

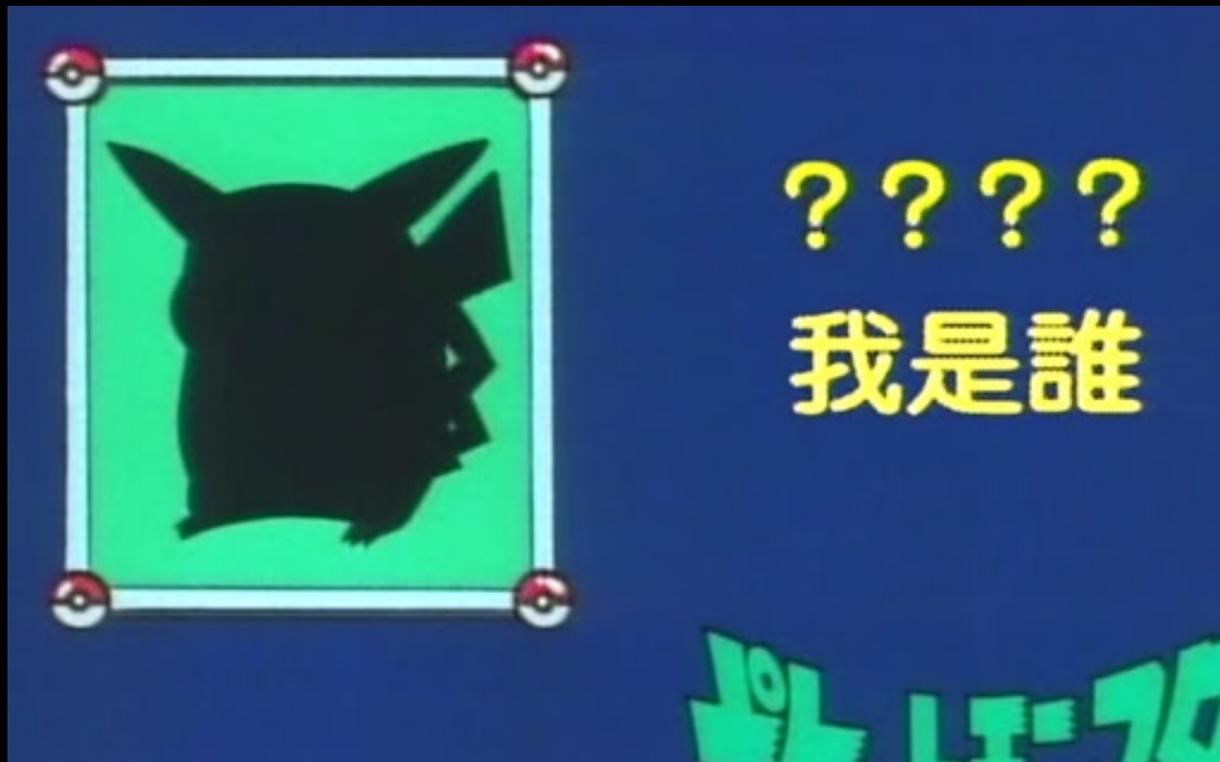


网络安全创新大会
Cyber Security Innovation Summit

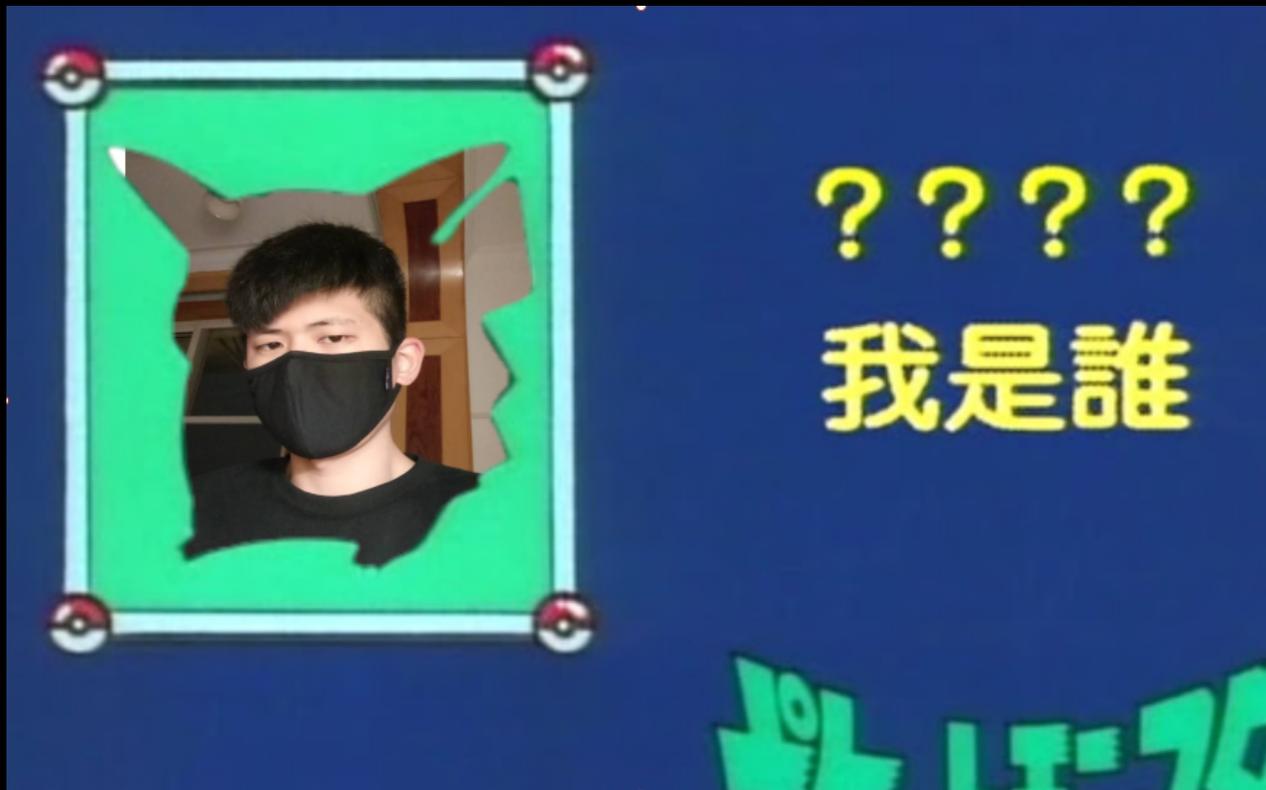
我的一键 getshell 代码开发之路

Ali0th 斗象科技资深安全研究员

一、我是谁



一、我是谁



Ali0th

斗象科技资深安全研究员，熟悉领域有自动化渗透、流量安全检测、代码审计、区块链等。

GitHub

github.com/Martin2877

一、我是谁



Ali0th



二、我的安全开发之路

挖洞

鸿沟

懒

二、我的安全开发之路

挖洞：

google
dork

intext :hacked by
intitle: "web client: login"
inurl:/?op=register
...



PoC工具

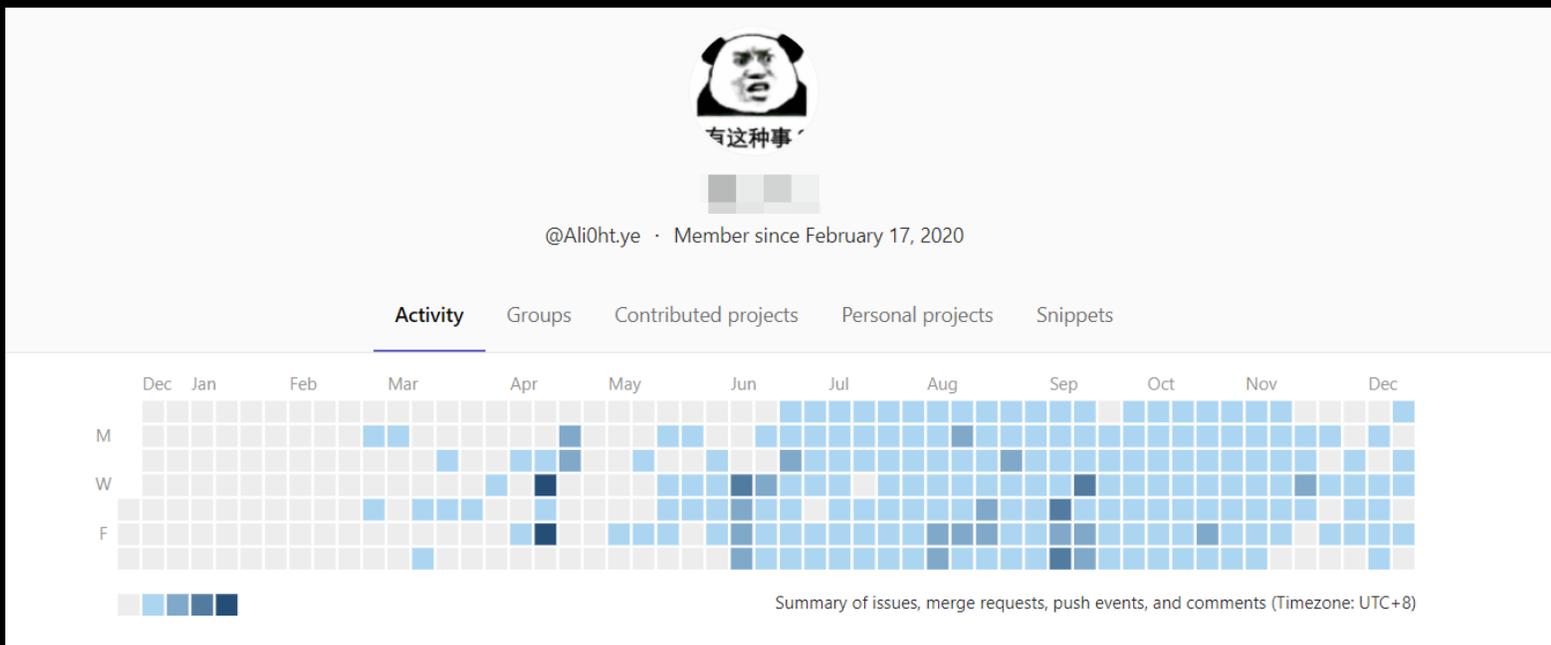
pocsuite
osprey
poc-T
...



(半)自动化渗透

AWVS
Appscan
Metasploit
...
GourdScanV2
w13scan
Xray
...

二、我的安全开发之路



探讨：

- 1、怎么样的代码才算一份合格的代码？
- 2、如何实现自动化渗透？
- 3、如何避免重复造轮子？

三、合格代码？

写一份好代码的秘籍是什么？

Class

三、合格代码？

```
def ...  
def ...  
def ...
```

```
class ...  
class ...  
class...
```

bad



good



三、合格代码？

复杂

```
for dp in range(1,self.depth):  
    urls_new = self.comp(set(self.flat_list(list(map(self.url_html,urls_new))))  
    if len(urls_new) == 0:break  
    if verbose : print("[result]urls_new:{0},self.urls:{1}".format(urls_new,self.urls))  
    self.urls.extend(urls_new)
```

bad



简单

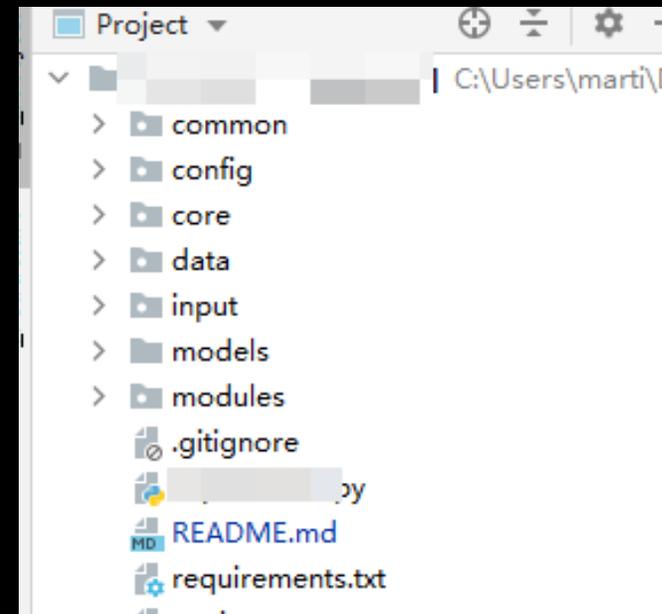
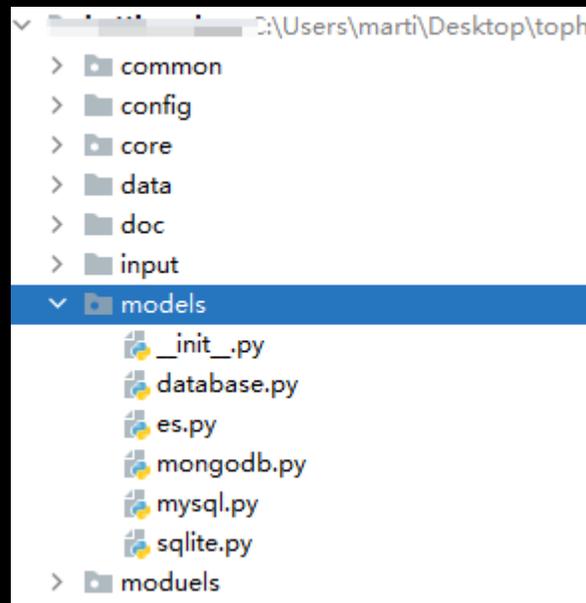
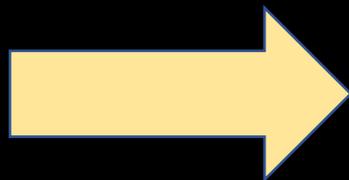
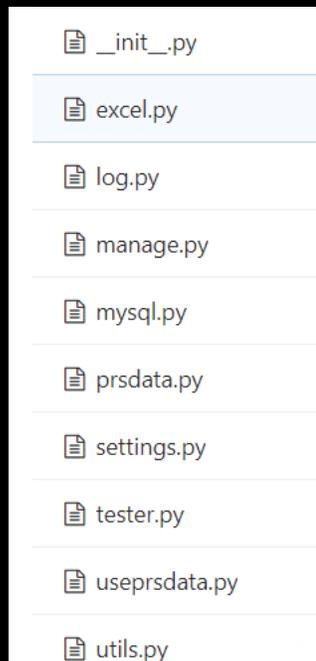
```
for a in list:  
    Fun(a)
```

good



三、合格代码？

架构



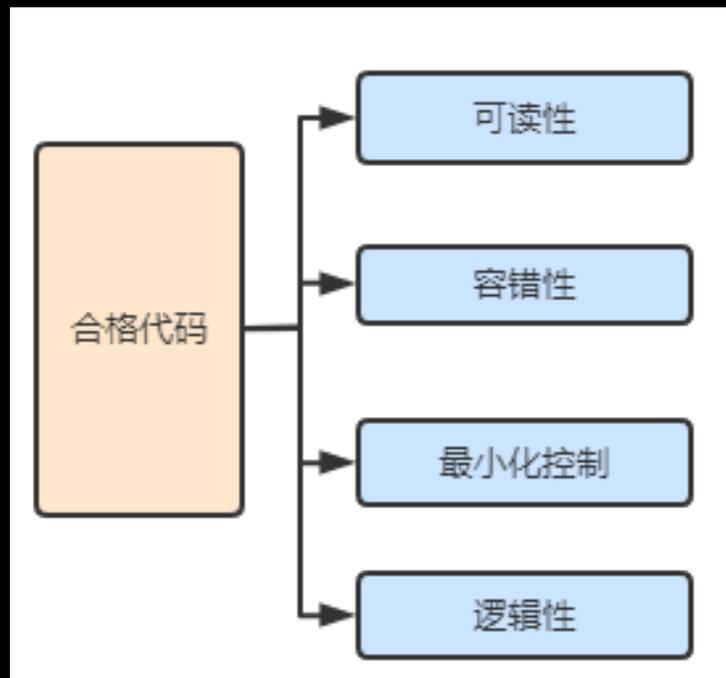
bad



good



1、怎么样的代码才算一份合格的代码？



如何实现自动化渗透和避免重复造轮子？

嫖

白嫖大军，使命必达



白嫖大军，使命必达



白嫖大军，使命必达



白嫖大军，使命必达



白嫖大军，使命必达



如何实现？

方法论：STP

Target:需求

Situation:现状

Solution:解决方案

如何实现？

Target

一键 Getshell

Situation

单类工具多
简单集成
半自动

Solution

- a 收集目标
- b 资产识别、爬虫
- c 针对框架打 poc
- d 漏洞扫描
- e 进一步深入利用漏洞
- f 写 shell，反弹 shell
- g 整体自动化
- h 操作简单
- i 体积小

方法论：STP

四、自动化渗透



快了, 已经在做了

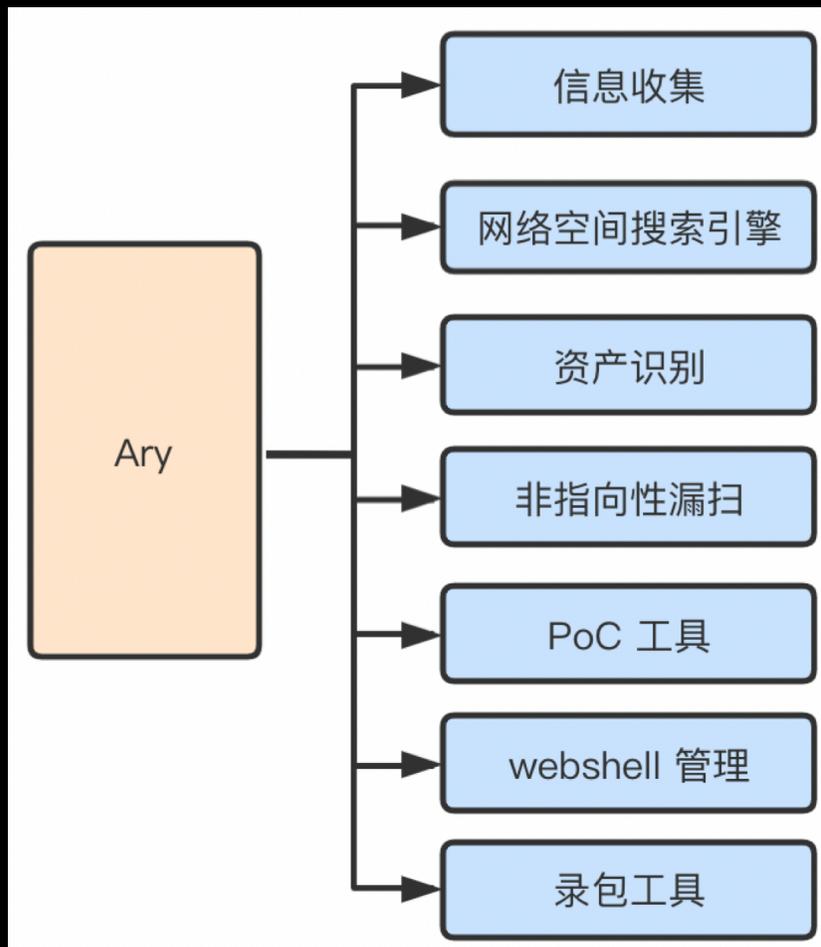


进度:0%

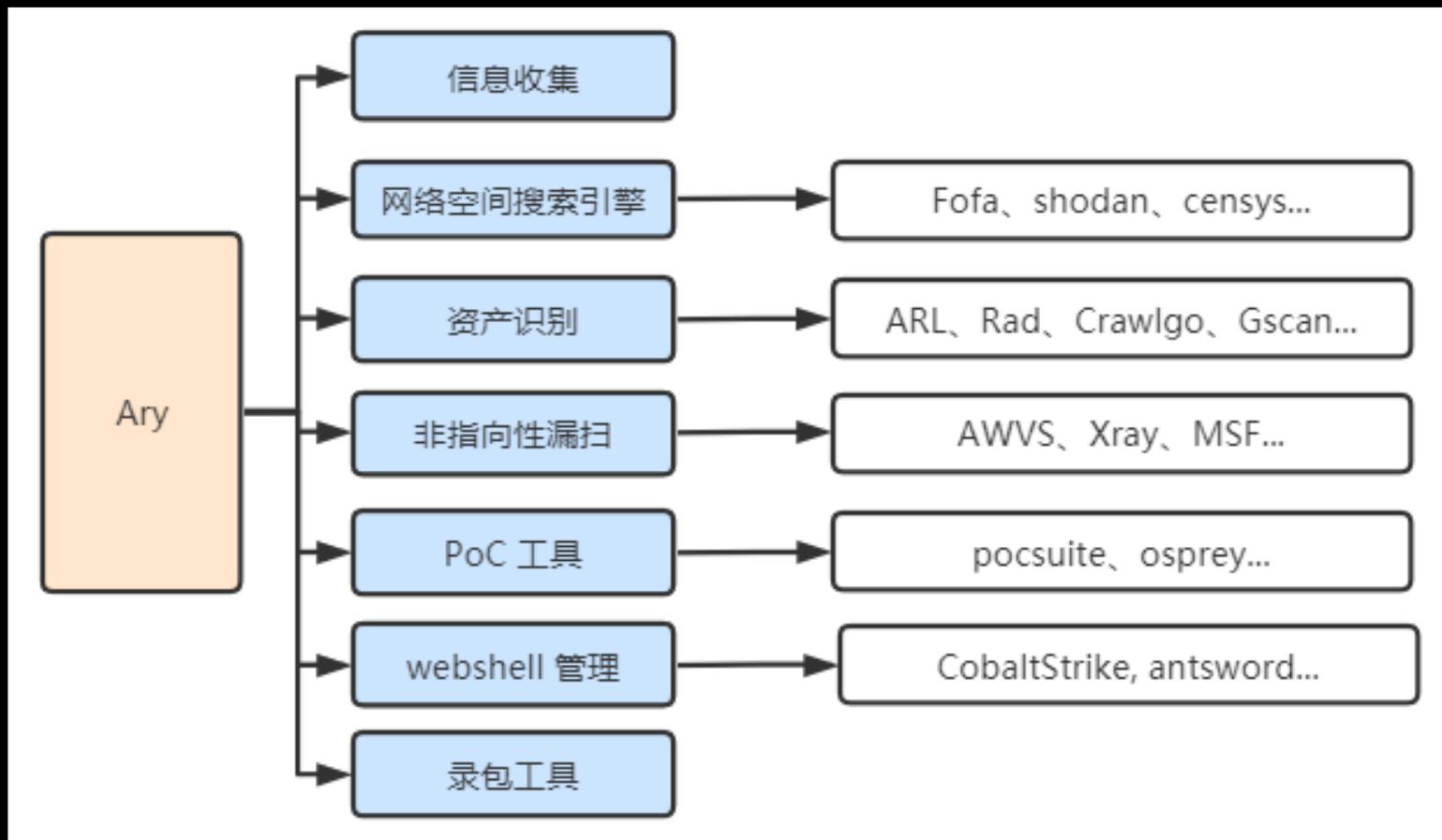
(可以理解 为 Armory 武器库缩写)

Ary 是一个集成类工具, 主要用于调用各种安全工具, 从而形成便捷的一键式渗透。

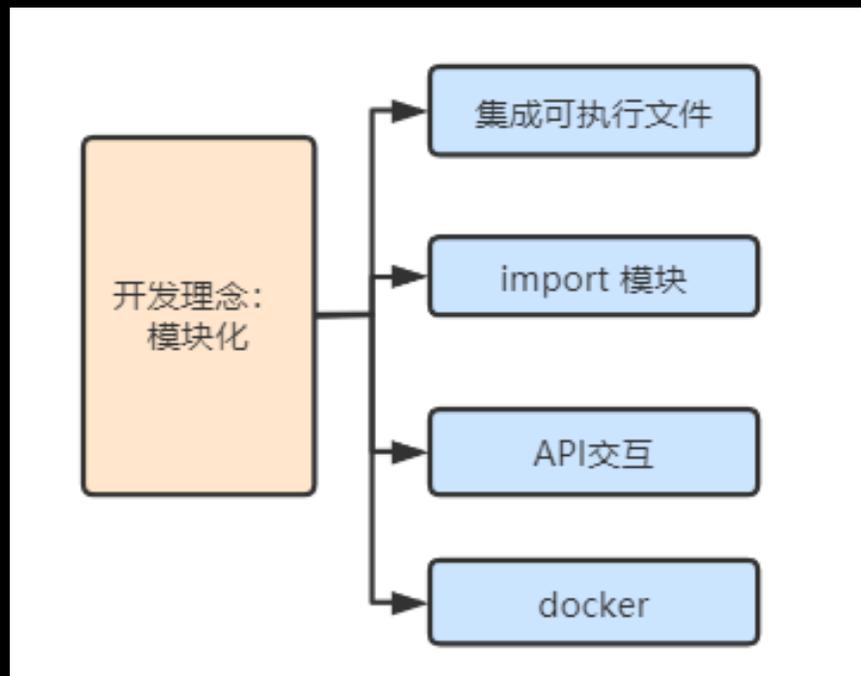
ARY



ARY



ARY



越简单越好



ARY

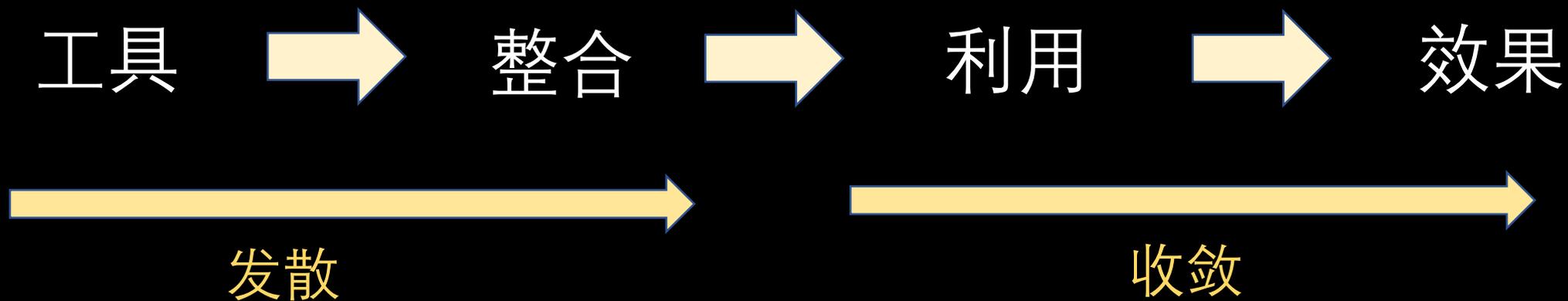
```
./ary --netsearch --engine shodan --keyword dedecms  
./ary --assertscan --engine rad --url vulnweb.com -v  
./ary --vulnscan --engine xray --url xx.xx.xx.xx -v  
./ary --pocscan --input redis.txt --poc redis -v
```

越简单越好



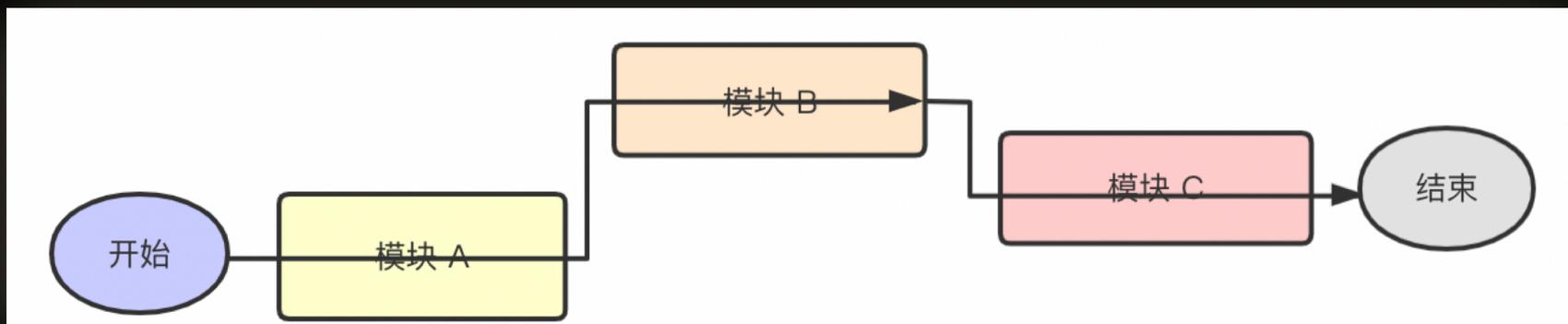
四、自动化渗透

ARY

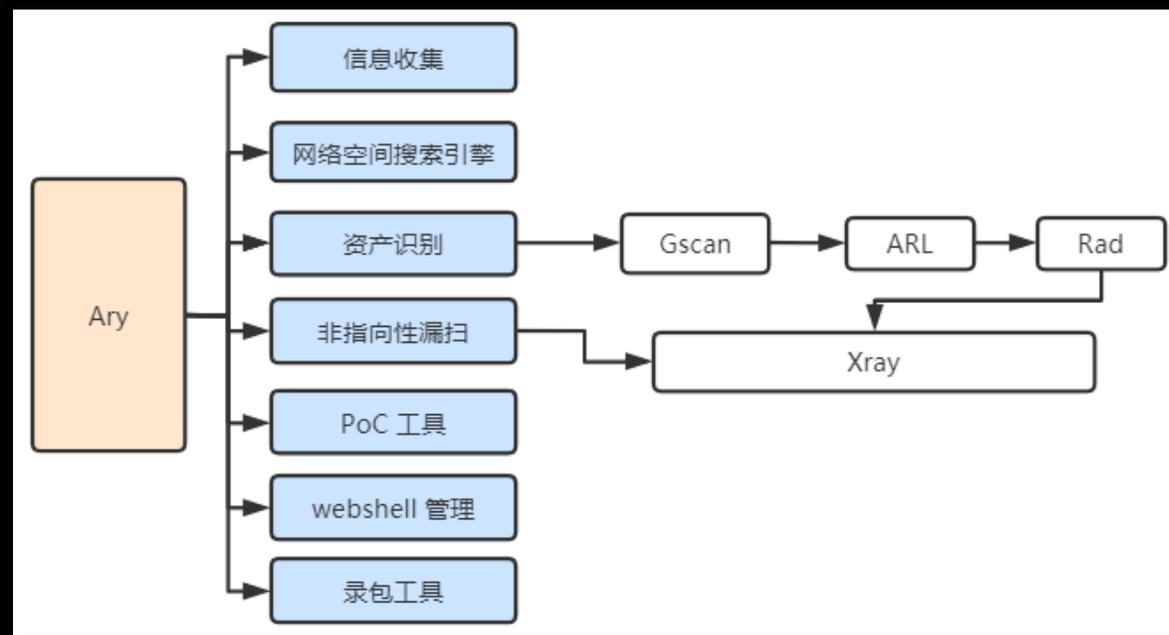
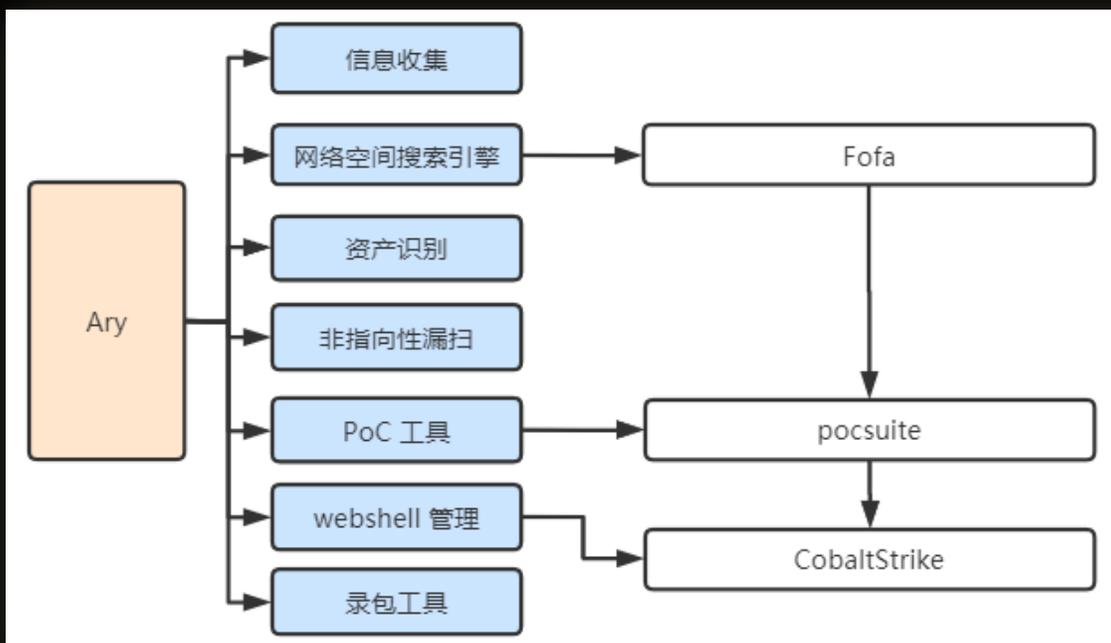


ARY

整合 → 归一化 → 逻辑化



ARY : 自动挖洞思路 -> 执行流



ARY : 自动挖洞思路 -> 执行流

Step 1 : 收集网站

```
./ary --netsearch --engine shodan --keyword reids -v
```

Step 2 : 打 poc

```
./ary --pocscan --keyword redis --poc redis -v
```

ARY : 自动挖洞思路 -> 执行流

Step 1 : 启动 xray 后台

```
./ary --vulnscan --engine xray --port 7778 --background -v
```

Step 2 : 启动爬虫

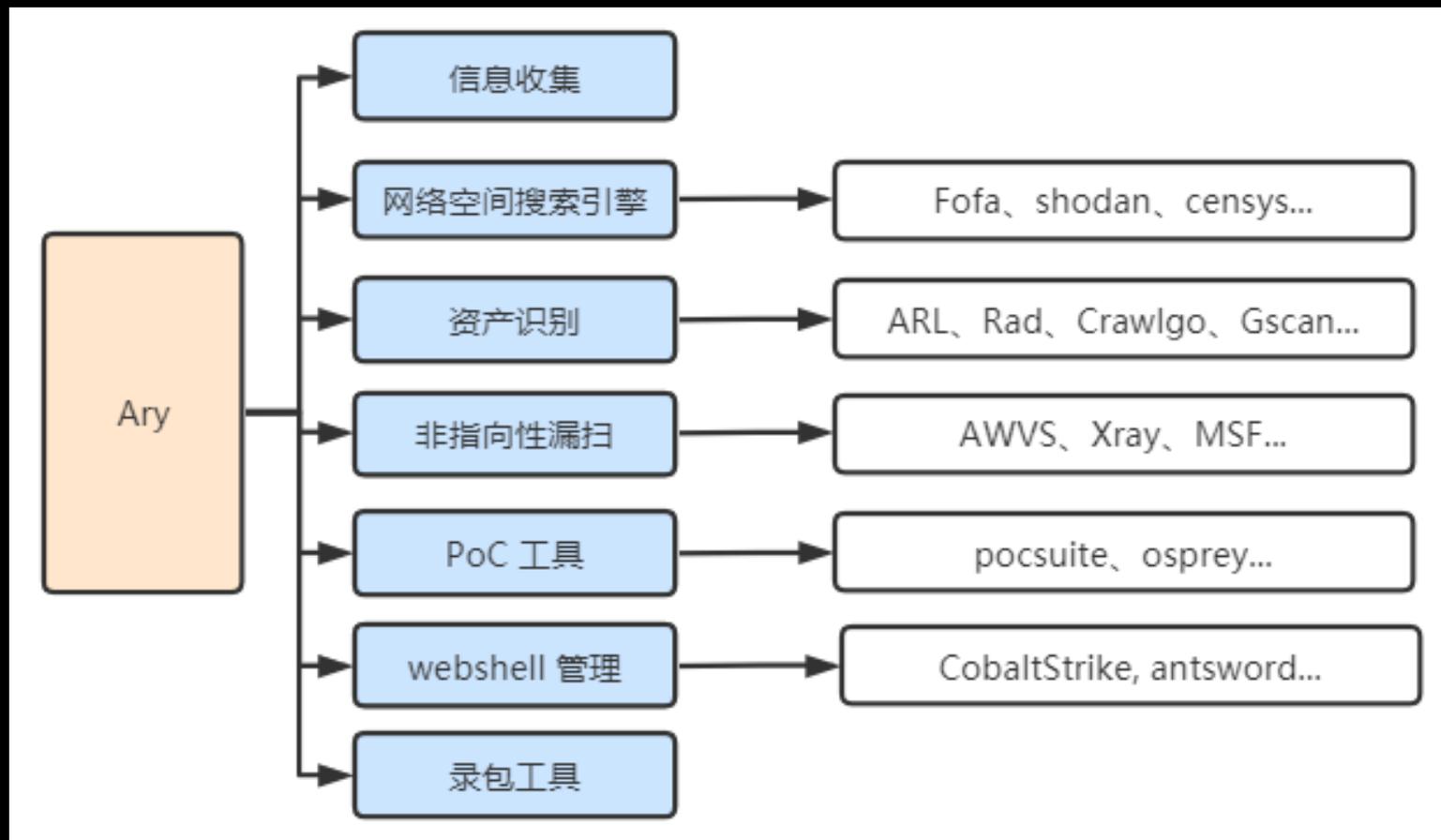
```
./ary --assertscan --engine crawlergo --url testphp.vulnweb.com  
--passive --port 7778
```

ARY : 自动挖洞思路 -> 执行流

```
streams.yaml 321 Bytes
1 redis:
2   name: redis 未授权访问漏洞
3   steps:
4     - netsearch: True
5       engine: shodan
6       keyword: redis
7       limit: 10
8     - pocscan: True
9       keyword: redis
10      poc: redis.py
11 vulnweb:
12   name: test testphp.vulnweb.com
13   steps:
14     - vulnscan: True
15       engine: xray
16       url: http://testphp.vulnweb.com/
17       passive: True
```

```
./ary -v --stream --input streams.yaml --keyword "redis 未授权访问漏洞"
```

ARY



扩展性+自定义

ARY

Ary 是一个集成类工具，主要用于调用各种安全工具，从而形成便捷的一键式渗透。

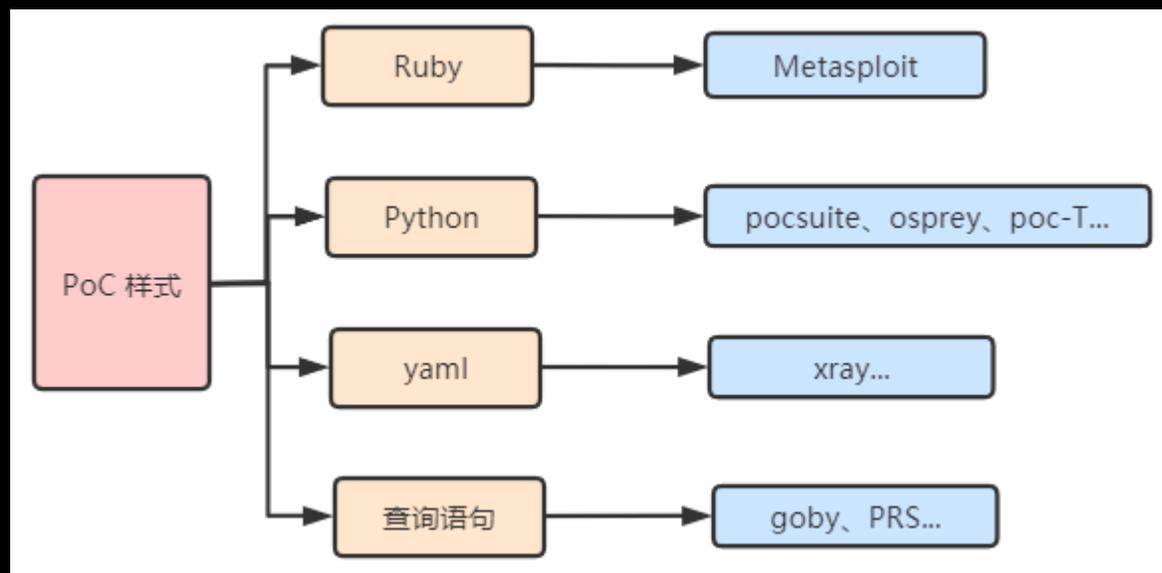
Github : <https://github.com/TeraSecTeam/ary>



```
version : 1.5.0
author : Ali0th / github.com/Martin2877 / martin2877@foxmail.com
descript:
This is a tool to use many other tools.
```

```
Statement:
This tool is only for learning and testing. It is strictly prohibited to use
```

PoC收集怎么办？



PoC收集怎么办？

自动收集：<https://github.com/TeraSecTeam/poc-collection>

```
5127 lines (5127 sloc) | 1000 KB
Raw Blame
We can't make this file beautiful and searchable because it's too large.
1 file_updated_at,filename,fileurl
2 2020-07-12 20:05:24,SMB_CVE-2020-0796.py,https://github.com/TeraSecTeam/poc-collection/blob/30d90f2d65765fb6182e3a10e347ead13/win/SMB_CVE-2020-0796.py
3 2020-11-26 13:27:01,sprint_cloud_config_cve_2020_5410.py,https://github.com/TeraSecTeam/poc-collection/blob/01ad9c524b94f0e5f3f0f/windows/CVE-2020-0796/attack/CVE-2020-0796.py
4 2020-09-06 18:50:06,cve-2020-5902.py,https://github.com/TeraSecTeam/poc-collection/blob/77c79dce66d0402f85a1c1a4e8d78d4ac613/F5/cve-2020-5902.py
5 2020-08-01 16:44:18,cve_2020_9374.py,https://github.com/TeraSecTeam/poc-collection/blob/cb2b3ce8cd9147b237c5b7/%E5%B7%B2%E9%AA%8C%E8%AF%81poc/exp/routers/TP-Link/2/cve_2020_9374.py
6 2020-10-06 00:34:11,CVE-2020-0796.py,https://github.com/TeraSecTeam/poc-collection/blob/8ad9c524b94f0e5f3f0f/windows/CVE-2020-0796/attack/CVE-2020-0796.py
7 2020-11-09 10:33:03,CVE-2020-1938.py,https://github.com/TeraSecTeam/poc-collection/blob/d9b9bfd2ad93f98c13f498214e1e3fee2/script/tomcat/CVE-2020-1938.py
8 2020-11-09 10:33:03,CVE-2020-0796.py,https://github.com/TeraSecTeam/poc-collection/blob/d9b9bfd2ad93f98c13f498214e1e3fee2/script/smb/CVE-2020-0796.py
9 2020-11-17 14:01:36,CVE-2020-1472.py,https://github.com/TeraSecTeam/poc-collection/blob/f196d0e697f3bb05f2c21d57c454c497d/exp/%E7%AC%AC%E4%BA%8C%E7%89%88/CVE-2020-1472.py
10 2020-11-17 14:01:36,CVE-2020-1472.py,https://github.com/TeraSecTeam/poc-collection/blob/f196d0e697f3bb05f2c21d57c454c497d/exp/%E7%AC%AC%E5%9B%9B%E7%89%88/CVE-2020-1472.py
11 2020-11-17 14:01:36,CVE-2020-1472.spec,https://github.com/TeraSecTeam/poc-collection/blob/9ff196d0e697f3bb05f2c21d57c454c497d/exp/%E7%AC%AC%E4%B8%89%E7%89%88/CVE-2020-1472.spec
12 2020-09-03 06:21:42,CVE-2020-17506-artica-rce.py,https://github.com/TeraSecTeam/poc-collection/blob/9ff196d0e697f3bb05f2c21d57c454c497d/exp/%E7%AC%AC%E4%B8%89%E7%89%88/CVE-2020-17506-artica-rce.py
13 2020-11-04 16:55:55,CVE-2020-2551.py,https://github.com/TeraSecTeam/poc-collection/blob/41398456fb5edd6bec15eccdd24318ebc0f/app/pocs/CVE-2020-2551.py
14 2020-11-09 10:33:03,CVE-2020-2551.py,https://github.com/TeraSecTeam/poc-collection/blob/f4d9b9bfd2ad93f98c13f498214e1e3fee2/script/weblogic/CVE-2020-2551.py
15 2020-11-04 16:55:55,CVE-2020-2883.py,https://github.com/TeraSecTeam/poc-collection/blob/41398456fb5edd6bec15eccdd24318ebc0f/app/pocs/CVE-2020-2883.py
16 2020-11-09 10:33:03,CVE-2020-5405.py,https://github.com/TeraSecTeam/poc-collection/blob/f4d9b9bfd2ad93f98c13f498214e1e3fee2/script/spring/CVE-2020-5405.py
17 2020-08-23 09:03:22,CVE-2020-8816.py,https://github.com/TeraSecTeam/poc-collection/blob/2d2b8a11cf520f76a193ed804945882dbe0f/CVE-2020-8816.py
```

五、总结



木禾 

北马里亚纳群岛



欢迎交流

Ali0th

Github : github.com/Martin2877

Email : martin2877@foxmail.com

TCC Team长期招聘,简历砸我

回顾

两个技巧，两个方法，两个工具

Class

嫖

方法论：STP

集成化->归一化->逻辑化

ARY

[poc-collection](#)



木禾 

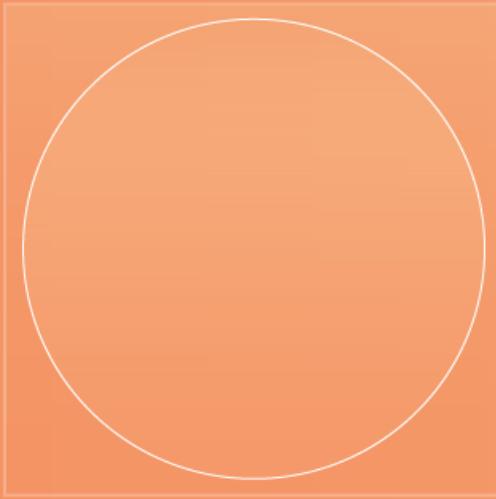
北马利亚纳群岛



Ali0th

Github : github.com/Martin2877

Email : martin2877@foxmail.com



THANKS

