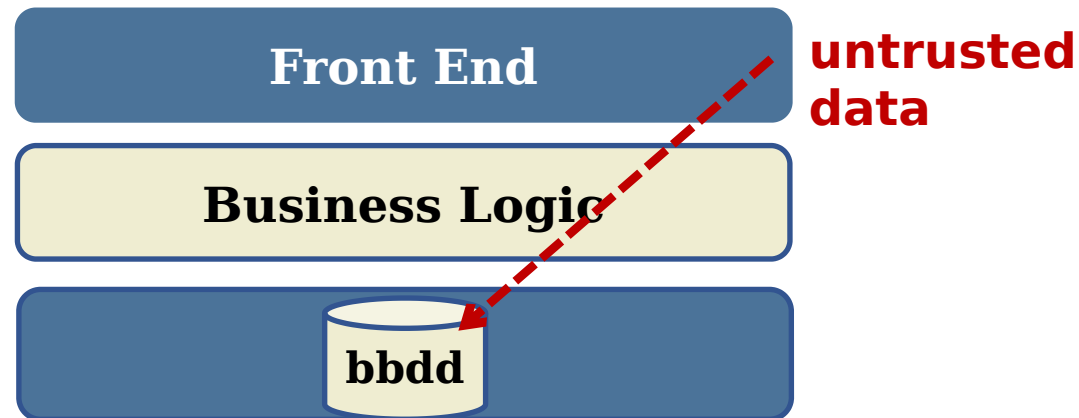
Injection happens when **untrusted data** is sent to an interpreter as part of a **command or query**. The attacker's hostile data can trick the interpreter into executing **unintended commands or accessing data** without proper authorization.

If the injection affects our database then we talk about **SQL Injection**.

If the injection affects our logs then we talk about **Log Injection**.

If the injection executes some commands of our system then we talk about **Command Injection**.

And there are more injection issues as LDAP injection, XPATH injection.



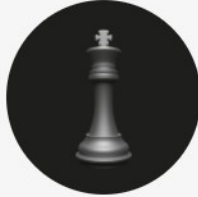


# Injection - demo

**Vulnerable Chess Game**

You can try SQL Injection in this page ;-)

According to the OWASP TOP TEN: it is risk #1, more info [here](#)



**Username**

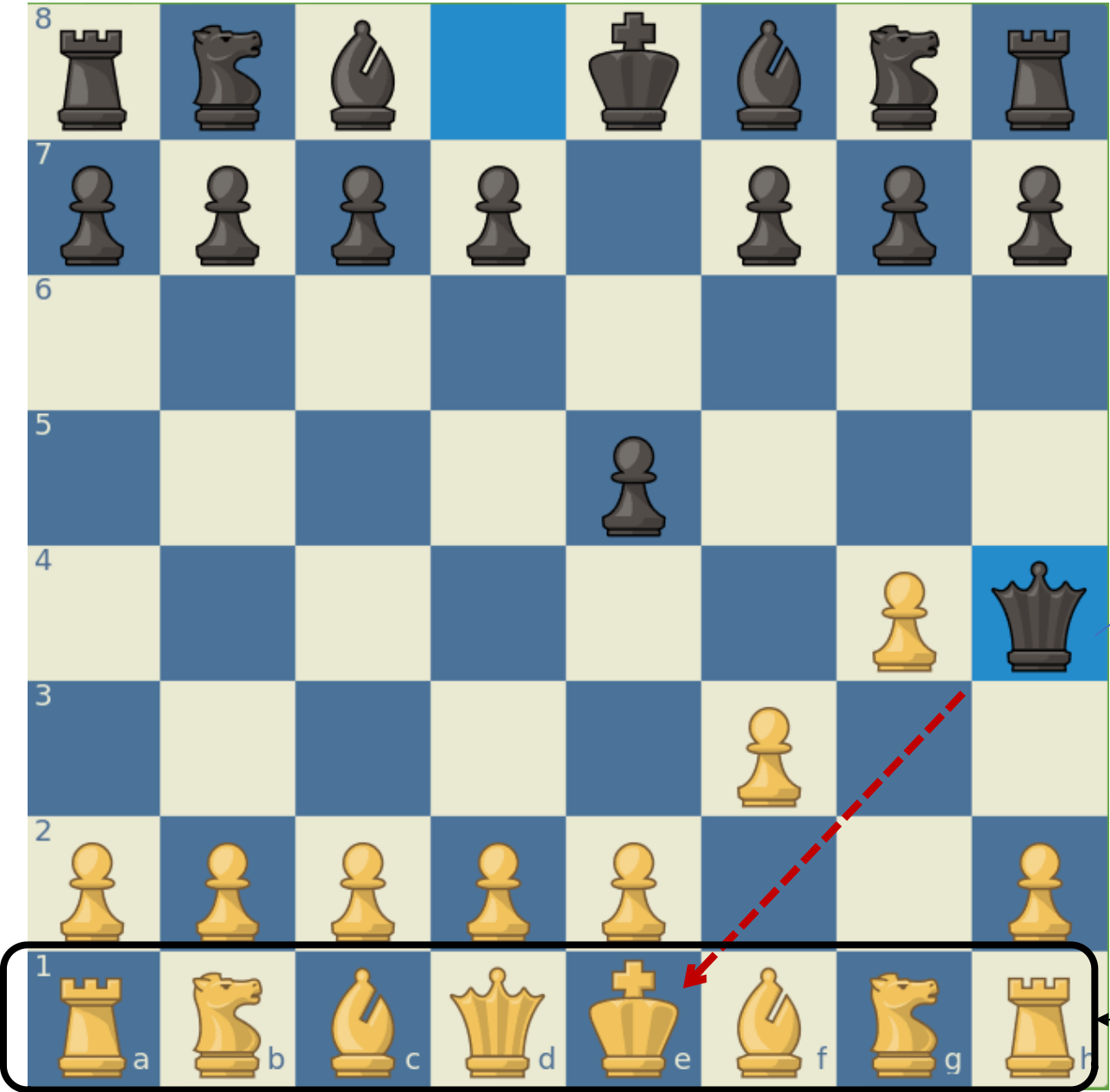
**Password**

Login

HINT 1: username: user' or '1'='1 Password: whatever

HINT 2: username: whatever Password: ' UNION select \* from user where '1'='1

Injection  
Chess  
comparison



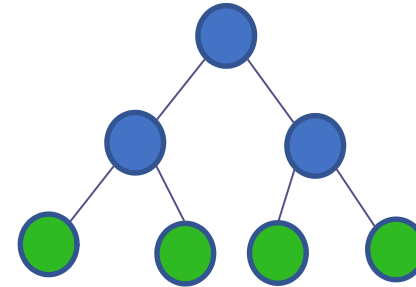
**Bad defense,  
no one can  
stop the  
attack.**

**It is the  
fastest check  
mate of the  
history.**

backend

# Injection: solutions

- Validate each input. “Don’t trust in the end user”
  - Whitelist
  - Regular expression
  - String length
  - Data type
- Use modern frameworks
- Use prepared statements (SQL Injection)
- Use stored procedure (SQL Injection)



[https://cheatsheetseries.owasp.org/cheatsheets/SQL\\_Injection\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html)

# OWASP TOP 10

Insufficient  
Logging

Injection

**Broken  
Authentication**

Using components  
with vulnerabilities

Sensitive data  
exposure

Insecure  
Deserialization

XML External Entity

Cross Site  
Scripting

Broken Access  
Control

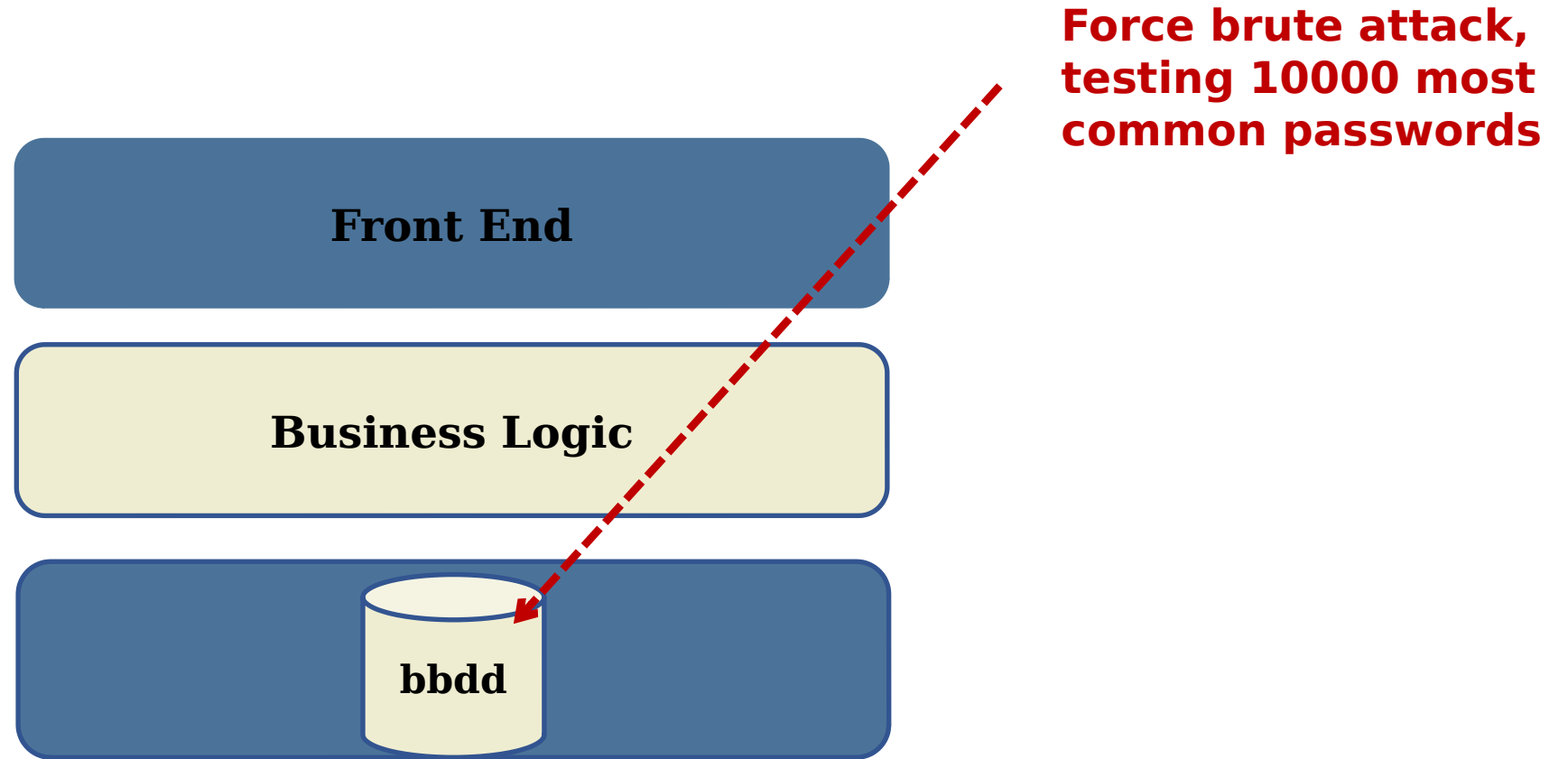
Security  
Misconfiguration

# Broken authentication - Definition

Application functions related to **authentications and session management** are often implemented incorrectly, allowing attackers to **compromise passwords, keys, or session tokens**, or to exploit other implementations flaws to assume other users' identities temporarily or permanently.



# Broken authentication - Demo



# Broken authentication - solutions

Multi factor authentication (**MFA**)

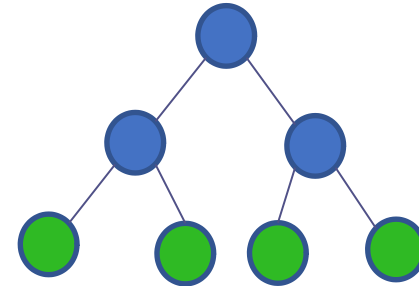
Remove any **default credentials**

Implement **weak-passwords check**

Implement **password policies** (NIST 800-63B).

Limit or increasingly delay **failed login attempts**.

**Session IDs** should not be in the URL.



# OWASP TOP 10

Insufficient  
Logging

Injection

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

**Sensitive data  
exposure**

Insecure  
Deserialization

XML External Entity

Cross Site  
Scripting

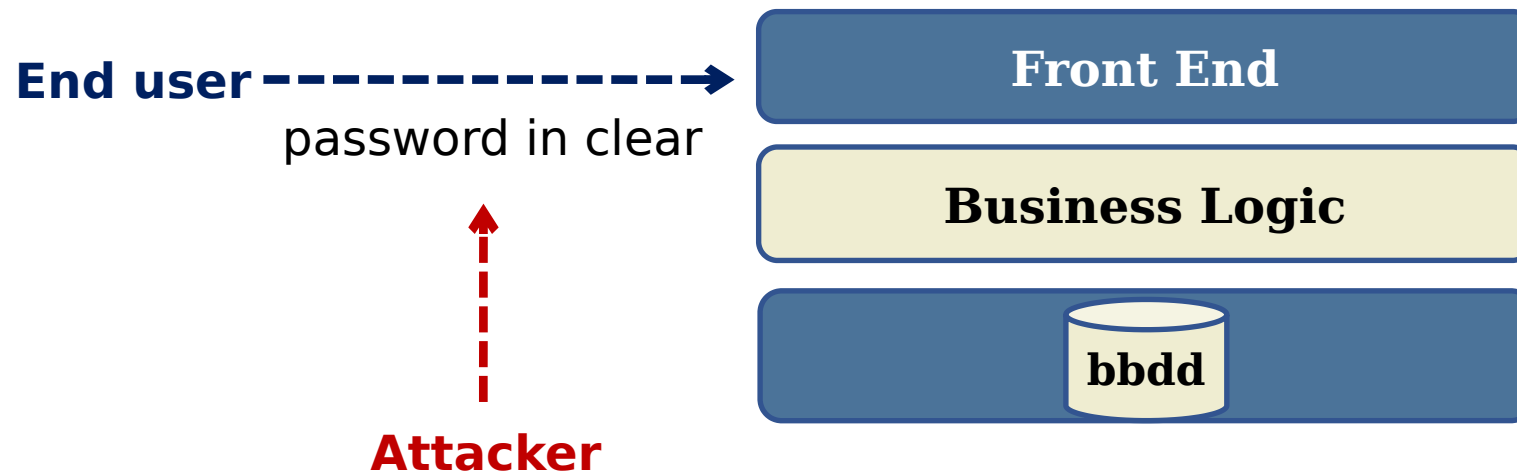
Broken Access  
Control

Security  
Misconfiguration



# Sensitive data exposure - definition

Many web applications and APIs **do not properly protect sensitive data**, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as **encryption** at rest or in transit, and requires **special precautions when exchanged with the browser**.



Sensitive  
data  
exposure  
Chess  
comparison



Exposed  
through open  
columns

Sensitive  
data

# Sensitive data exposure - Solutions

**Classify** data processed, stored or processed by an application. Identify which data is sensitive according to laws, regulatory requirements, or business need.

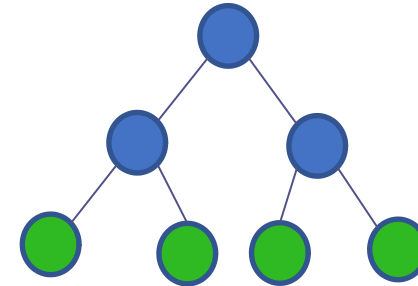
Apply **controls** as per the **classification**

**Don't store sensitive data unnecessarily.**

Make sure to **encrypt** all sensitive data at rest.

Ensure up-to-date and **strong standard algorithms**, protocols and keys are in place.

**Disable caching** for response that contain sensitive data.



# OWASP TOP 10

Insufficient  
Logging

Injection

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

Sensitive data  
exposure

Insecure  
Deserialization

**XML External Entity**

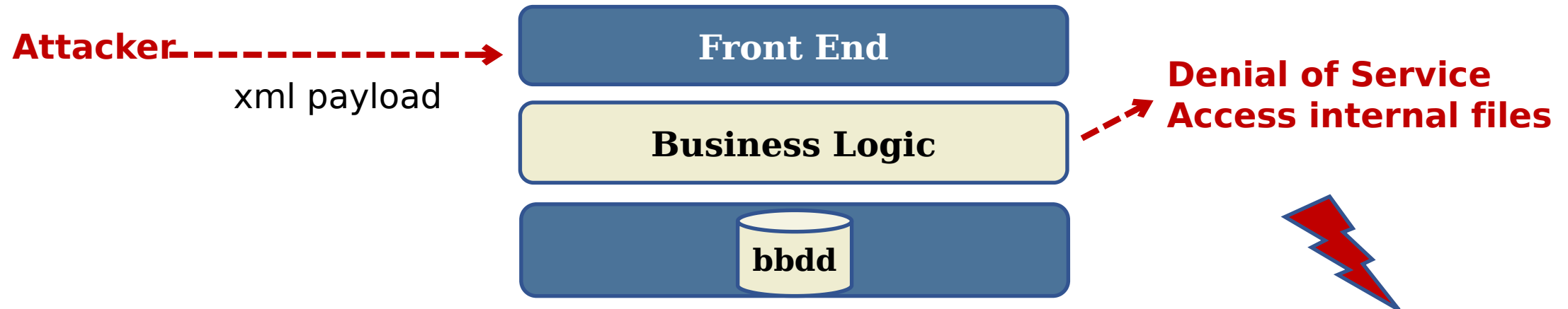
Cross Site  
Scripting

Broken Access  
Control

Security  
Misconfiguration



# XML External Entity - definition

XML External Entities (XXE) Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the URI handler, internal file shares, internal port scanning, remote code execution, and denial of Service attacks.



# XML External Entity – example 1

XML



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<profiles>
  <profile>
    <name> Siva </name>
    <Age> 24 </Age>
    <occupation> Lead </occupation>
  </profile>
  <profile>
    <name> Subbu </name>
    <Age> 25 </Age>
    <occupation> Developer </occupation>
  </profile>
</profiles>
```

**Access internal files**

# XML External Entity – example 1

XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE name
  <!ELEMENT name ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<profiles>
  <profile>
    <name>`&xxe;`</name>
    <address>`test`</address>
  </profile>
</profiles>
```

**Access internal files**

# XML External Entity – example 2

XML



```
<?xml version="1.0"?>
<!DOCTYPE lolz [
  <!ENTITY lol "lol">
  <!ELEMENT lolz (#PCDATA)>
  <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
  <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
  <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
  <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
  <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
  <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
  <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

**Denial of service**



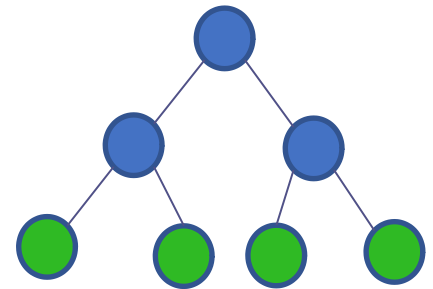
# XML External Entity – solutions

Whenever possible, use **less complex data** formats such as JSON.

**Patch or upgrade all XML processors** and libraries in use by application or on the underlying operating systems.

**Disable XML external entity and DTD processing** in all XML parsers in the application.

Verify that XML or XSL file upload functionality **validates incoming XML using XSD validation** or similar.



# OWASP TOP 10

Insufficient  
Logging

Injection

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

Sensitive data  
exposure

Insecure  
Deserialization

XML External Entity

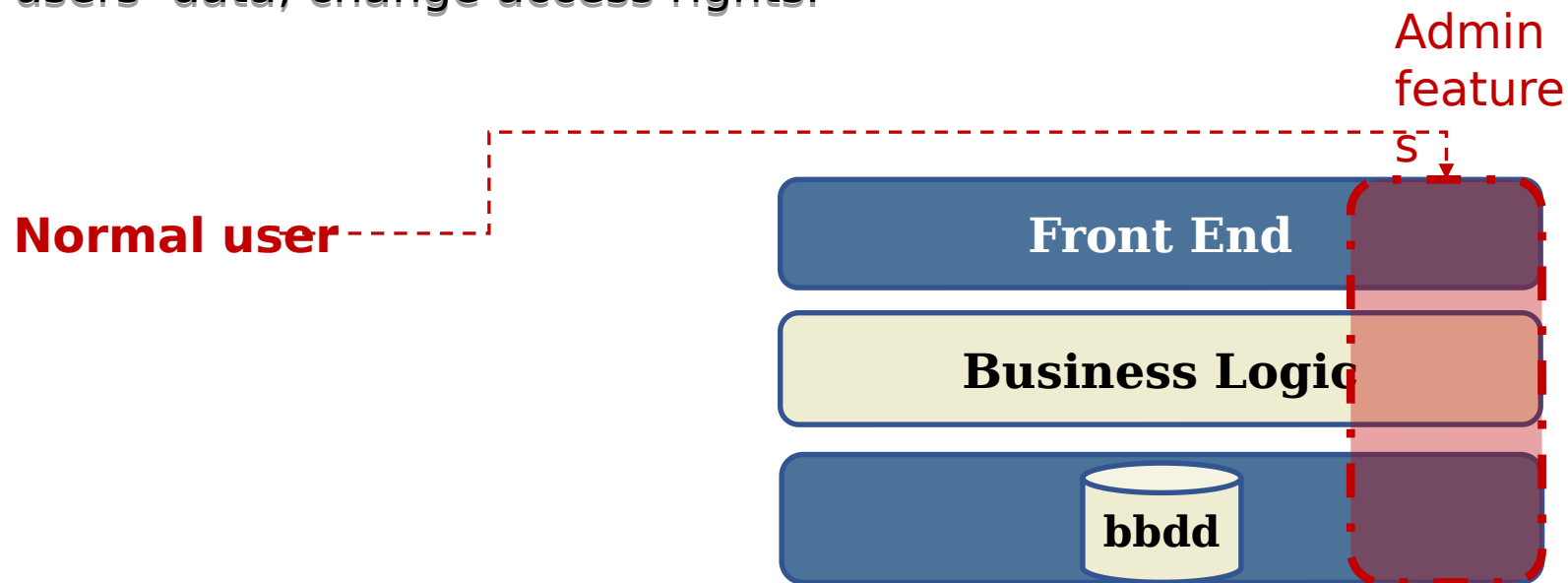
Cross Site  
Scripting

Security  
Misconfiguration

**Broken Access  
Control**

# Broken Access Control - definition

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to **access unauthorized functionality** and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights.

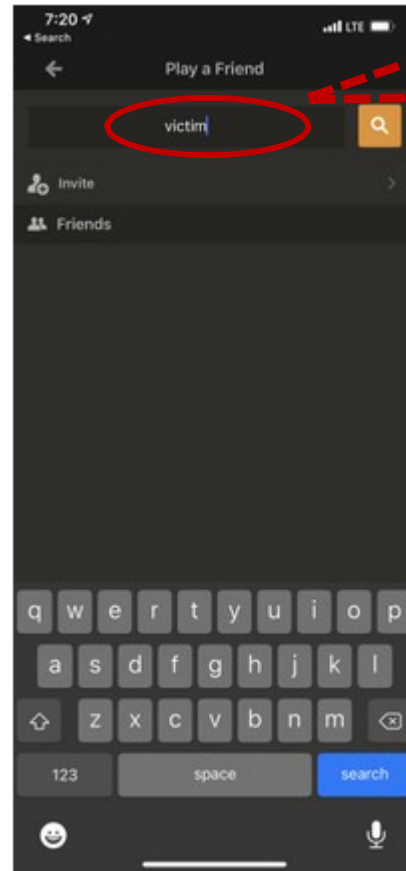


# Broken Access Control - example



**Hacking Chess.com and  
Accessing 50 Million  
Customer Records**

# Broken Access Control - definition



```
GET
/v1/users?loginToken=98a16127fb8cb4dc97a3a02103706890
&username=hikaru&
signed=i0S3.9.7-7b9f1383b669614302e9503ba7db81875e440
d7e HTTP/1.1
Host: api.chess.com
```

```
{
  "status": "success",
  "data": {
    "email": "REDACTED@REDACTED.COM",
    "premium_status": 3,
    "id": 15448422,
    "uuid": "REDACTED",
    "country_id": 2,
    "avatar_url": "https://images.chesscomfiles.com
/uploads/v1/user
/15448422.90503d66.200x200o.f323efa57fd0.jpeg",
    "last_login_date": REDACTED,
    "session_id": "REDACTED",
    "location": "Sunrise, Florida",
    "username": "Hikaru",
    "points": 52,
    "chess_title": "GM",
    "first_name": "Hikaru Nakamura",
    "last_name": null,
    "country_name": "United States",
    "member_since": REDACTED,
    "about": "",
    "is_blocked": false,
    "is_tracked": false,
    "are_friends": false,
    "friend_request_exists": true,
    "is_able_to_change_username": null,
    "flair_code": "diamond_traditional",
    "show_ads": true,
    "is_fair_play_agreed": true
  }
}
```

# Broken Access Control - definition



I clicked around the app until I remembered something from earlier testing: there was an "admin.chess.com" subdomain.

# Broken Access Control - definition

## Timeline

- 12/12/2020, 12:34 AM - Reported
- 12/12/2020, 03:17 AM - Validated
- 12/12/2020, 07:42 AM - Remediated
- 12/16/2020, 02:40 PM - Rewarded

# Broken Access Control - solution

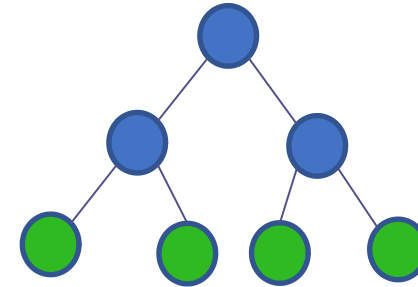
Implement **access control mechanism** once and re-use them

**Disable web server directory listing** and ensure file metadata and backup files are not present within web roots.

**Log access control** failures, alerts admins when appropriate.

**Rate limit API** and controller access to minimize the harm from automated attack tooling.

JWT **tokens** should be **invalidated** on the server after logout.





# OWASP TOP 10

Insufficient  
Logging

Injection

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

Sensitive data  
exposure

Insecure  
Deserialization

XML External Entity

Cross Site  
Scripting

Broken Access  
Control

**Security  
Misconfiguration**

# Security misconfiguration - definition

Security misconfiguration is the **most commonly** seen issue. This is commonly a result of **insecure default configurations**, incomplete or ad hoc configurations, **open cloud storage**, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all **operating systems, frameworks, libraries and applications** be securely configured, but they must be **patched/upgraded** in a timely fashion.



# Security misconfiguration - example

[NEWS](#) ▾[DOWNLOADS](#) ▾[VIRUS REMOVAL GUIDES](#) ▾[TUTORIALS](#) ▾

[Home](#) > [News](#) > [Security](#) > FBI: Hackers stole government source code via SonarQube instances

## FBI: Hackers stole government source code via SonarQube instances

### Previous attacks and mitigation measures

The threat actors start their attacks by first scanning for Internet-exposed SonarQube instances using the default port number (i.e., 9000) the FBI explains.

After discovering an exposed server, they attempt to gain access to vulnerable instances using default admin/admin credentials.

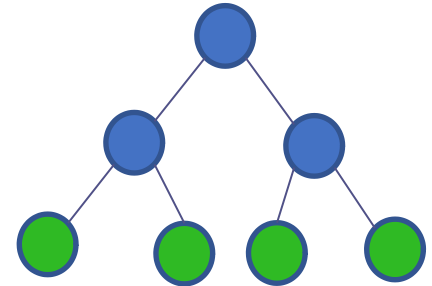
# Security misconfiguration - solutions

A repeatable **hardening** process that makes it fast and easy to deploy another environment that is properly locked down.

A **minimal platform** without any unnecessary features, components, documentation, and samples.

A **task to review** and update the configurations appropriate to all security notes, updates and patches as part of the patch management process.

A **segmented application architecture** that provides effective, secure separation between components or tenants, with segmentation, containerization, or cloud security groups.



# OWASP TOP 10

Insufficient  
Logging

Injection

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

Sensitive data  
exposure

Insecure  
Deserialization

XML External Entity

**Cross Site  
Scripting**

Broken Access  
Control

Security  
Misconfiguration

# Cross site scripting

XSS flaws occur whenever an application **includes untrusted data** in a web page without proper validation or escaping, or updates an existing web page with-user supplied data using a browser API that can create HTML or Javascript. XSS allow attackers **to execute scripts in the victim's browser** which can hijack user sessions, deface web sites, or redirect the user to malicious sites.



# Cross site scripting

## Vulnerable Chess Game

You can try Cross Site Scripting (XSS) in this page ;-)

According to OWASP TOP TEN: it is risk #7, more info [here](#)

Welcome kasparov!

Set players names and the time for the match

# Cross site scripting

Using frameworks that automatically escape XSS by design, such as the latest Ruby on Rails, React JS.

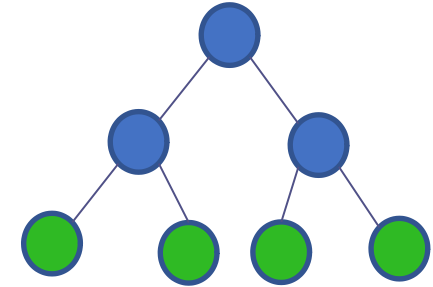
Escaping untrusted HTTP request data based on the in the HTML output.

Validate all inputs.

Enable a Content Security Policy (CSP) as a defense-in-depth mitigating control against XSS.

*To enable XSP you need to configure your web server to return the Content-Security-Policy HTTP header.*

*CSP makes it possible for server administrator to reduce or eliminate the vectors by which XSS can occur by specifying the domains that the browser should consider to be valid sources of executable scripts.*





# OWASP TOP 10

Insufficient  
Logging

Injection

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

Sensitive data  
exposure

**Insecure  
Deserializatio  
n**

XML External Entity

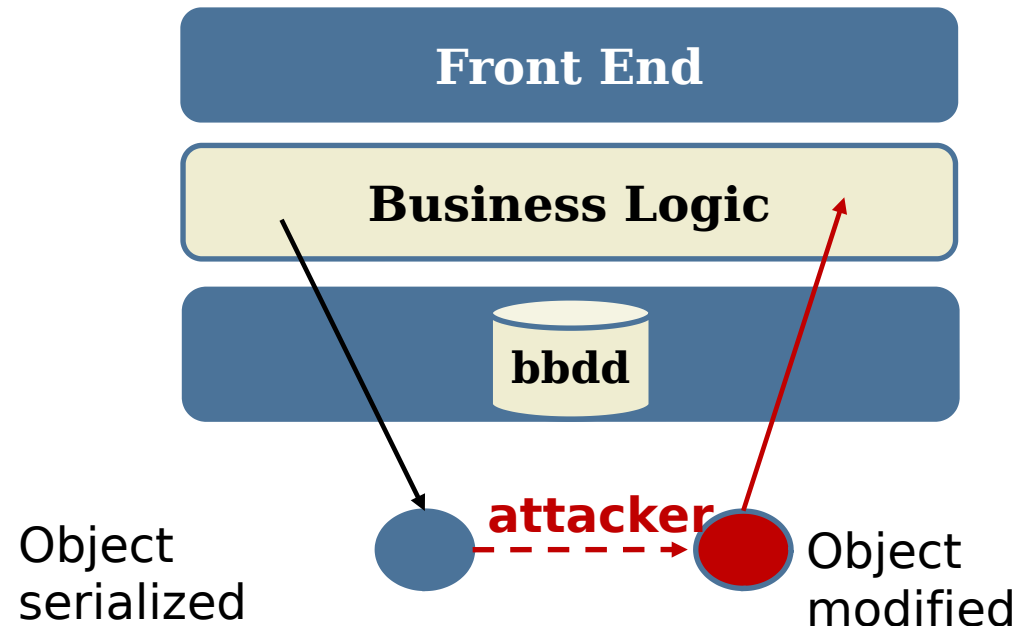
Cross Site  
Scripting

Broken Access  
Control

Security  
Misconfiguration

# Insecure deserialization - definition

Insecure deserialization often leads **to remote code execution**. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.



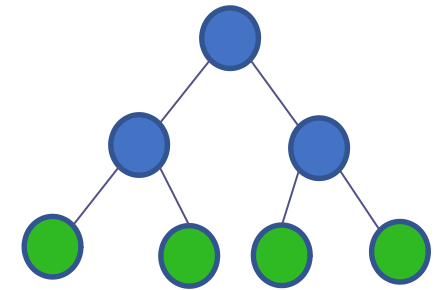
# Insecure deserialization - solution

Implementing integrity checks such as **digital signatures** on any serialized objects to prevent hostile object creation or data tampering.

**Isolating and running code** that deserializes in low privilege environments when possible.

**Restricting** or monitoring **incoming and outgoing network connectivity** from containers or servers that deserialize.

**Monitoring** deserialization, alerting if a user deserializes constantly.



# OWASP TOP 10

Insufficient  
Logging

Injection

Broken  
Authenticatio  
n

**Using  
components with  
vulnerabilities**

Sensitive data  
exposure

Insecure  
Deserialization

XML External Entity

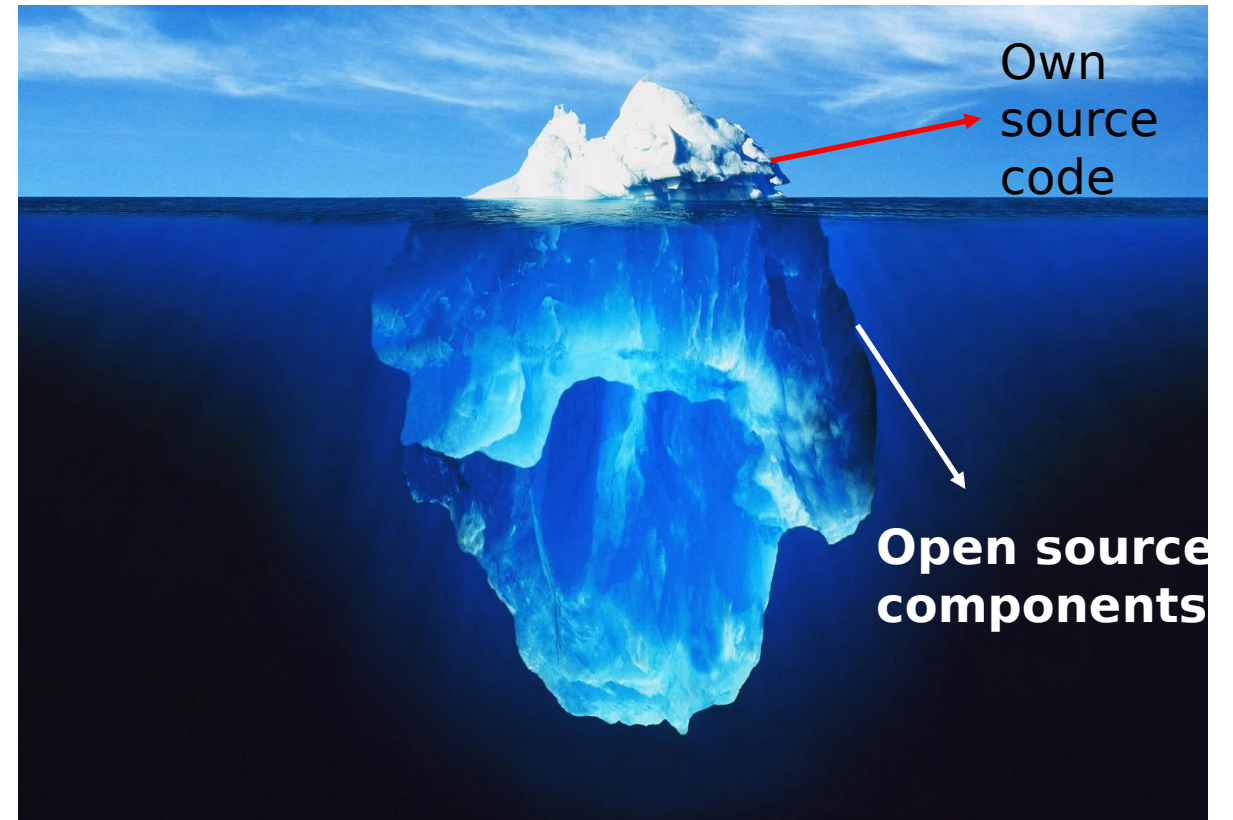
Cross Site  
Scripting

Broken Access  
Control

Security  
Misconfiguration

# Using components with vulnerabilities - definition

Components, such as **libraries**, **frameworks**, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with **known vulnerabilities** may undermine application defenses and enable various attacks and impacts.



# Using components with vulnerabilities – example 1

pip install package-name

**Typosquatting attack**

**pip install reqeusts**

**pip install coffe-script**



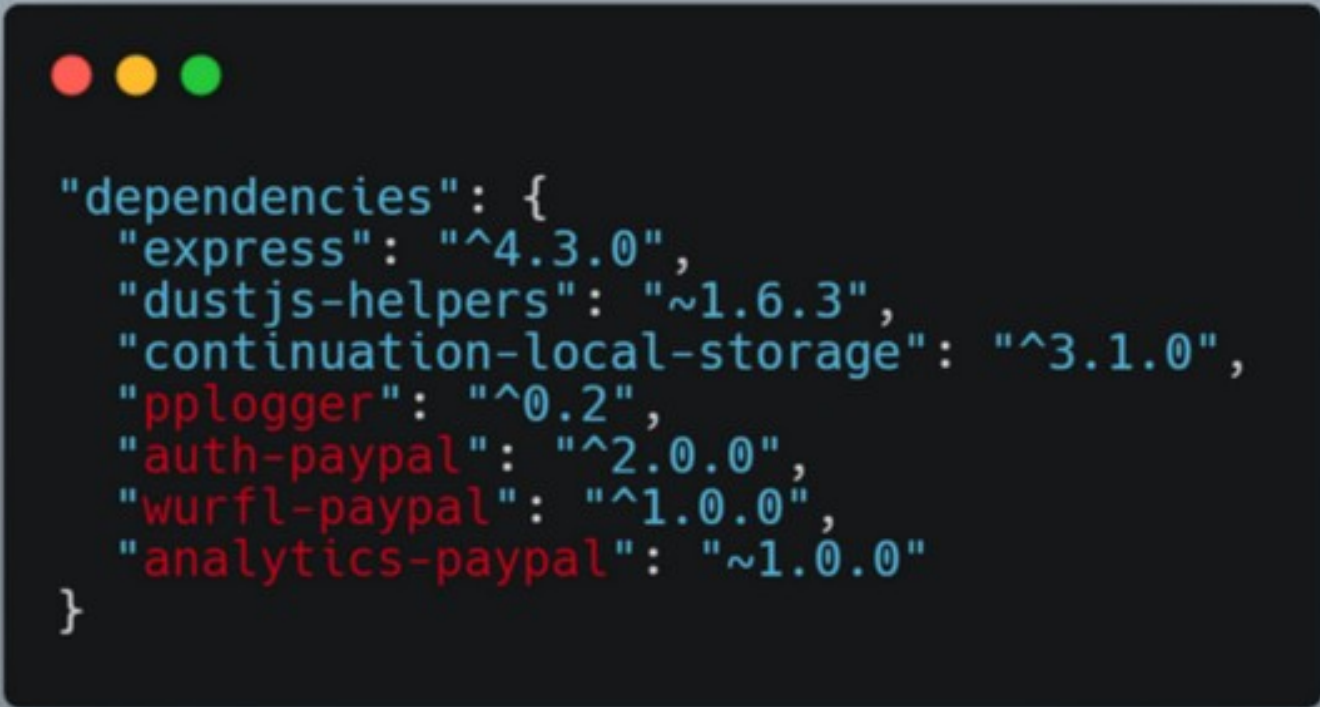
# Using components with vulnerabilities – example 2

**Dependency Confusion: How I Hacked Into Apple, Microsoft and Dozens of Other Companies**





# Using components with vulnerabilities – example 2

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays a JSON object representing a list of dependencies.

```
"dependencies": {  
  "express": "^4.3.0",  
  "dustjs-helpers": "~1.6.3",  
  "continuation-local-storage": "^3.1.0",  
  "pplogger": "^0.2",  
  "auth-paypal": "^2.0.0",  
  "wurfl-paypal": "^1.0.0",  
  "analytics-paypal": "~1.0.0"  
}
```



# Using components with vulnerabilities – example 2

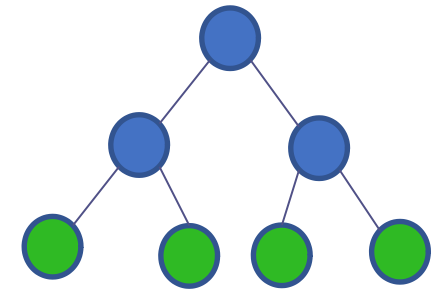
Time	Organization	IP Address	Package Name	Hostname	Current Path
FRI AUG 21 2020 16:37:56 GMT	APPLE-ENGINEERING - Apple Inc.	17.149.2	@idms/idms-pmrc	.lan	/Users/ /gitlab/appleauth/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
FRI AUG 21 2020 20:14:32 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.	@idms/idms-pmrc	8faa3092cc97	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
FRI AUG 21 2020 20:15:23 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	91c057281d0f	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
MON AUG 24 2020 17:40:43 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	1f3cc975c67b	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
MON AUG 24 2020 17:41:38 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.1	@idms/idms-pmrc	fe01f79c7146	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
MON AUG 24 2020 17:46:06 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	7df2bb892313	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
MON AUG 24 2020 17:46:07 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.1	@idms/idms-pmrc	c6269b74ec56	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
MON AUG 24 2020 19:55:16 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.1	@idms/idms-pmrc	580f8f68bad3	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
MON AUG 24 2020 19:55:37 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	d7bea26b6122	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 07:15:17 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.	@idms/idms-pmrc	f507d7c91170	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 07:16:00 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	e0b80fce2ded	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:04:20 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	fab9b33c62b4	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:21:45 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	dfec8557ad01	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:22:24 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.	@idms/idms-pmrc	23495738a747	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:22:33 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.	@idms/idms-pmrc	0b238c2f3792	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:23:34 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	b9986c648086	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:23:56 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	b44ff6b9bd5b	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:24:01 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.1	@idms/idms-pmrc	dbe40d2f0d7b	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:35:18 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.1	@idms/idms-pmrc	46ec329453e0	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:35:26 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	493d6929fa02	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:35:31 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.	@idms/idms-pmrc	efdbc138d349	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:35:42 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	c1f3c7e9dd7b	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:56:39 GMT	APPLE-ENGINEERING - Apple Inc.	17.151.1	@idms/idms-pmrc	s-MacBook-Pro.local	/Users/ /Repositories/idms/appleauth/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 17:56:39 GMT	APPLE-ENGINEERING - Apple Inc.	17.150.2	@idms/idms-pmrc	s-MacBook-Pro.local	/Users/ /Repositories/idms/appleauth/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 19:12:59 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	ef4d6be2634f	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 19:28:51 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	74fb58c6b33f	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 19:38:10 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	2f9a02c2d36e	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 20:15:06 GMT	APPLE-ENGINEERING - Apple Inc.	17.151.1	@idms/idms-pmrc	MacBook-Pro.local	/Users/ Documents/workspace/apple/appleauth/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 20:15:06 GMT	APPLE-ENGINEERING - Apple Inc.	17.149.2	@idms/idms-pmrc	MacBook-Pro.local	/Users/ Documents/workspace/apple/appleauth/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 21:33:49 GMT	APPLE-ENGINEERING - Apple Inc.	17.171.1	@idms/idms-pmrc	51659637f4bc	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc
TUE AUG 25 2020 21:34:14 GMT	APPLE-ENGINEERING - Apple Inc.	17.122.	@idms/idms-pmrc	37ed0d2d0047	/workspace/node_modules/@idms/idms-widget-auth-service/node_modules/@idms/idms-pmrc

# Using components with vulnerabilities – solutions

Remove **unused dependencies**, unnecessary features, components, files and documentation.

Continuously **inventory the versions** of both client-side and server-side components (frameworks and libraries) and their dependencies and monitor sources like **CVE and NVD for vulnerabilities** in the components.

Only obtain components from **official sources** over secure links. Prefer signed packages to reduce the chance of including a modified, malicious component.



# OWASP TOP 10

**Insufficient  
Logging**

Injection

Broken  
Authenticatio  
n

Using components  
with vulnerabilities

Sensitive data  
exposure

Insecure  
Deserialization

XML External Entity

Cross Site  
Scripting

Broken Access  
Control

Security  
Misconfiguration

# Insufficient Logging and Monitoring - definition

Insufficient Logging and Monitoring, coupled with missing or ineffective integration with incident response, **allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.**

Most breach studies **show time to detect a breach is over 200 days**, typically detected by external parties rather than internal processes or monitoring.

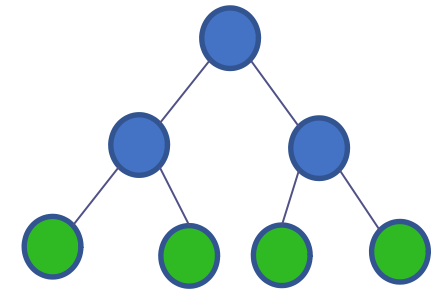


# Insufficient Logging and Monitoring - solutions

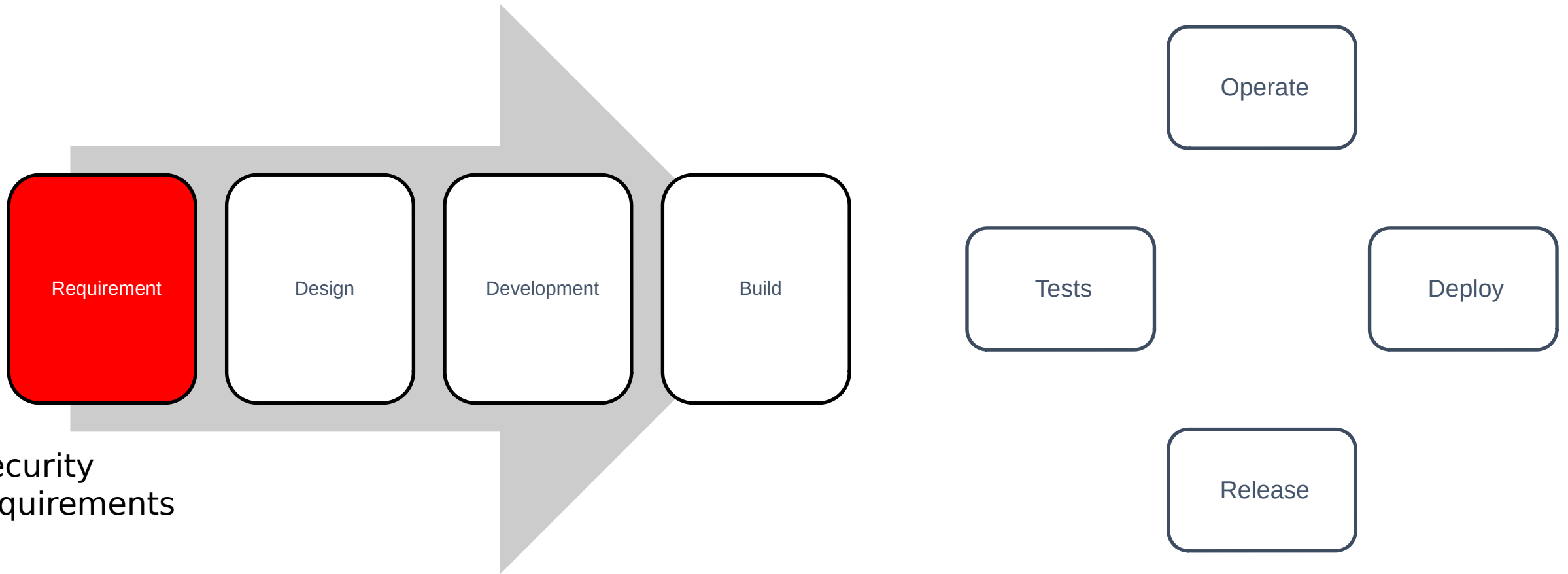
Ensure all login, access control failures, and server-side input validation failures **can be logged with sufficient user context** to identify suspicious or malicious accounts, and held for sufficient time to allow delayed forensic analysis.

Ensure that logs are generated in a **format** that can be easily consumed by a centralized log management solutions.

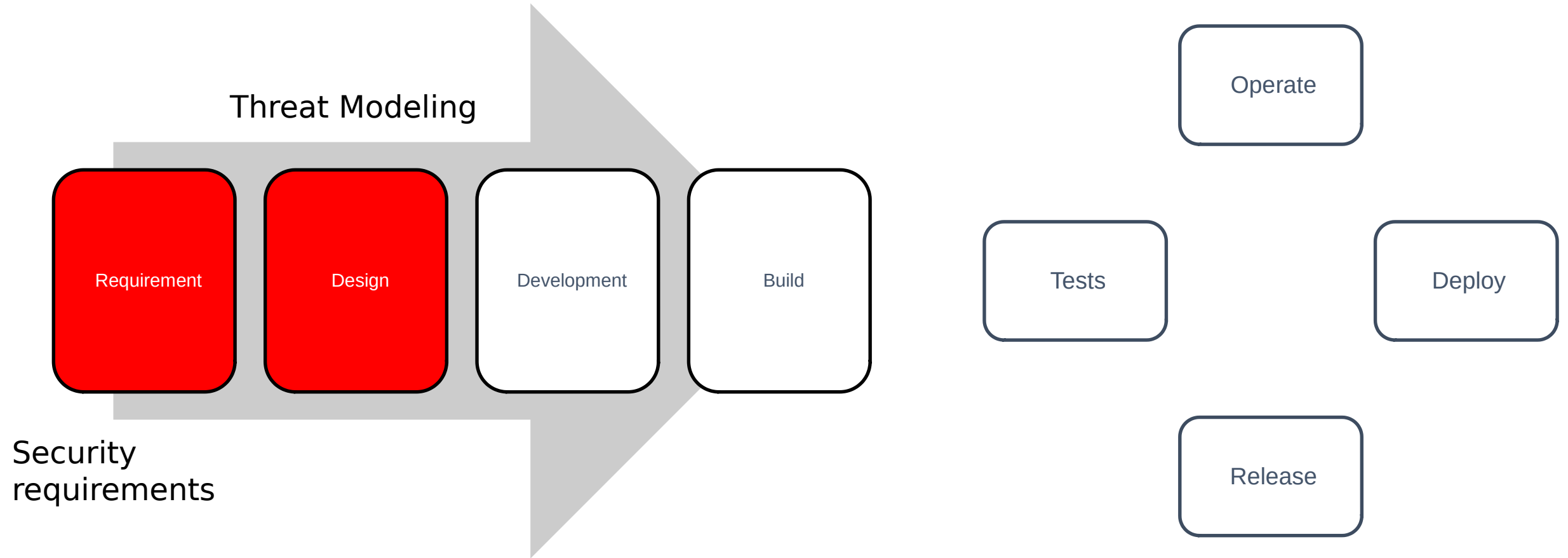
Ensure **high-value transactions have an audit trail with integrity controls** to prevent tampering or deletion, such as append-only database tables or similar.



# Owasp and secure software life cycle

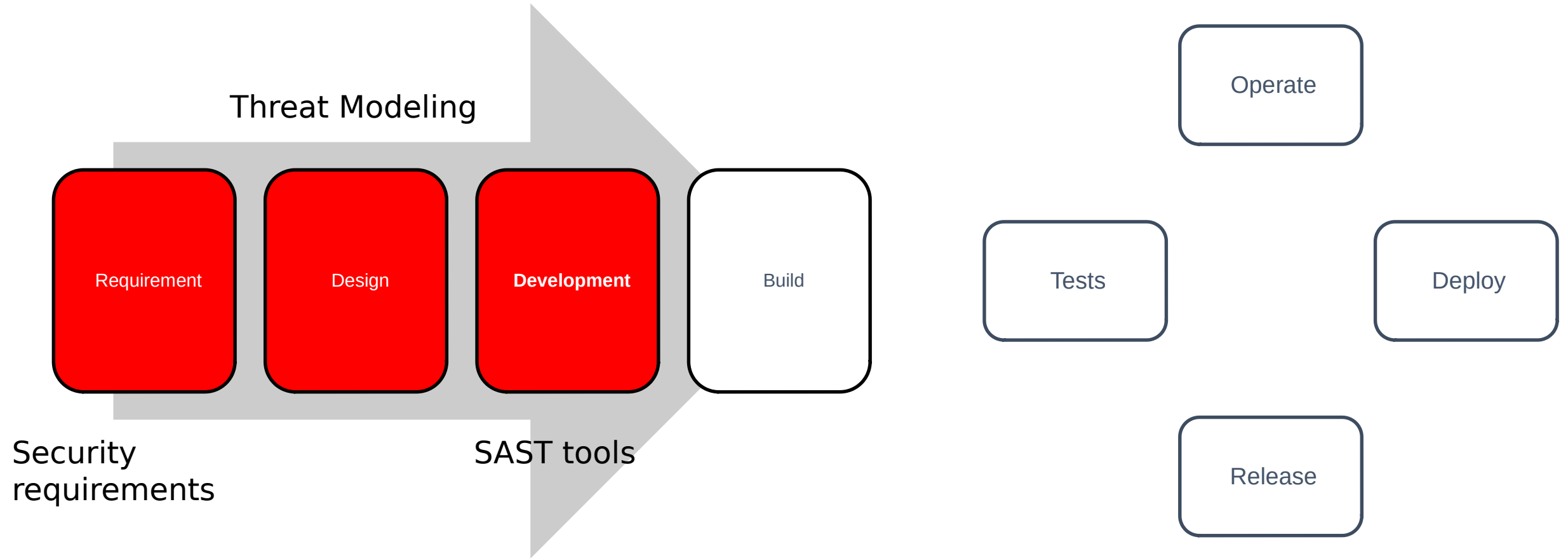


# Owasp and secure software life cycle



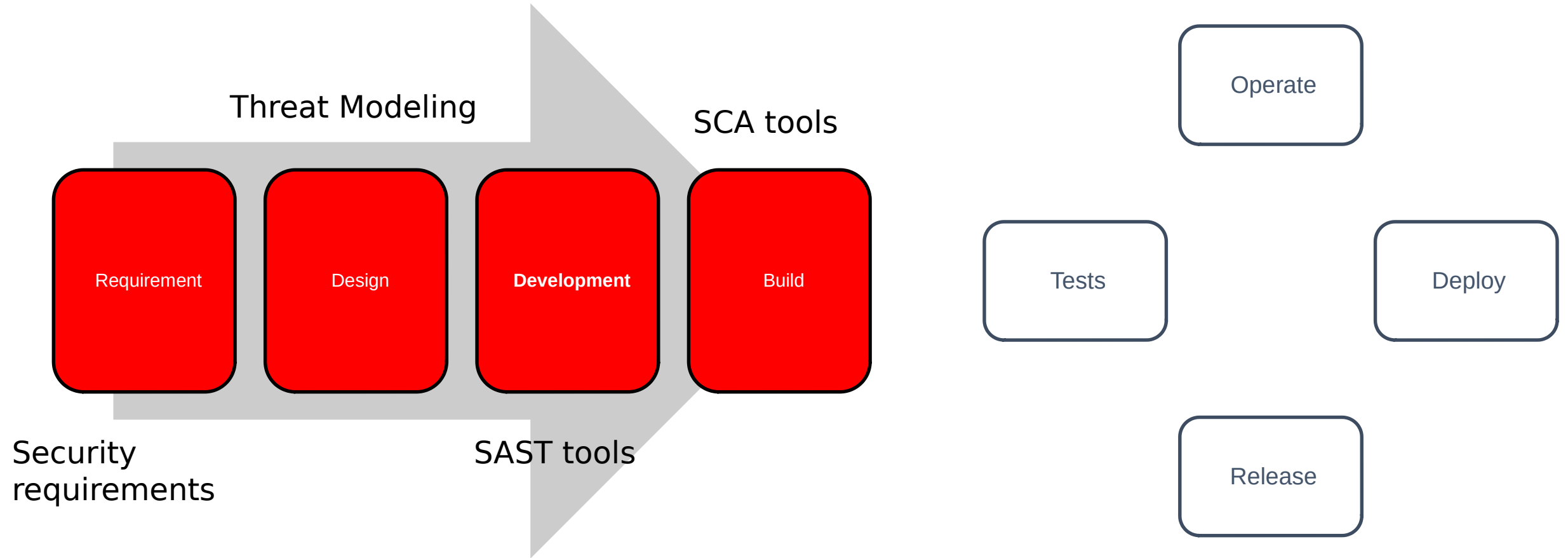


# Owasp and secure software life cycle

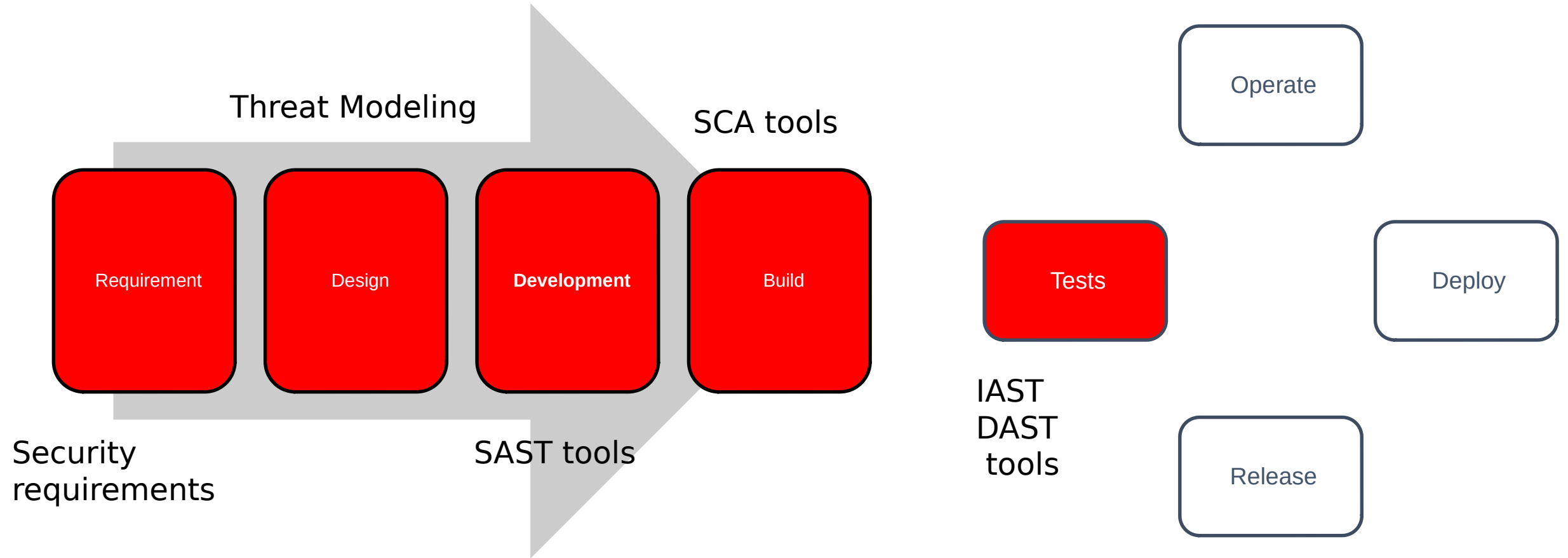




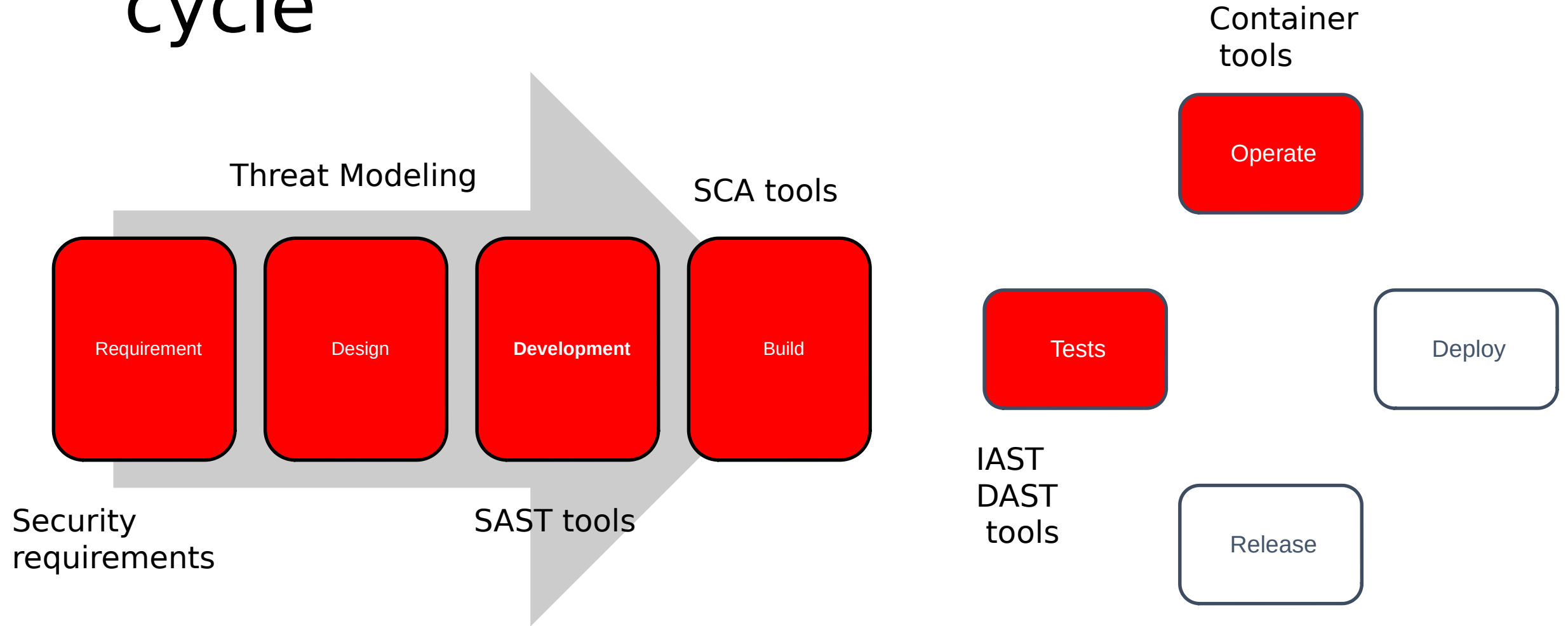
# Owasp and secure software life cycle



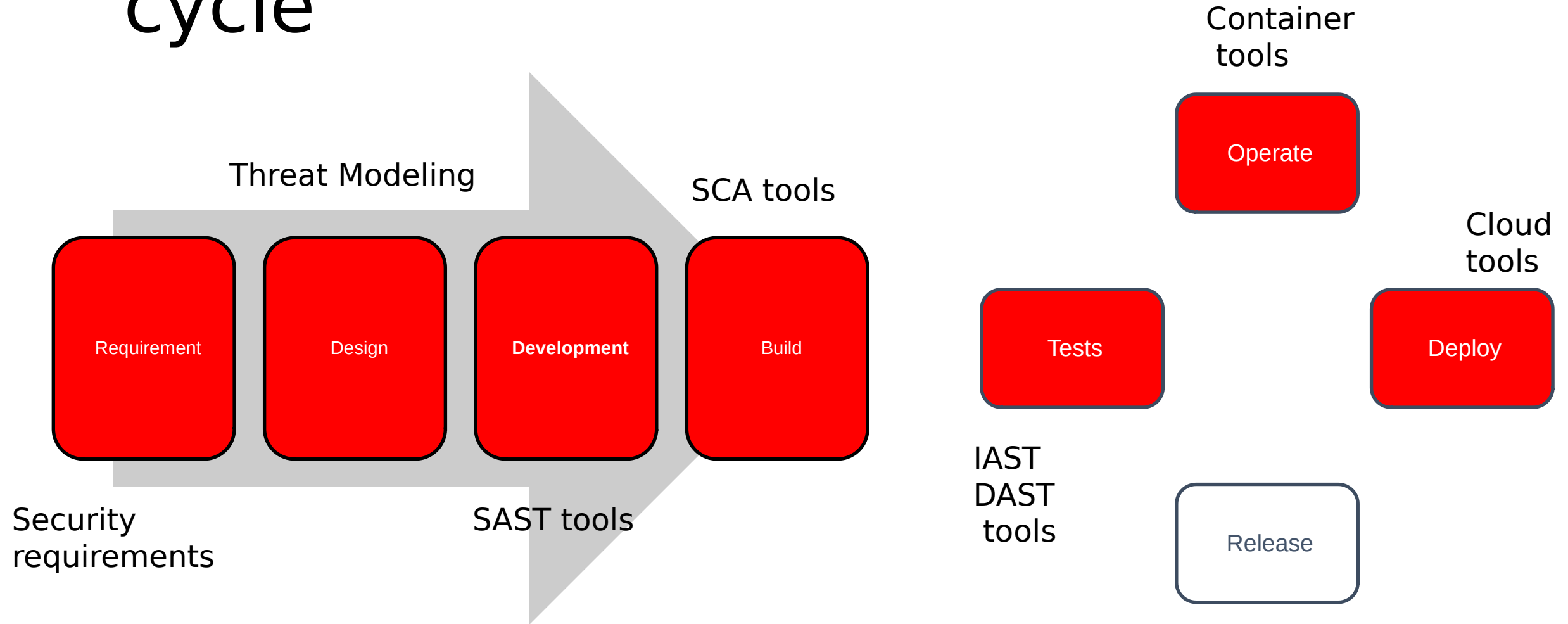
# Owasp and secure software life cycle



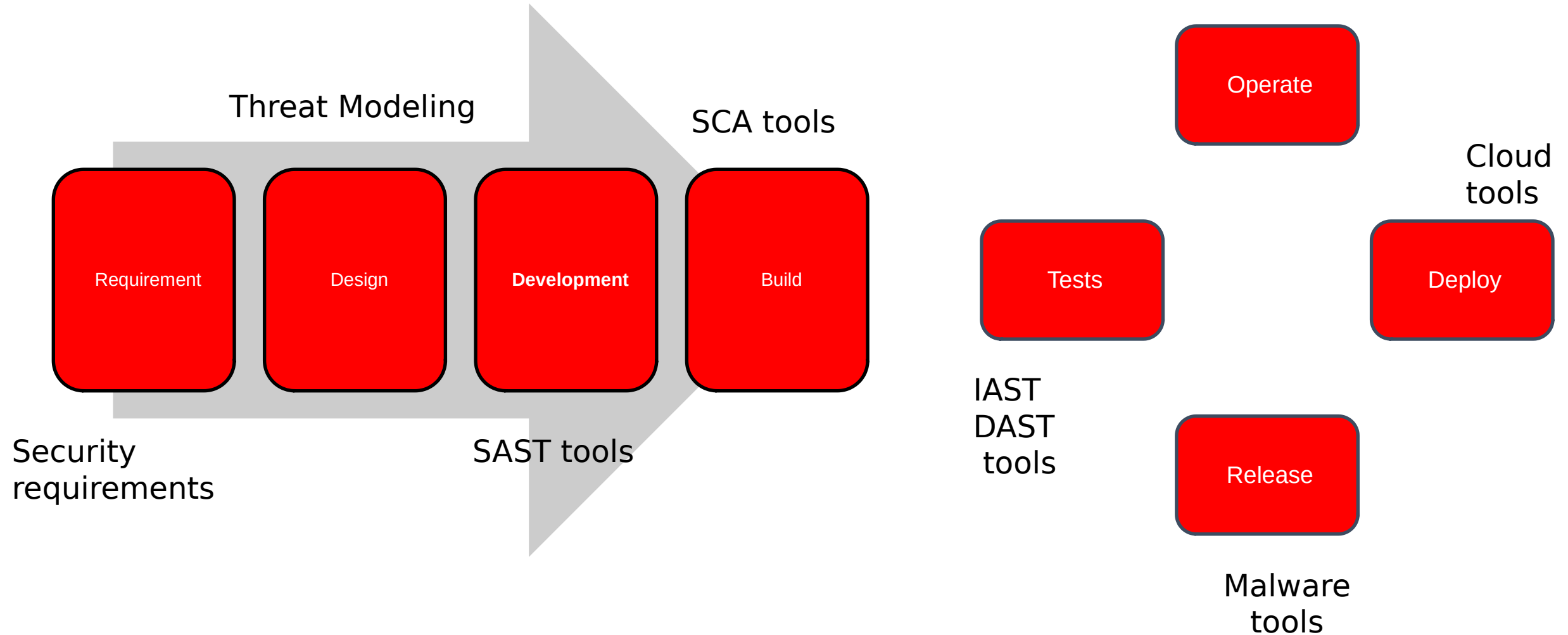
# Owasp and secure software life cycle



# Owasp and secure software life cycle



# Owasp and secure software life cycle



Thanks ++

[sebastianrevuelta@gmail.com](mailto:sebastianrevuelta@gmail.com)  
[https://www.linkedin.com/in/  
sebasrevuelta/](https://www.linkedin.com/in/sebasrevuelta/)