

Introducción a la

Los diez mejores de OWASP



Kirk Jackson

RedShield

kirk@pageofwords.com

<http://hack-ed.com>

@kirkj

OWASP

Nueva Zelanda [https://](https://www.meetup.com/)

www.meetup.com/

OWASP-Wellington/

www.owasp.org.nz @

Grabaciones:

<https://goo.gl/a2VSG2>

¿Qué es OWASP?

Open Web Application Security Project (OWASP) es una fundación sin fines de lucro que trabaja para mejorar la seguridad del software.

- Un sitio web: owasp.org
- Un montón de herramientas interesantes: Zed Attack Proxy, Juice Shop, controles proactivos, modelo de madurez de garantía de software (SAMM), estándar de verificación de seguridad de aplicaciones (ASVS).
- Una comunidad global de personas con ideas afines, reuniones y conferencias.



Who is the OWASP Foundation?

The Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.

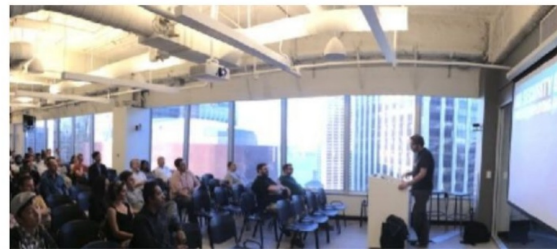
- Tools and Resources
- Community and Networking
- Education & Training

For nearly two decades corporations, foundations, developers, and volunteers have supported the OWASP Foundation and its work. [Donate](#), [Join](#), or become a [Corporate Member](#) today.

Project Spotlight: Zed Attack Proxy



Featured Chapter: Bay Area



OWASP Top Ten

[Main](#)[Sponsors](#)

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications.

Globally recognized by developers as the first step towards
more secure coding.

Companies should adopt this document and start the process of ensuring that their web applications minimize these risks. Using the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces more secure code.

Top 10 Web Application Security Risks

1. **Injection**. Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur 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.
2. **Broken Authentication**. Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
3. **Sensitive Data Exposure**. Many web applications and APIs do not properly protect sensitive data, such as

[Watch](#)





18

[★ Star](#)

32

The OWASP Foundation works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

Project Information

-  Flagship Project
-  Documentation
-  Builder
-  Defender

[Current Version \(2017\)](#)

Downloads or Social Links

[Download](#)[Social Link](#)

Code Repository

[repo](#)

Leaders

[Nail G. J. Li](#)

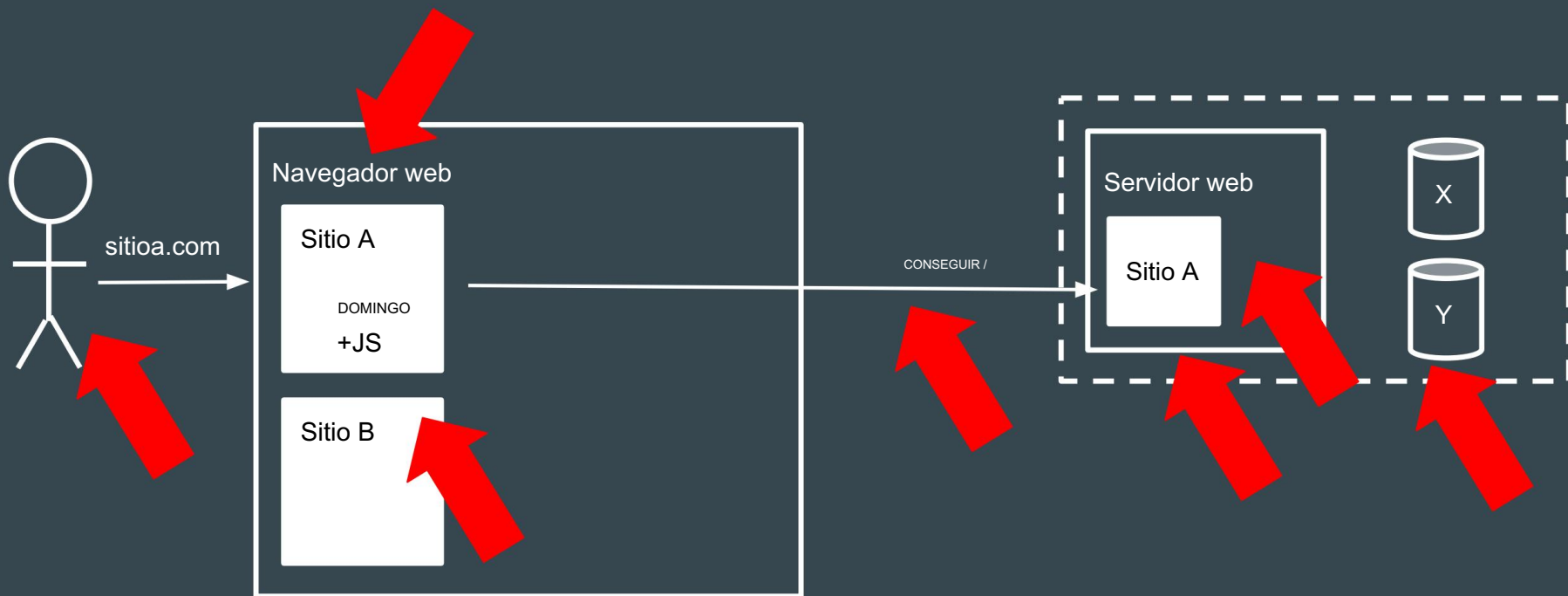
Los diez mejores de OWASP

Reconocido mundialmente por los desarrolladores como el primer paso hacia una codificación más segura.

Los riesgos de seguridad más críticos para las aplicaciones web.

Actualizado cada 2-3 años desde 2003 hasta 2017 (2020
está en progreso)

Asegurar al usuario



OWASP Top Ten 2017

A1 Inyección

Autenticación A2 rota

Exposición de datos confidenciales A3

Entidades externas XML A4 (XXE)

Control de acceso roto A5

Configuración incorrecta de seguridad A6

Secuencias de comandos entre sitios A7 (XSS)

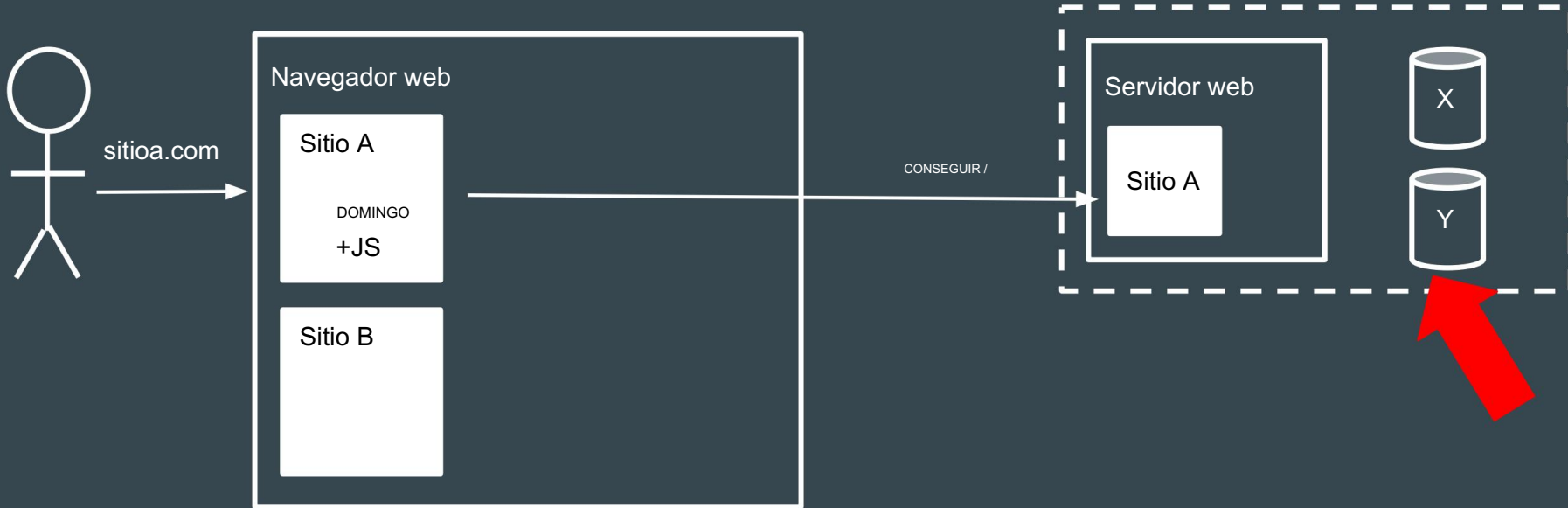
A8 Deserialización insegura

A9 Uso de componentes con vulnerabilidades conocidas

A10 Registro y monitoreo insuficientes

A1 Inyección

Envío de datos hostiles a un intérprete (por ejemplo, SQL, LDAP, línea de comando)



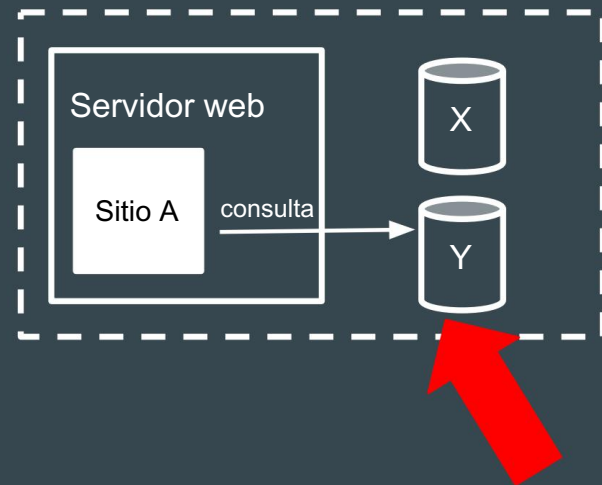
A1 Inyección

Envío de datos hostiles a un intérprete (por ejemplo, SQL, LDAP, línea de comando)

Consulta de cadena = "SELECCIONAR * DE cuentas DONDE custID=" + request.getParameter("id") + "";

identificación = " "; eliminar cuentas de la tabla --

Las declaraciones SQL combinan código y datos



Demostración de SQLi

A1 Inyección

Prevención:

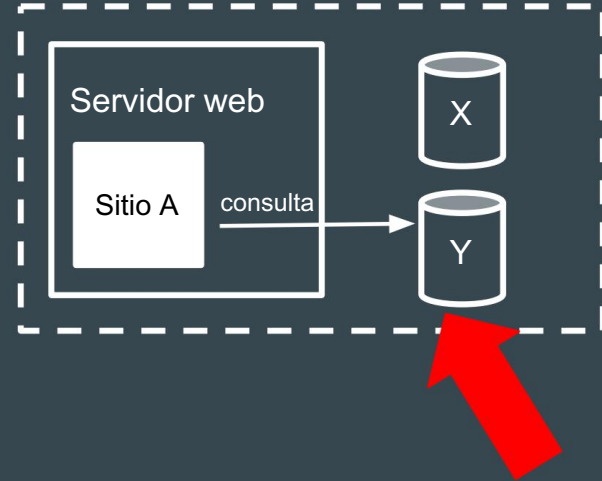
Las declaraciones SQL combinan código y datos

=> Código y datos separados

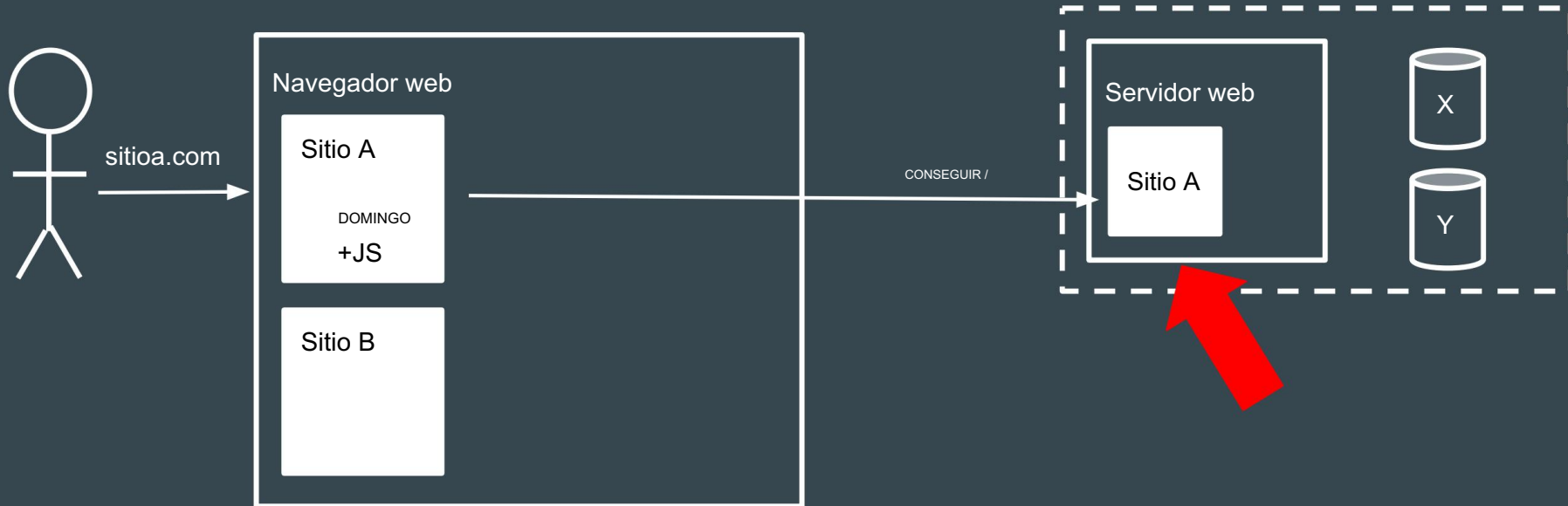
- Parametrizar tus consultas •

Validar qué datos se pueden introducir

- Escapar de caracteres especiales



Autenticación A2 rota



Autenticación A2 rota

- Gestión de sesiones débil •

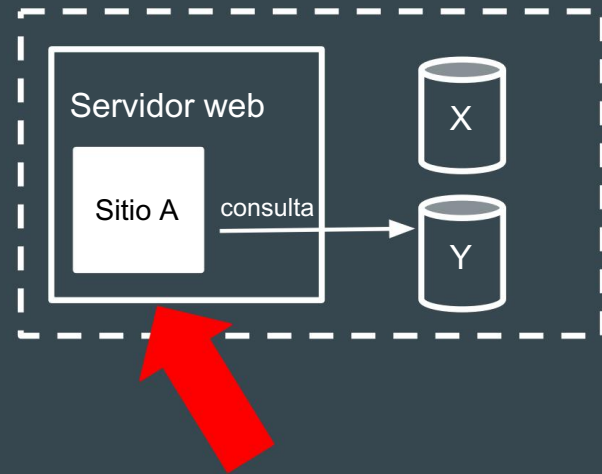
Relleno de credenciales

- Fuerza bruta

- Contraseña olvidada •

Sin autenticación multifactor

- Las sesiones no caducan



Autenticación A2 rota

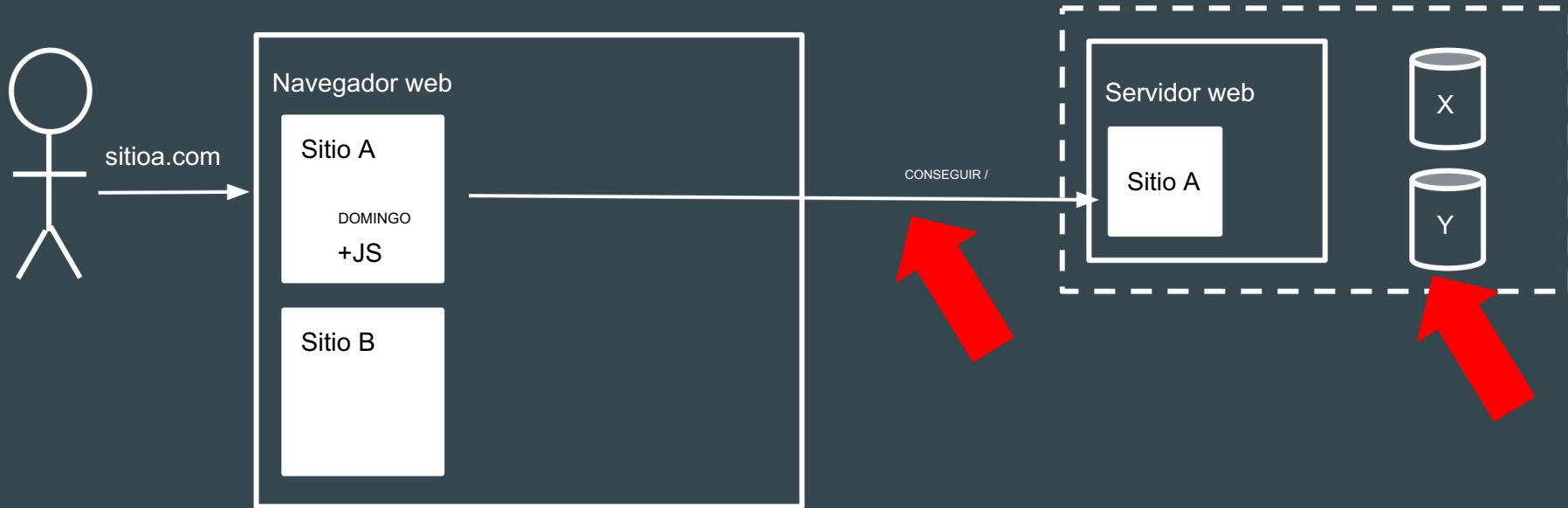
Prevención:

- Utilice buenas bibliotecas de autenticación
- Utilice MFA
- Aplicar contraseñas seguras •

Detectar y prevenir ataques de fuerza
bruta o de relleno

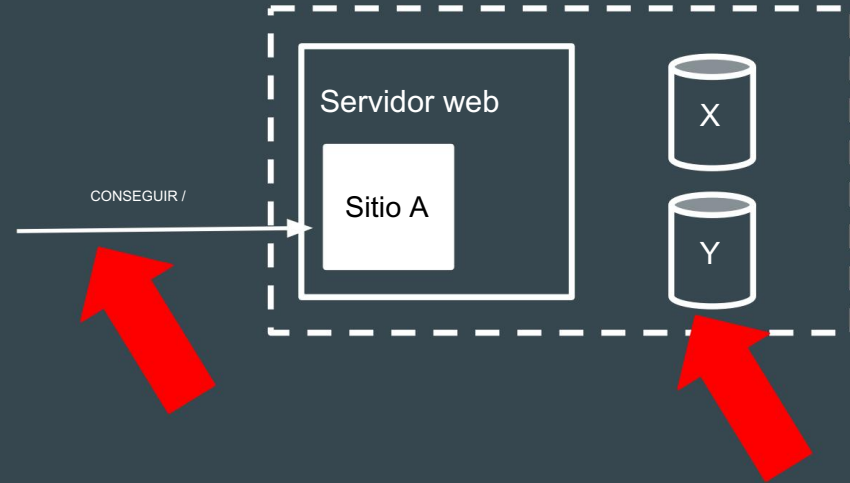
	Downstairs Auditorium (Room 098) Track Two: Technical
10:45	Improving Identity Management with W3C Verifiable Credentials <i>David Chadwick - University of Kent</i>

Exposición de datos confidenciales A3



Exposición de datos confidenciales A3

- Transferencia de datos en texto claro
- Almacenamiento no cifrado
- Cripto o claves débiles •
- Certificados no validados
- Exponer PII o tarjetas de crédito



Demostración de exposición de datos

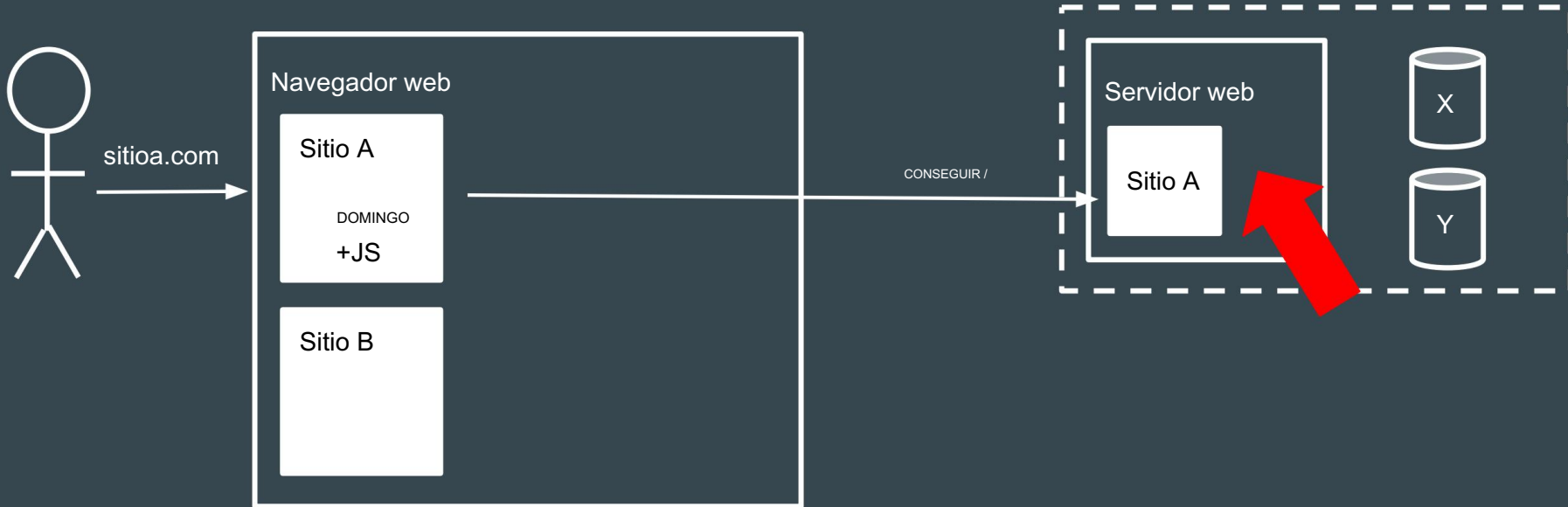
Exposición de datos confidenciales A3

Prevención:

- ¡No almacene datos a menos que sea necesario!
- Cifrar en reposo y en tránsito • Usar criptografía segura

	Downstairs Auditorium (Room 098) Track Two: Technical
13:30	Wyh Ranmdnoses Mattres <i>Frans Lategan - Aura Information Security</i>
16:55	A Recipe for Password Storage: Add Salt to Taste <i>Nick Malcolm - Aura Information Security</i>

Entidades externas XML A4 (XXE)

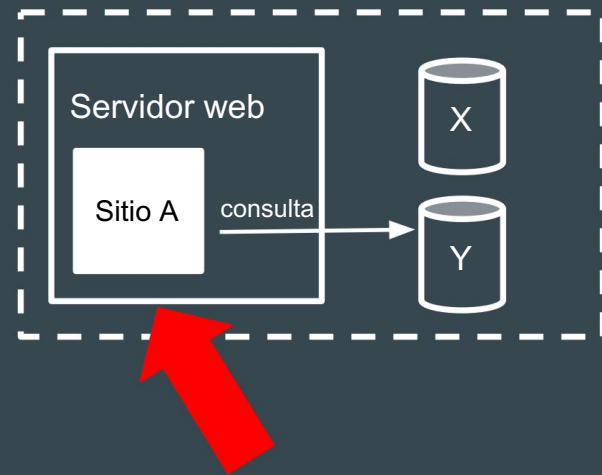


Entidades externas XML A4 (XXE)

La aplicación acepta XML y asume que es seguro.

```
<?xml versión="1.0" codificación="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo CUALQUIER >
<!ENTITY xxe SISTEMA "archivo:///etc/contraseña" >]>
<foo>&xxe;</foo>
```

Puede permitir el acceso a recursos confidenciales, la ejecución de comandos, el reconocimiento o provocar denegación de servicio.



Demostración XXE

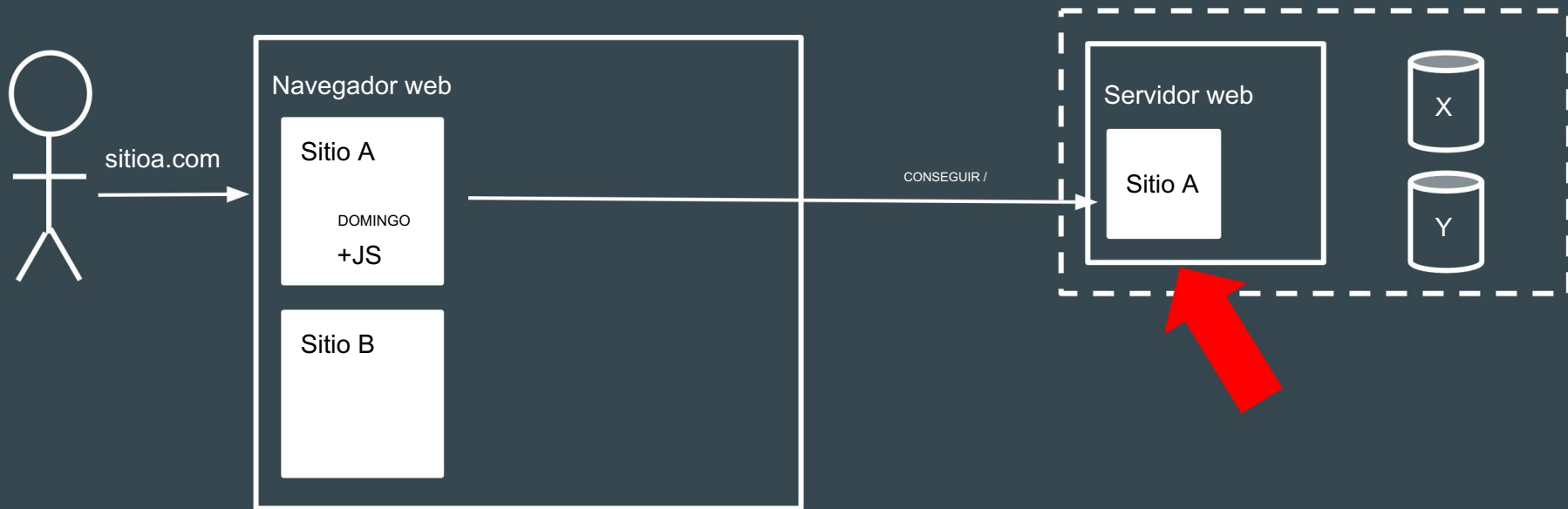
Entidades externas XML A4 (XXE)

Prevención:

- Evite XML
- Utilice bibliotecas modernas
y configúrelas bien.
- Validar XML

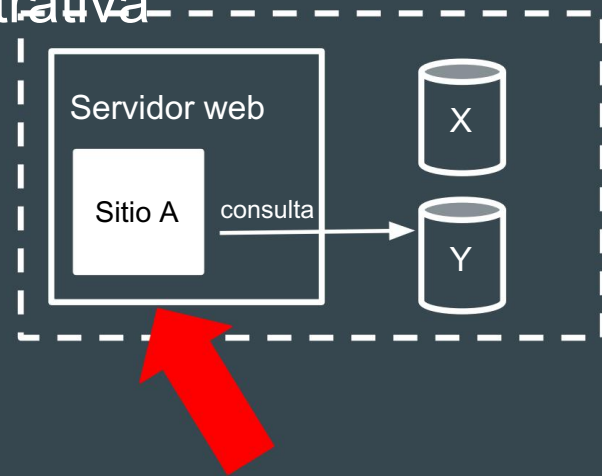
	Downstairs Auditorium (Room 098) Track Two: Technical
14:25	Web App Attacks of the Modern World <i>Karan Sharma</i>

Control de acceso roto A5



Control de acceso roto A5

- Acceder a páginas ocultas `http://site.com/admin/user-management`
- Elevarse a una cuenta administrativa
- Ver los datos de otras personas `http://site.com/user?id=7`
- Modificar cookies o tokens JWT



Control de acceso roto A5

Prevención:

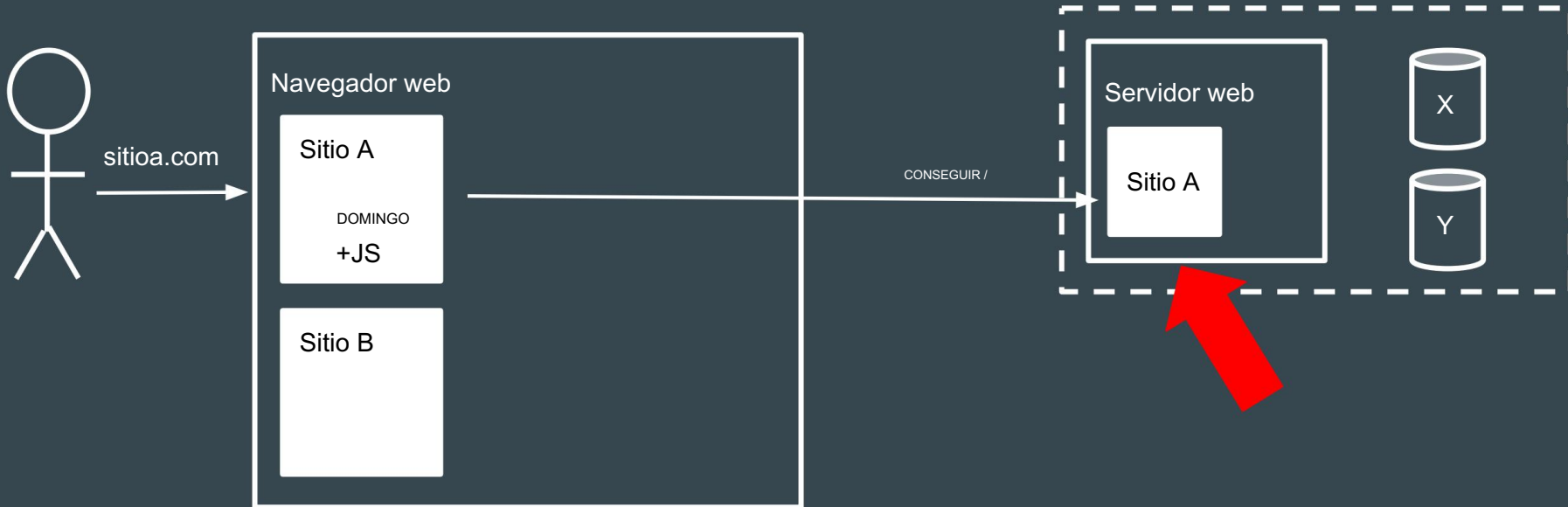
- Usar código o bibliotecas comprobadas •

Denegar el acceso de forma

predeterminada • Registrar fallas

y alertas • Limitar la velocidad de acceso a los recursos

Configuración incorrecta de seguridad A6

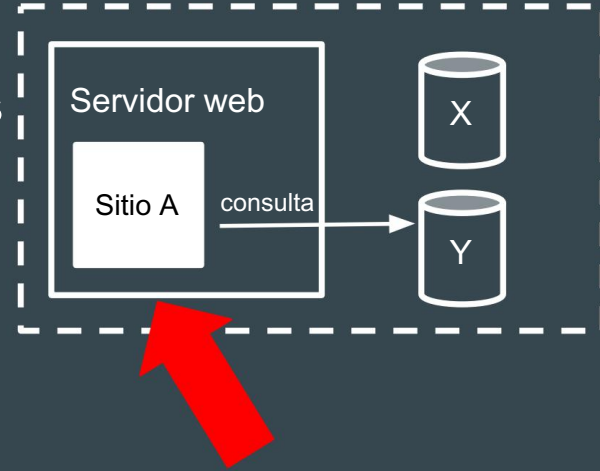


Configuración incorrecta de seguridad A6

- Funciones de seguridad no configuradas

correctamente • Funciones innecesarias
habilitadas • Cuentas predeterminadas no eliminadas

- Los mensajes de error exponen información
confidencial



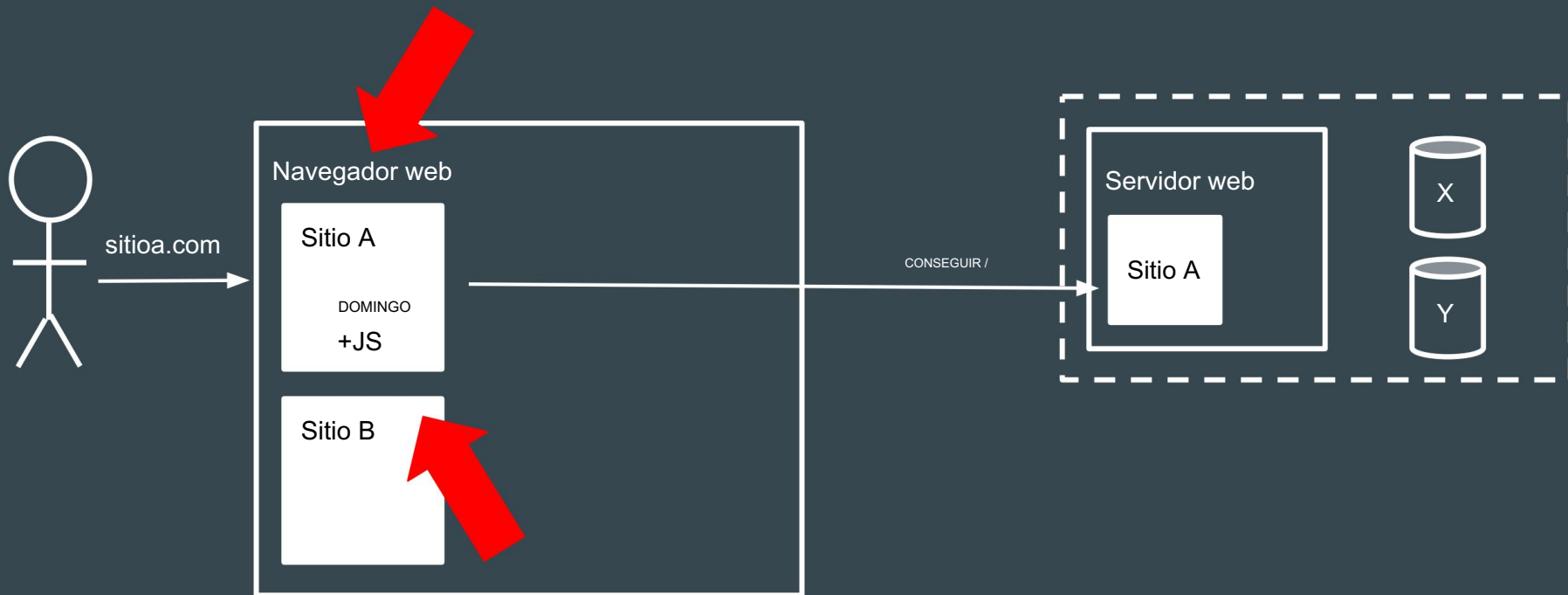
Configuración incorrecta de seguridad A6

Prevención:

- Tener un proceso de compilación repetible o "maestro dorado" • Deshabilitar todos los servicios no utilizados
- Usar herramientas para revisar la configuración

	Downstairs Auditorium (Room 098) Track Two: Technical
17:30	Self-Service SSH Certificates <i>Jeremy Stott</i>

Secuencias de comandos entre sitios A7 (XSS)



Secuencias de comandos entre sitios A7 (XSS)

HTML mezcla contenido, presentación y código en una sola cadena (HTML+CSS+JS)

Si un atacante puede alterar el DOM, puede hacer cualquier cosa que el usuario pueda hacer.

XSS se puede encontrar utilizando herramientas automatizadas.



Demostración XSS

Secuencias de comandos entre sitios A7 (XSS)

Prevención:

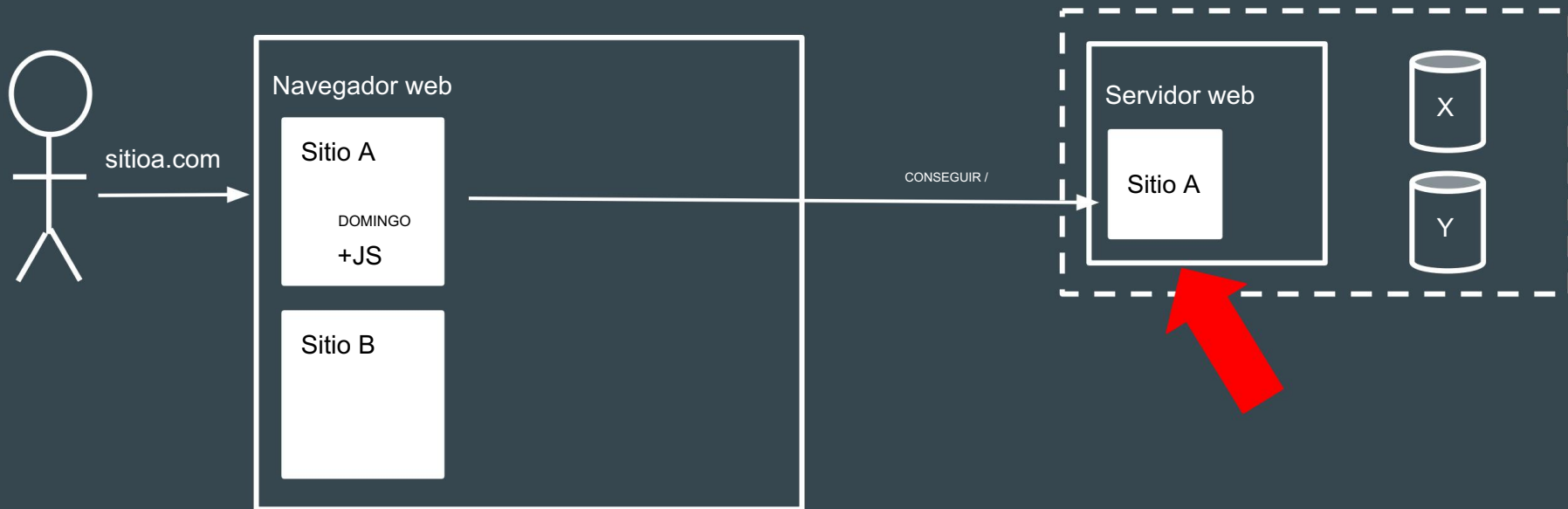
- Codificar todos los datos proporcionados por el usuario para hacerlos seguros.

Kirk <guión> => Kirk <guión>; • Usar

codificación apropiada para el contexto • Usar marcos

de plantillas que ensamblan HTML de forma segura • Usar política de seguridad de contenido

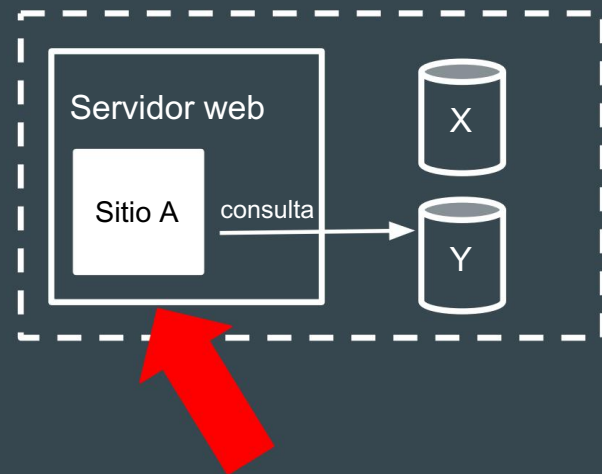
A8 Deserialización insegura



A8 Deserialización insegura

Los lenguajes de programación le permiten convertir un árbol de objetos en una cadena que puede enviarse al navegador.

Si deserializa datos que no son de confianza, puedes permitir que se creen objetos o que se ejecute código.



Demostración de deserialización

A8 Deserialización insegura

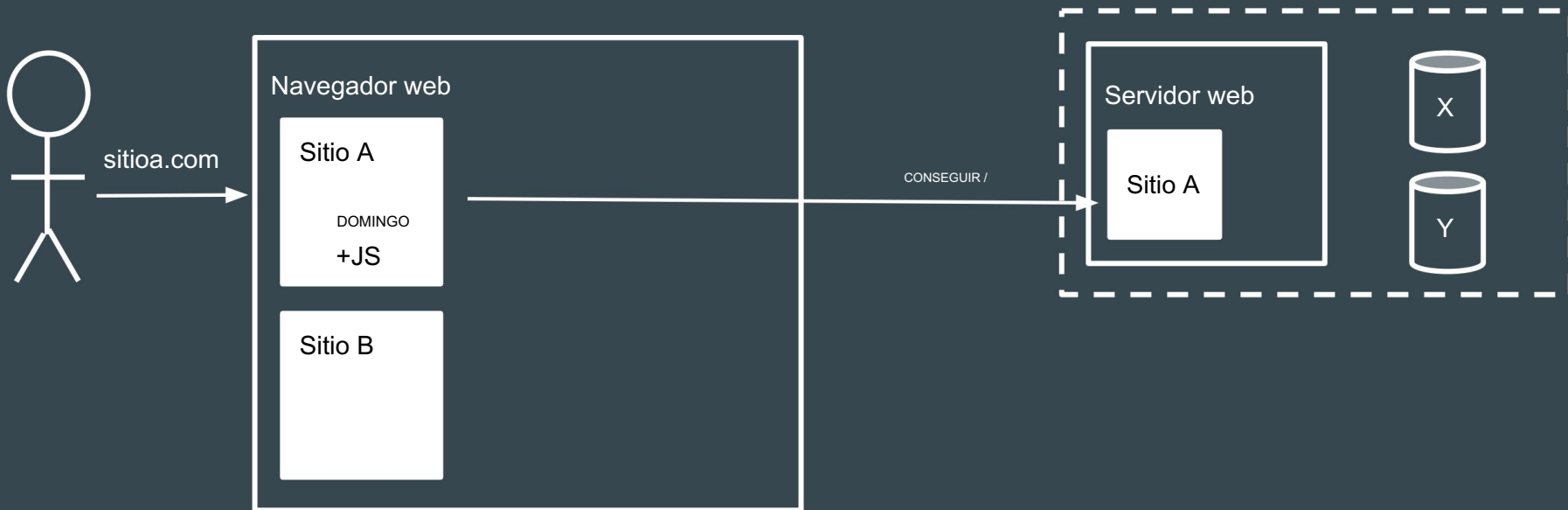
Prevención:

- Evite serializar y deserializar objetos
- Utilice firmas para detectar manipulaciones

Configure su biblioteca de forma segura

- Consulte la hoja de referencia de deserialización de OWASP

A9 Uso de componentes con vulnerabilidades conocidas



A9 Uso de componentes con vulnerabilidades conocidas

Las aplicaciones modernas contienen una gran cantidad de código de terceros.

Es difícil mantenerlo todo actualizado.

Los atacantes pueden enumerar las bibliotecas que utiliza y desarrollar exploits.

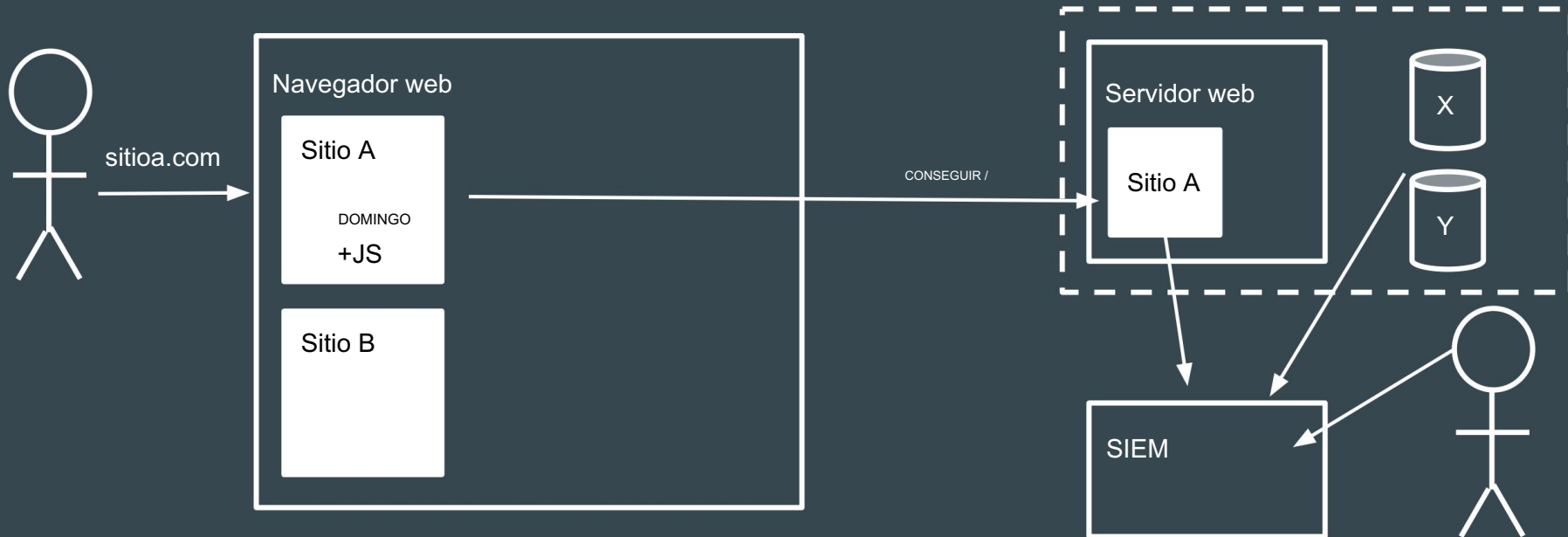
A9 Uso de componentes con vulnerabilidades conocidas

Prevención:

- Reducir las dependencias
- Gestión de parches
- Buscar anuncios desactualizados
componentes
- Presupuesto para el mantenimiento continuo de
todos los proyectos de software.

	Downstairs Auditorium (Room 098) Track Two: Technical
11:20	Scanning Your Container Images using Anchore <i>Vince Sesto - Foodstuffs North Island</i>

A10 Registro y monitoreo insuficientes



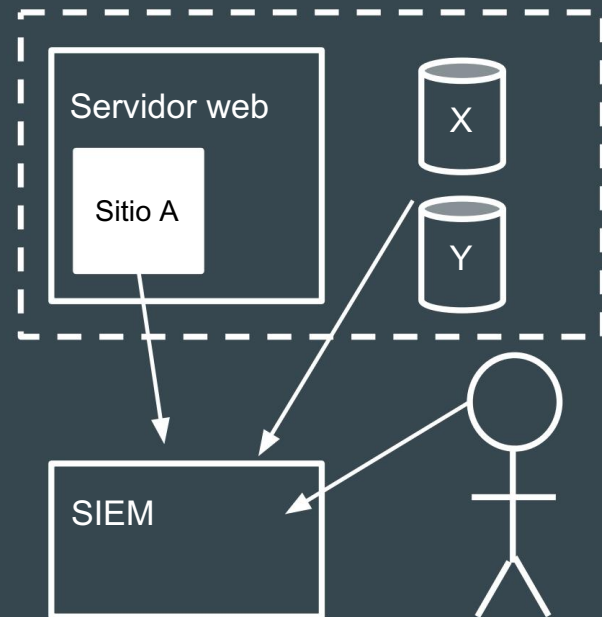
A10 Registro y monitoreo insuficientes

No puedes reaccionar ante ataques que no conoces.

Los registros son importantes para:

- Detectar incidentes •

Comprender qué sucedió • Demostrar
quién hizo algo



OWASP Top Ten 2017

A1 Inyección

Autenticación A2 rota

Exposición de datos confidenciales A3

Entidades externas XML A4 (XXE)

Control de acceso roto A5

Configuración incorrecta de seguridad A6

Secuencias de comandos entre sitios A7 (XSS)

A8 Deserialización insegura

A9 Uso de componentes con vulnerabilidades conocidas

A10 Registro y monitoreo insuficientes

Próximos pasos

Próximos pasos

- Asista a eventos de OWASP
- Busque los nombres de las diez categorías principales de OWASP y su marco.

Por ejemplo, “Protección C#

XSS” • Mire videos de YouTube o Pluralsight

- Utilice los términos cuando discuta errores con colegas •

Mantenga un registro de qué problemas le afectan más •

Vaya más allá del Top Ten

Introducción a la

Los diez mejores de OWASP



Kirk Jackson

RedShield

kirk@pageofwords.com

<http://hack-ed.com>

@kirkj

OWASP

Nueva Zelanda [https://](https://www.meetup.com/)

www.meetup.com/

OWASP-Wellington/

www.owasp.org.nz @

Grabaciones:

<https://goo.gl/a2VSG2>