App-Agnostic Post-Execution Semantic Analysis of Android In-Memory Forensics Artifacts

Aisha Ali-Gombe Ph.D.*, *Alexandra Tambaoan**, *Angela Gurfolino**

Golden G. Richard III Ph.D.†

TOWSON UNIVERSITY

LSU

LOUISIANA STATE UNIVERSITY

Mobile Forensics – The need

Memory Forensics

Clipboard data

Volatile registry branches

Network connections

Running processes

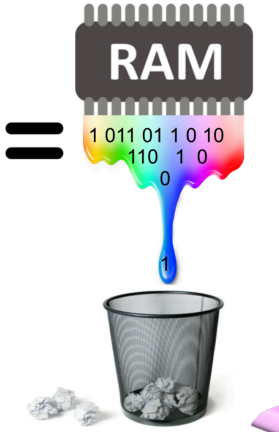
Open files

Encryption keys

Private browsing data

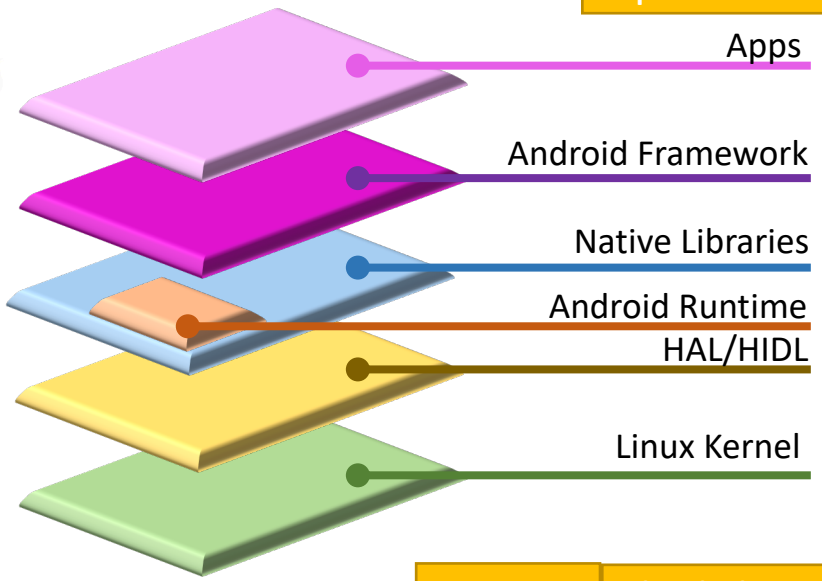
Kernel structures

Application structures



Userland Memory Forensics (UMF)

I stole the squirrels

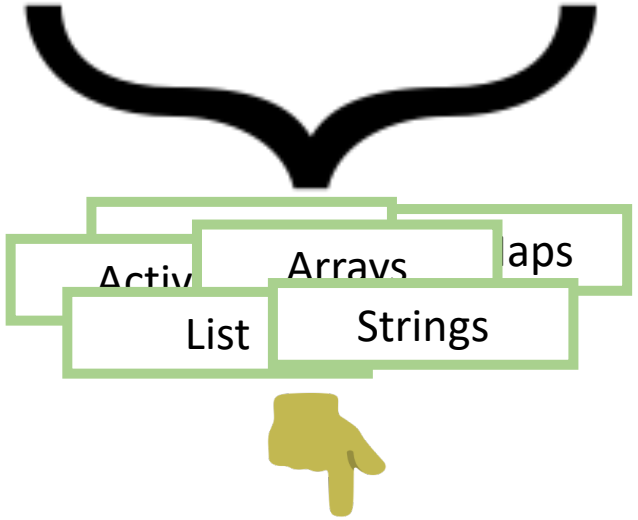


Kernel-land Memory Forensics

IP Header	sbcjkshuqeh qwnbNjhfiid

Prior Work on UMF

App-Specific Techniques
(Anglano et. al, 2017)…

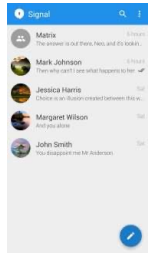


POST: ID: 11/11/2020 - I love cats

Domain-Specific Techniques
(Saltaformaggio et. al, 2015)…

Learning
Brute-force
Instrumentation
Reverse Engineering
….

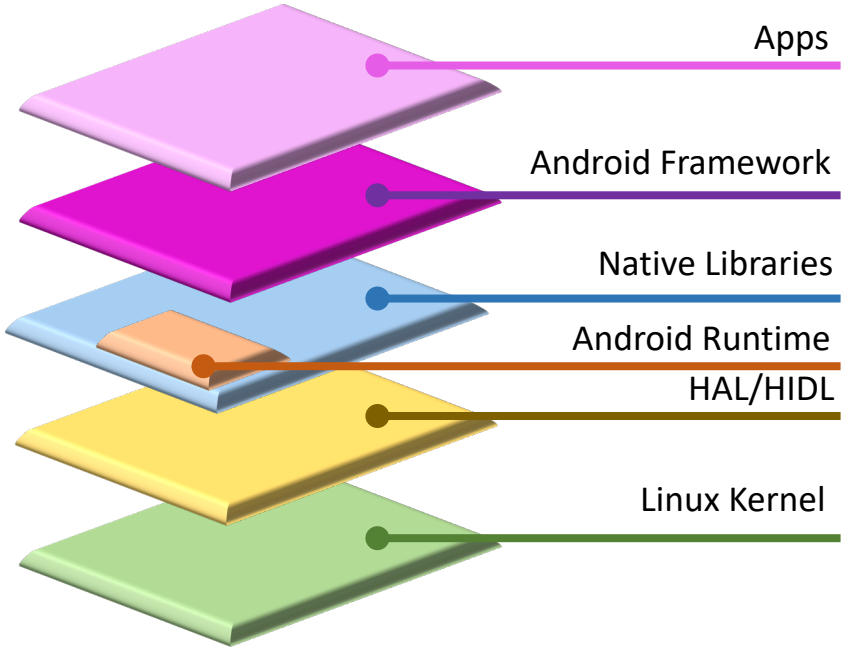
Images
Activities
Frame
Network Objects
…..



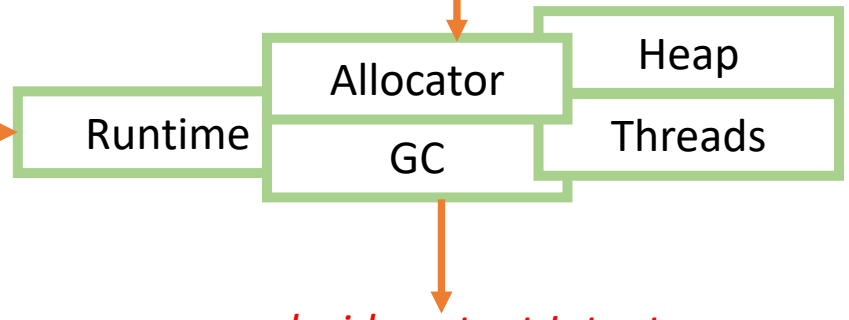
Prior Work on UMF (2)

Generalized Techniques

(Pridgen et. al, 2017, Ali-Gombe et. al 2019)…



Reverse Engineer Runtime



android.content.Intent
java.lang.String
java.lang.String
android.content.Compo
nentName
java.lang.String
android.app.ActivityThr
ead$BindServiceData
…..

Research Objective

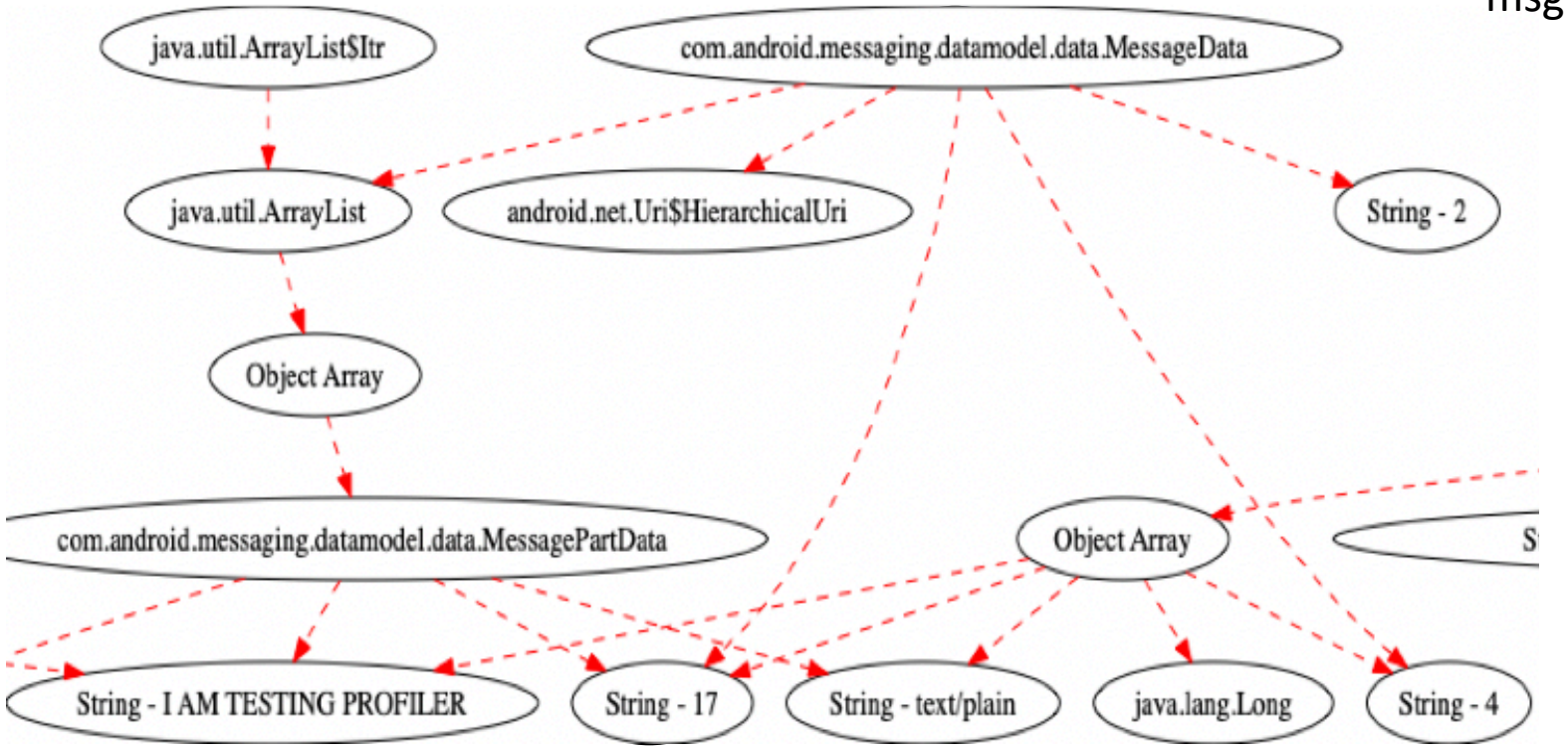
- Generalized and app-agnostic UMF technique called ***OAGen***
  - Automated object relationship (OAG) generation
  - Robust semantic analysis modules (Paths, Context and Scope)

OAGen – Design

G = (N, E)
where n ∈ N and n1 → n2 ∈ E

MessageData msgData = new MessageData()

msgData.setURI (new URI())



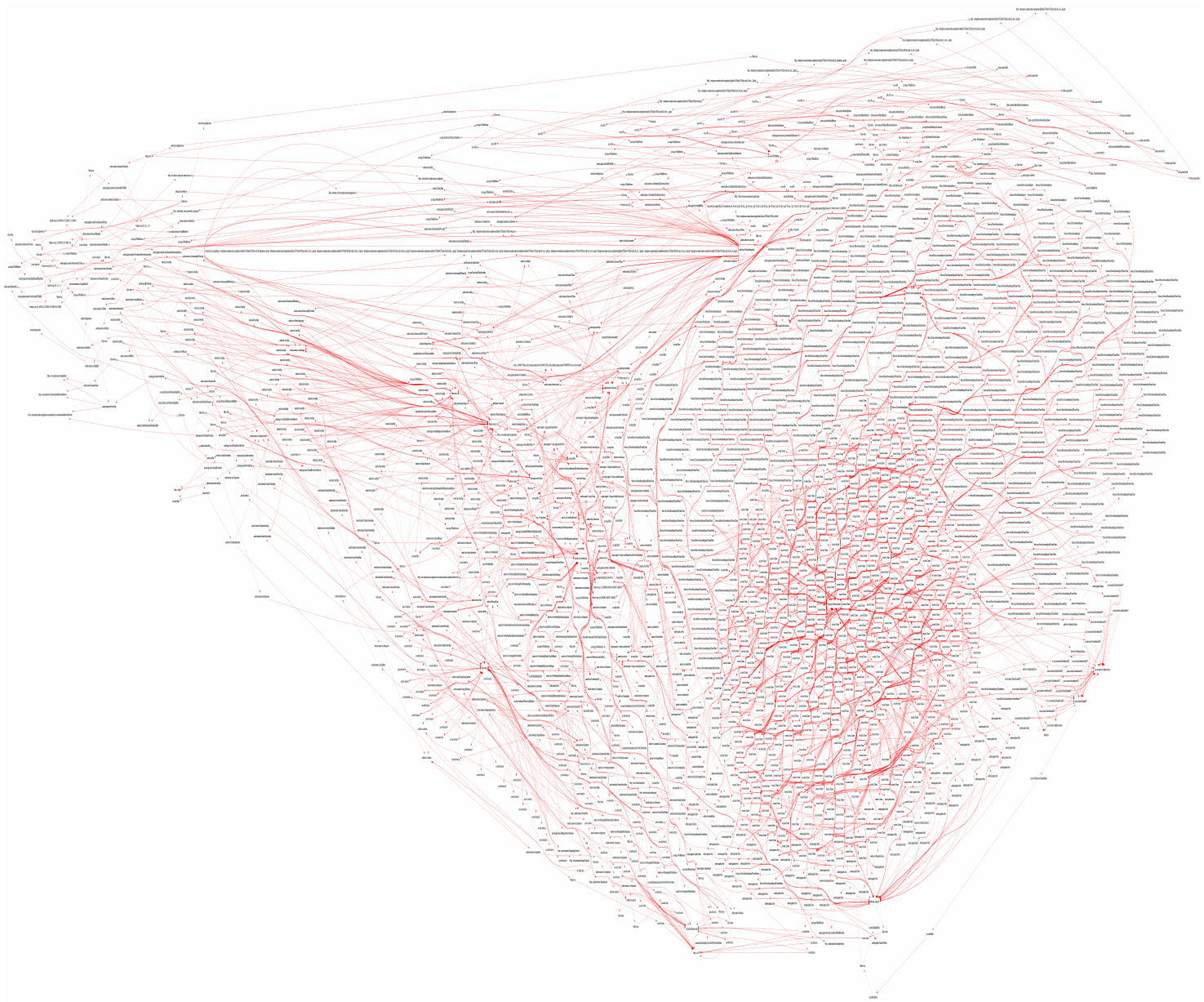
OAG Generation

Algorithm 1: Illustrating the Initial Phase of Object Allocation Graph (OAG) Generation

```
parameters: String:path
nodeList = HeadDump(path)
G=Graph(strict=False,directed=True)
for node in nodeList do
    G.addNode(node.id, node.label, node.data)
end
for node in G.iternodes() do
    recurseObject(G, node)
end
```

Algorithm 2: Recursively Searching and Adding Nodes and Edges to the Graph

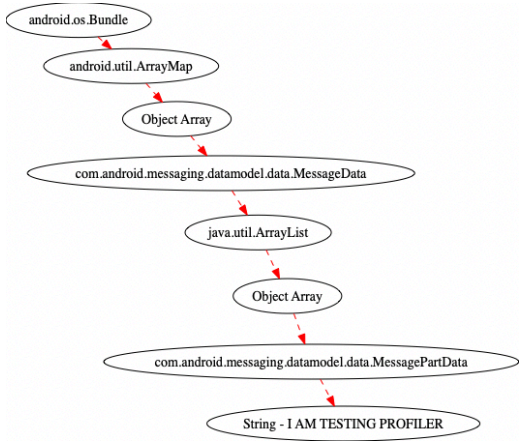
```
parameters: Node:node, Address:heapBegin_
decoded = decodeObject(node, heapBegin_)
if decoded is String then
    node.label= "String Class"
    node.data= decoded
else if decoded is Array then
    if decoded is PrimitiveArray then
        node.label= "Primitive Array Class"
        node.data= decoded
    else if decoded is ObjectArray then
        node.label= "Object Array Class"
        arr = getAllRefs(decoded)
        for x in arr do
            G.addEdge(node, G.addNode(x, node.id=x))
        end
        node.data= decoded
else if decoded is Object then
    node.label= "Object Class"
    ObjectFields=getAllRefs(decoded)
    for x in ObjectFields do
        if x is ref then
            G.addEdge(node, G.addNode(x, node.id=x))
    end
    node.data= decoded
```



Semantic Analysis

Path Exploration

Given directed graph G = (N, E), and two nodes s and t, k-Edge-disjoint Paths are paths with no common edge.



Context Determination

Algorithm 3: Algorithm for Context Determination

```
parameters: Node:target, Graph:G
nodePred=[target]
for n in nodePred do
    for i in G.iterpred(n) do
        nodePred.append(n)
    end
end
edgeList=G.edges(nodePred)
H = G.subgraph(edgeList)
return H
```

Object Scope

Algorithm 4: Algorithm for Object Scope Search

```
parameters: Node:target, Graph:G
nodeList = G.getJThreads() || G.getGCRoot() || G.getComponents()
for node in nodeList do
    nodeSuccessors=[node]
    for n in nodeSuccessors do
        for i in G.itersucc(n) do
            nodeSuccessors.append(n)
        end
    end
    H = G.subgraph(nodeSuccessors)
    if H.has_node(target) then
        Flow(target, node, H)
end
```

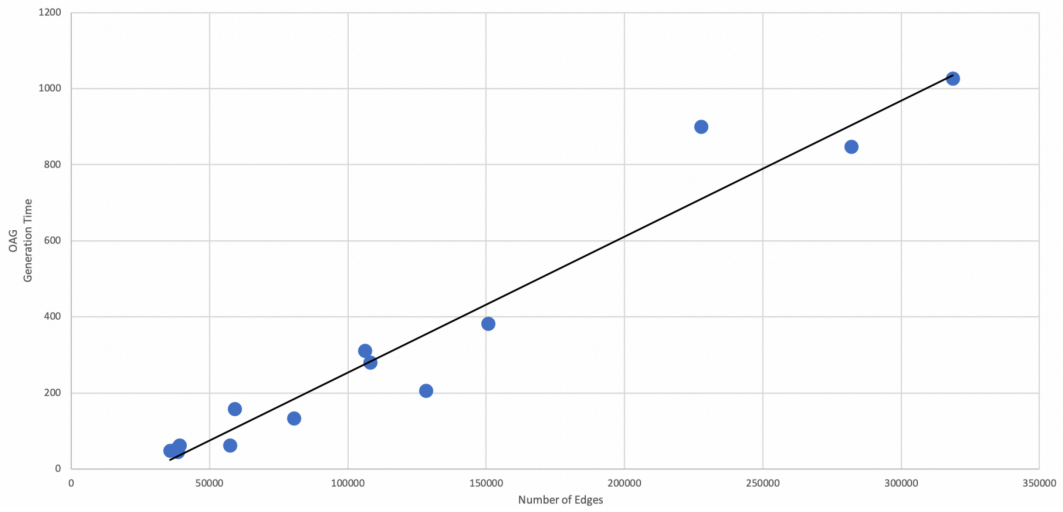
Search Utility – Strings and Objects

OAGen – Implementation & Evaluation

- Python - ~ 3000 loc
- Five modules (OAG Gen, Search utility, path exploration, context and scope determination)
- Prototype - git@github.com:apphackuno/OAGen.git
- Object recovery is based on the default Region-Space memory allocation algorithm
  - Utilizes the base Heapdump of DroidScraper (***Ali-Gombe et al. 2019***)
- Evaluation - MacBook Pro with a 2.6GHz Intel processor and 16GB RAM
  - Performance of the graph generation algorithm
  - Robustness of the recursive search
  - Application of the semantic analysis algorithms

Evaluation - Processing Time

Applications	Image Size (in Mb)	Number of Nodes	Number of Edges	Processing Time (in Minutes)	Throughput (in Mbps)
com.whatsapp	1338	318632	476586	1025.5	0.03
com.facebook.katana	1733	282120	411522	847	0.04
com.callpod.android_apps.keeper	1237	227776	330240	899.5	0.03
org.thoughtcrime.securesms	1249	150824	229308	381.5	0.06
com.peaksel.gothicwallpapers	1272	128256	183276	206.5	0.11
com.klinker.android.evolve_sms	1208	108248	154014	280	0.08
com.android.messaging	1257	106304	156318	311.5	0.07
com.dd.monkey	1169	80512	116472	133	0.15
com.android.tencent.zdevs.bah	1192	59232	81462	157.5	0.13
com.easyhin.usereasyhin	1192	57504	85212	63	0.32
com.husor.beibei	1211	39320	50964	63	0.33
org.chromium.webview_shell	1289	38560	50070	45.5	0.48
com.yandex226.yandex	1201	38528	49590	56	0.36
com.yxxinglin.xzid	1195	35832	45294	49	0.41



Average throughput = 0.19Mbps

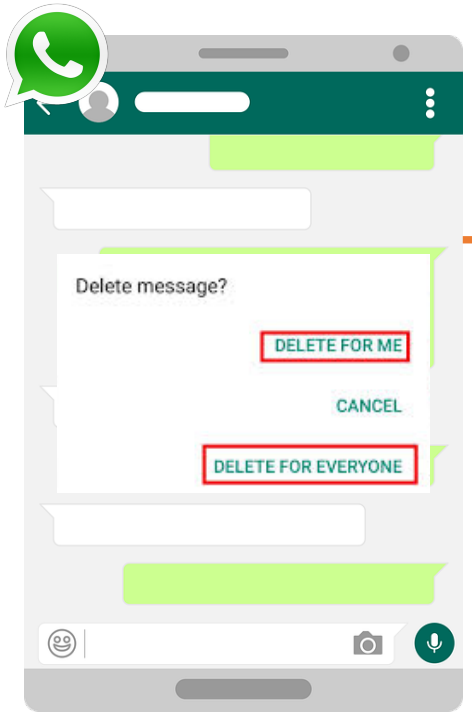
Evaluation – Macro Benchmark

Objective: Robustness of OAGen in searching and finding object references

	com.whatsapp	com.android.messaging	com.yandex226.yandex
Heap's objects_allocated_ field	301,742	92,046	11,172
Droidscraper	301,467	90,053	10,328
OAGen	318,632	106,304	35,528

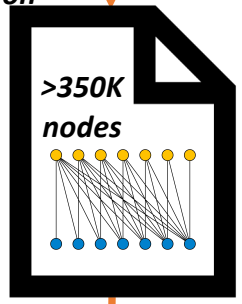
*OAGen recovers objects in heap, stack, initialized data and code segments

Evaluation – Semantic Analysis 1: Cybercrime Investigation



Memfetch

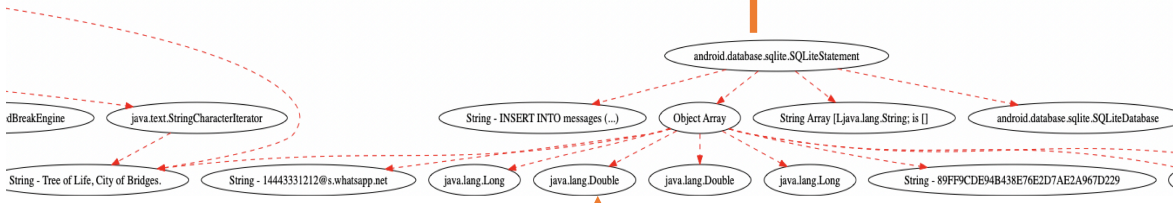
OAG Generation



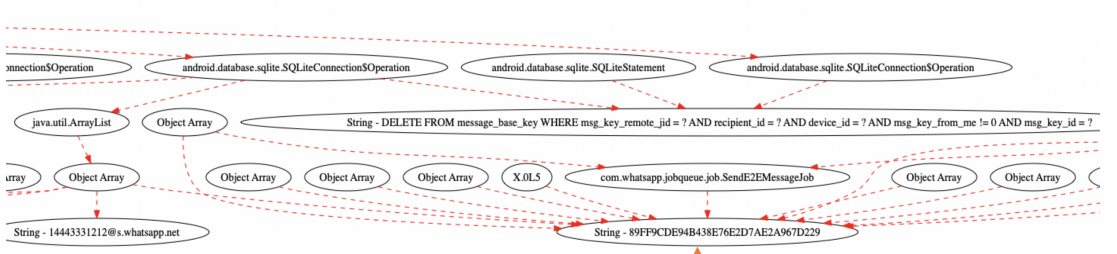
Strings Search Utility

```
0x130d37e8 String - 9189@s.whatsapp.net
0x130d3960 String - Key[id=89FF9CDE94B438E76E2D7AE2A967D229, from_me=true, remote_jid=9189@s.whatsapp.net]
0x130d3c78 String - msgstore/add/send; key=Key[id=89FF9CDE94B438E76E2D7AE2A967D229, from_me=true, remote_jid=9189@s.
whatsapp.net]; media_wa_type=0; status=0
0x130d3ea0 String - LL_I W
0x130d4060 String - LL_I W [268:Messages Async Commit Thread] msgstore/add/send; key=Key[id=89FF9CDE94B438E76E2D7AE2
A967D229, from_me=true, remote_jid=9189@s.whatsapp.net]; media_wa_type=0; status=0
0x130d4200 String - BEG
0x130d4490 String - FMessage/setStorageType/should only update storage type when it is undefined; current=0; new=1
0x130d4740 String - CoreMessageSTore/insertMessage/Chat row not inserted; chatRowId=14
0x130d4828 String - 14443331212@s.whatsapp.net
0x130d4898 String - INS
0x136b4078 String - Tree of Life, City of Bridges.
0x130d4a38 String - this
0x130d4b30 String - this
0x130d4b48 String - is
0x130d4c30 String - is
0x130d4c48 String - just
0x130d4d40 String - just
0x130d4d58 String - a
0x130d4e40 String - a
0x130d4e58 String - test
0x130d4f50 String - test
```

Context Determination with a depth of 10



Context Determination with a depth of 50

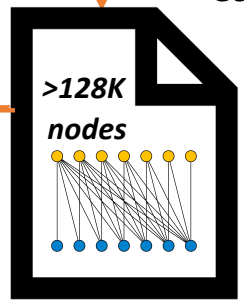


- End-to-end encryption
- Complete destruction of evidence

Evaluation – Semantic Analysis 2: Data Exfiltration

Memfetch

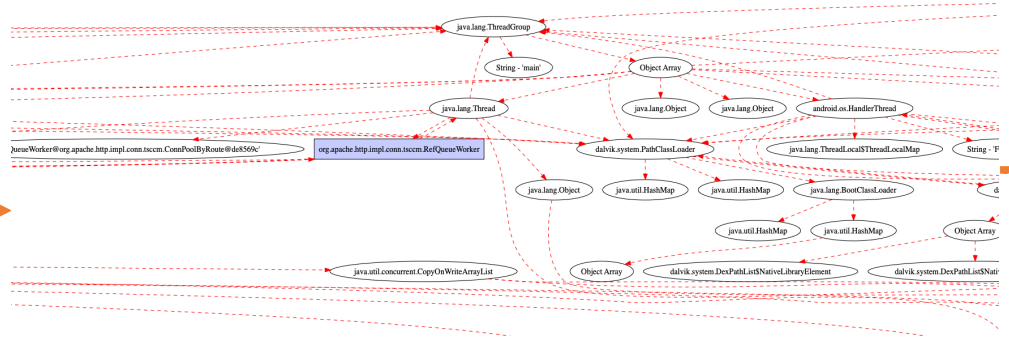
OAG Generation



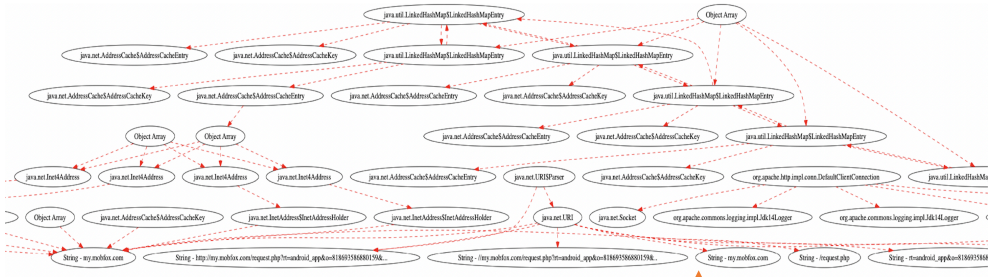
Object Search Utility

```
Android Object Allocation Graph
org.apache.http.impl.client.DefaultHttpClient 0x12c57030
org.apache.http.impl.client.DefaultHttpClient 0x12c5f490
org.apache.http.impl.client.DefaultHttpClient 0x12d42ea8
org.apache.http.impl.client.DefaultHttpClient 0x132c0ba0
```

Object Scope within RefQueueWorker thread



Context Determination



Context Determination

```
java.lang.ref.FinalizerReference 0x12d40118
android.os.BinderProxy 0x12d400f8
java.lang.ref.WeakReference 0x12d40140
java.lang.ref.FinalizerReference 0x12d489f8
com.android.internal.telephony.ITelephony$Stub$Proxy 0x12d40158
String – '818693586880159 ' 0x12d40168
java.lang.StringBuilder 0x12d40188
```

Challenges and Future Work

- Require considerable amount of time to weed out noise

- Tentative execution path reconstruction
  - mapping OAG to in-memory code
- Need more testing to explore the effects:
  - garbage collection/process state on recovered data
  - program obfuscation

Key Takeaway

- OAGen - a tool for post-execution semantic analysis of Android applications
  - Object Allocation Graph Generation
  - Algorithms
    - Path exploration
    - Context
    - Object Scope Determination

Acknowledgement

This work is supported by the U.S. National Science Foundation under the awards -

NSF-CRII-SaTC-1850054

THANK YOU!
QUESTIONS!

git@github.com:apphackuno/OAGen.git
aaligombe@towson.edu