# VizAsm

- Aufgabenstellung:
  - Analyse Mach-O Binary (Mac und iOS)
  - Durchführen eines Security-Audit (automatisiert)
  - Visualisierung von sicherheitskritischen Stellen
  - Erweiterbarkeit durch eigene Filter → Filter Api
- Benutzte Werkzeuge / Sprachen:
  - Eclipse / Python
  - Xcode / Objective-C
  - Hopper Disassembler / Assembler (x86, x86_64, arm)

# Objective-C

- Superset von C
- Objektorientiert
- Method Calls
  - Java
    - objPtr.method(param1, param2)
  - Objective-C:
    - [objPtr method:param1 param2Name:param2]
    - id objc_msgSend(id theReceiver, SEL theSelector, …)

      → objc_msgSend(objPtr, @selector(method:param2Name:), param1, param2)

# Disassembly (x86_64)

- (void)method:(int)param1 paramname2:(int)param2 {

[self method:param1 paramname2:0];} // param1 und 0 als Argumente

```
methImpl_HelloWorldClass_method_paramname2_:
      push        rbp
      mov         rbp, rsp
      sub         rsp, 0x20
      mov         eax, 0x0
      mov         qword [ss:rbp-0x20+var_24], rdi
      mov         qword [ss:rbp-0x20+var_16], rsi
      mov         dword [ss:rbp-0x20+var_12], edx
      mov         dword [ss:rbp-0x20+var_8], ecx
      mov         rsi, qword [ss:rbp-0x20+var_24]
      mov         edx, dword [ss:rbp-0x20+var_12]
      mov         rdi, qword [ds:objc_sel_method_paramname2_] ; @selector(method:paramname2:)
      mov         qword [ss:rbp-0x20+var_0], rdi
      mov         rdi, rsi
      mov         rsi, qword [ss:rbp-0x20+var_0]
      mov         ecx, eax
      call        imp___stubs__objc_msgSend
      add         rsp, 0x20
      pop         rbp
      ret
```

# Disassembly (x86_64) (2)

- (void)method:(int)param1 paramname2:(int)param2 {

[self method:param1 paramname2:0];} // param1 und 0 als Argumente

```
; Method: -[HelloWorldClass method:paramname2:]
; [HelloWorldClass method:arg1 paramname2:0]
;
; rdi = ObjcClass(HelloWorldClass)
; rdx = arg1
; rcx = arg2
; rsi = Selector(method:paramname2:)
    methImpl_HelloWorldClass_method_paramname2_:
            push        rbp
            mov         rbp, rsp                        ; rbp = rsp
            sub         rsp, 0x20
            mov         eax, 0x0                        ; eax = 0
            mov         qword [ss:rbp-0x20+var_24], rdi ; var_24 = ObjcClass(HelloWorldClass)
            mov         qword [ss:rbp-0x20+var_16], rsi ; var_16 = Selector(method:paramname2:)
            mov         dword [ss:rbp-0x20+var_12], edx ; var_12 = arg1
            mov         dword [ss:rbp-0x20+var_8], ecx ; var_8 = arg2
            mov         rsi, qword [ss:rbp-0x20+var_24] ; rsi = ObjcClass(HelloWorldClass)
            mov         edx, dword [ss:rbp-0x20+var_12] ; edx = arg1
            mov         rdi, qword [ds:objc_sel_method_paramname2_] ; rdi = Selector(method:paramname2:)
            mov         qword [ss:rbp-0x20+var_0], rdi ; var_0 = Selector(method:paramname2:)
            mov         rdi, rsi                        ; rdi = ObjcClass(HelloWorldClass)
            mov         rsi, qword [ss:rbp-0x20+var_0] ; rsi = Selector(method:paramname2:)
            mov         ecx, eax                        ; ecx = 0
; [HelloWorldClass method:arg1 paramname2:0]
            call        imp___stubs__objc_msgSend       ; rax = [HelloWorldClass method:arg1 paramname2:0]
            add         rsp, 0x20
            pop         rbp
            ret
```

# Security Audit (Filter)

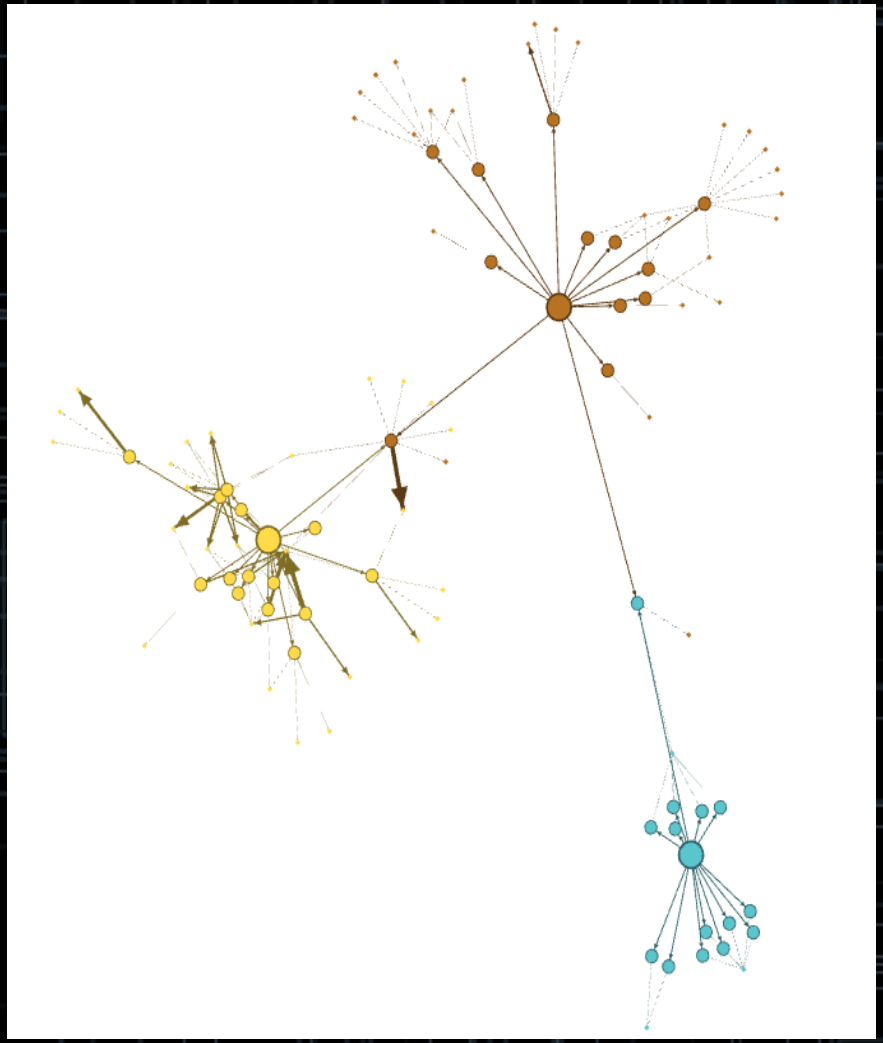
- Network
  - UntrustedSSLCertsFilter
  - CFStreamSSLLevelFilter
  - NSStreamSSLLevelFilter
  - CookiePolicyFilter
- NSUserDefaultsFilter
- RandomFuncFilter
- FormatStringFilter
- SQLiteFilter

# Security Audit (Filter)

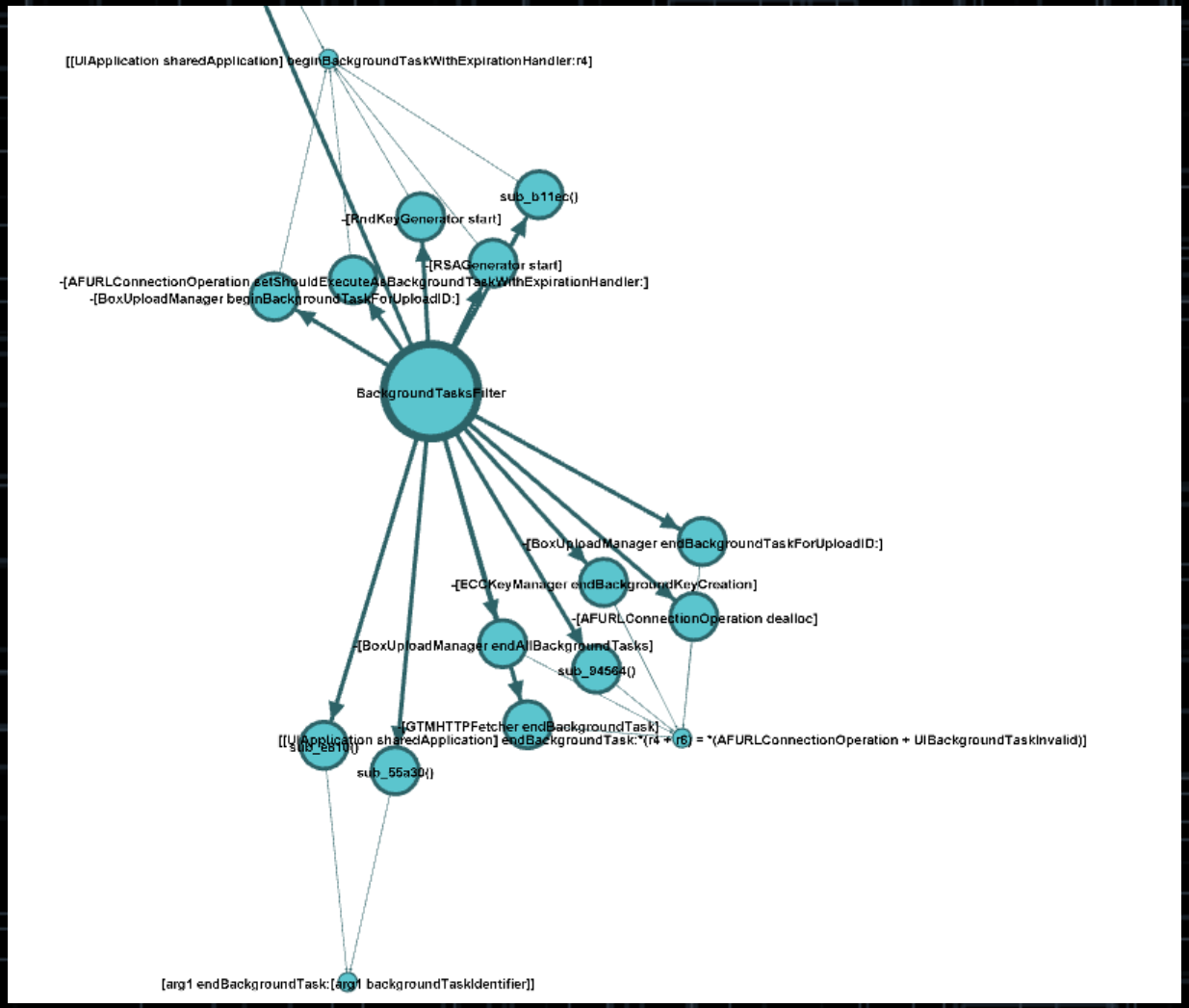
- KeyChainFilter
- GeoLocationFilter
- SeatBeltFilter

- iOS only:
  - IPCFilter
  - DataProtectionFilter
  - BackgroundTasksFilter
  - UIPasteBoardFilter

# Security-Audit Ergebnis



# Security-Audit Ergebnis (2)



# Method calls als Graph