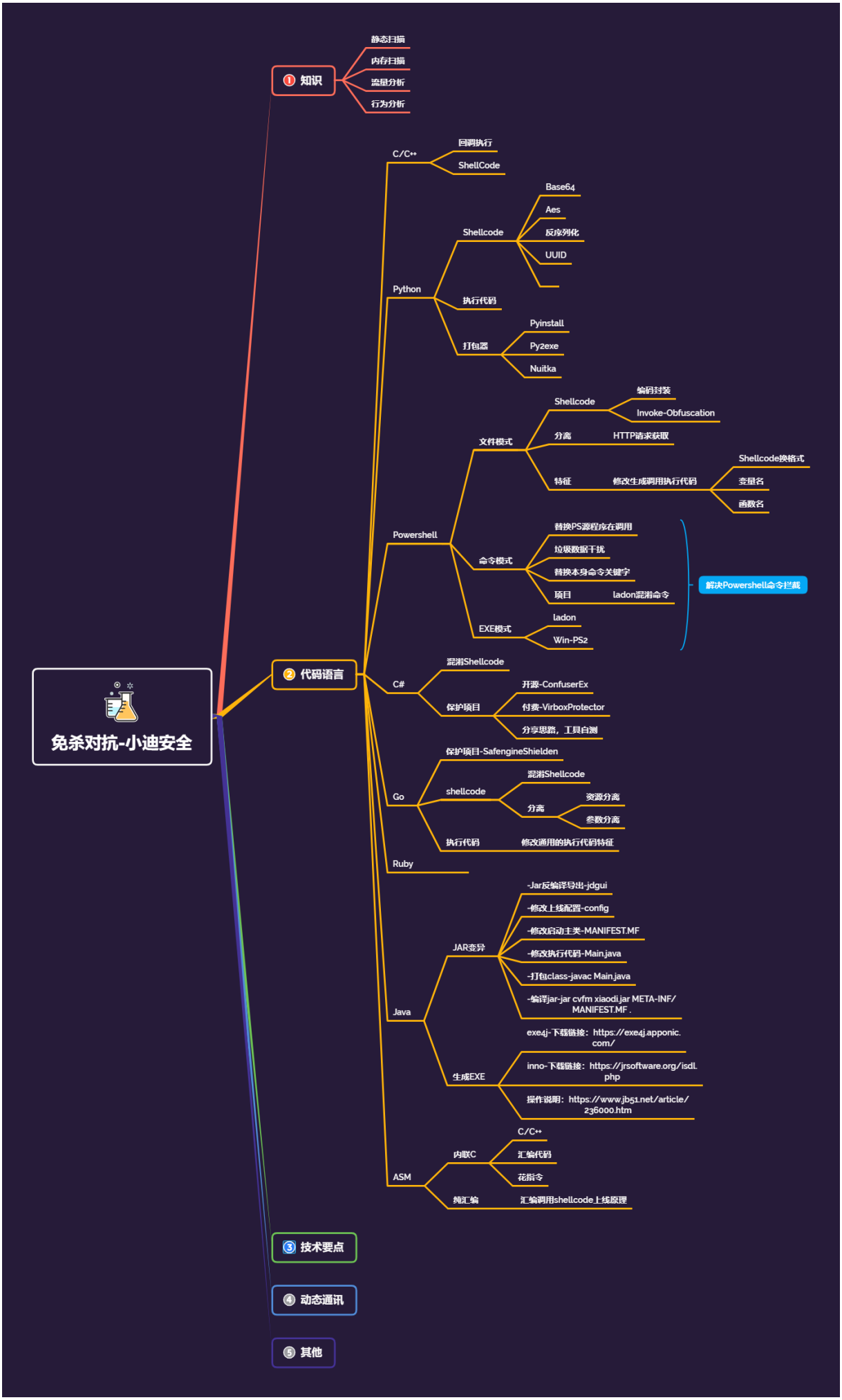
免杀对抗-内存加载&API 封装&UUID 标识&MAC 地址&IPV4 地址&各语言


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
JAR变异
-Jar反编译导出-jdgui
-修改上线配置-config
-修改启动主类-MANIFEST.MF
-修改执行代码-Main.java
-打包class-javac Main.java
-编译jar-jar cvfm xiaodi.jar META-INF/MANIFEST.MF .
生成EXE
exe4j-下载链接：https://exe4j.apponic.com/
inno-下载链接：https://jrsoftware.org/isdl.php
操作说明：https://www.jb51.net/article/236000.htm
ASM
内联C
C/C++
汇编代码
花指令
纯汇编
汇编调用shellcode上线原理
3 技术要点
4 动态通讯
5 其他

#知识点：

1、内存加载-UUID 地址-ShellCode 转换

2、内存加载-MAC 地址-ShellCode 转换

3、内存加载-IPV4 地址-ShellCode 转换

#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，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：

- 内存加载-UUID 地址-ShellCode 转换
- 内存加载-MAC 地址-ShellCode 转换
- 内存加载-IPV4 地址-ShellCode 转换

文章参考：

https://www.anquanke.com/post/id/262666

#内存加载-UUID 方式-ShellCode 转换

通用唯一识别码(UUID)，是用于计算机体系中以识别信息数目的一个 128 位标识符，根据标准方法生成，不依赖中央机构的注册和分配，UUID 具有唯一性。

演示加载语言：C++ C# Python2 Go

#内存加载-MAC 地址-ShellCode 转换

MAC 地址也叫物理地址、硬件地址，由网络设备制造商生产时烧录在网卡的 EPROM 一种闪存芯片，通常可以通过程序擦写。IP 地址与 MAC 地址在计算机里都是以二进制表示的，IP 地址是 32 位的，而 MAC 地址则是 48 位（6 个字节）的。

演示加载语言：Python2 Go

#内存加载-IPV4 方式-ShellCode 转换

IPv4 是一种无连接的协议，操作在使用分组交换的链路层（如以太网）上。此协议会尽最大努力交付数据包，意即它不保证任何数据包均能送达目的地，也不保证所有数据包均按照正确的顺序无重复地到达。IPv4 使用 32 位（4 字节）地址。

演示加载语言：Go

涉及资源：

补充：涉及录像课件资源软件包资料等下载地址
