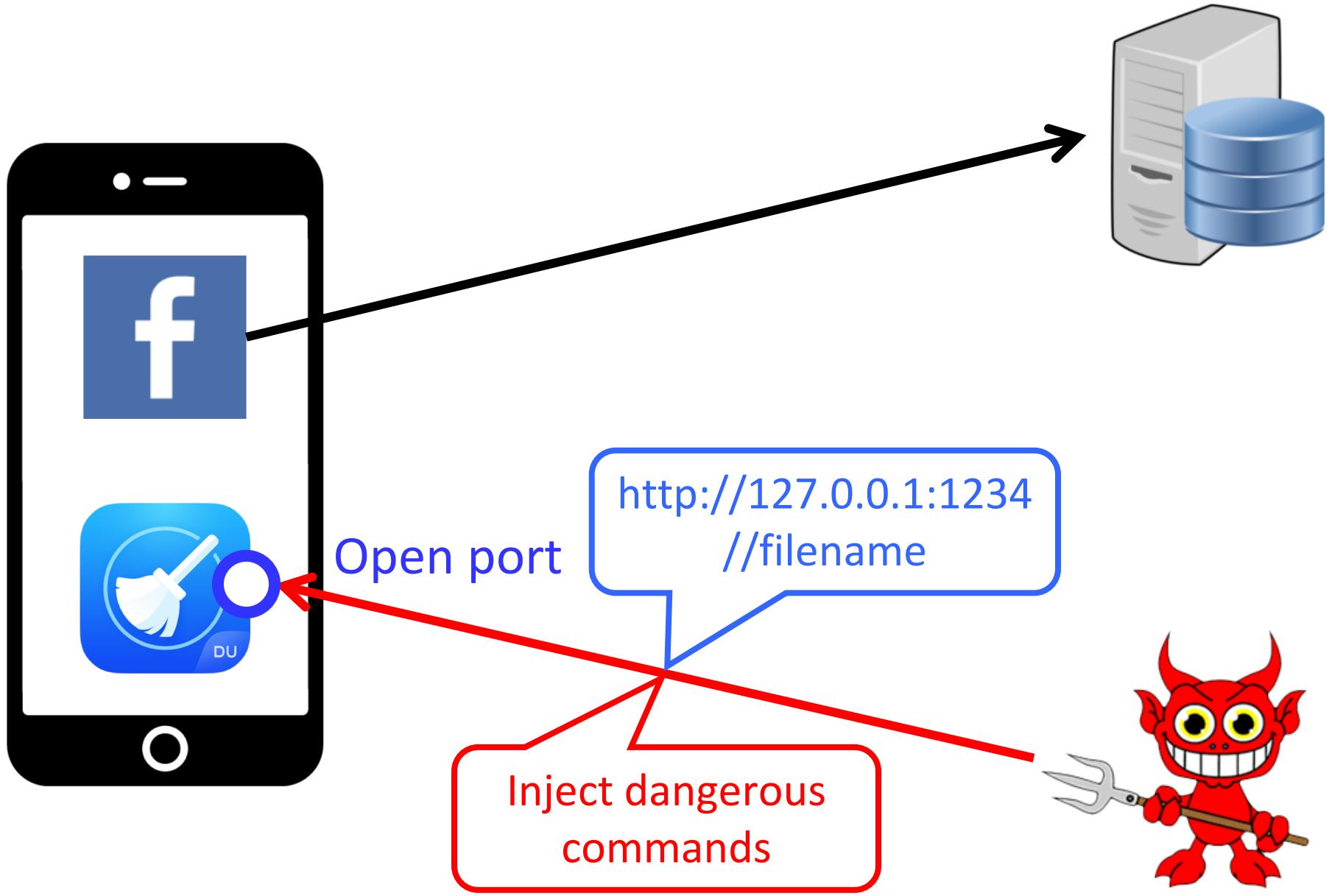
Understanding Open Ports in Android Applications: Discovery, Diagnosis, and Security Assessment

Daoyuan Wu[1], Debin Gao[1], Rocky K. C. Chang[2],
En He[3], Eric K. T. Cheng[2], and Robert H. Deng[1]

[1] SMU SINGAPORE MANAGEMENT UNIVERSITY

[2] THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

[3] China Electronic Technology Cyber Security Co., Ltd.


Open port
http://127.0.0.1:1234
//filename
Inject dangerous
commands

The First Step: Discovering Open Ports in Apps

Static Analysis

In-lab Dynamic Analysis

OPAnalyzer [EuroS&P'17]

Cannot mimic real user inputs to driven apps

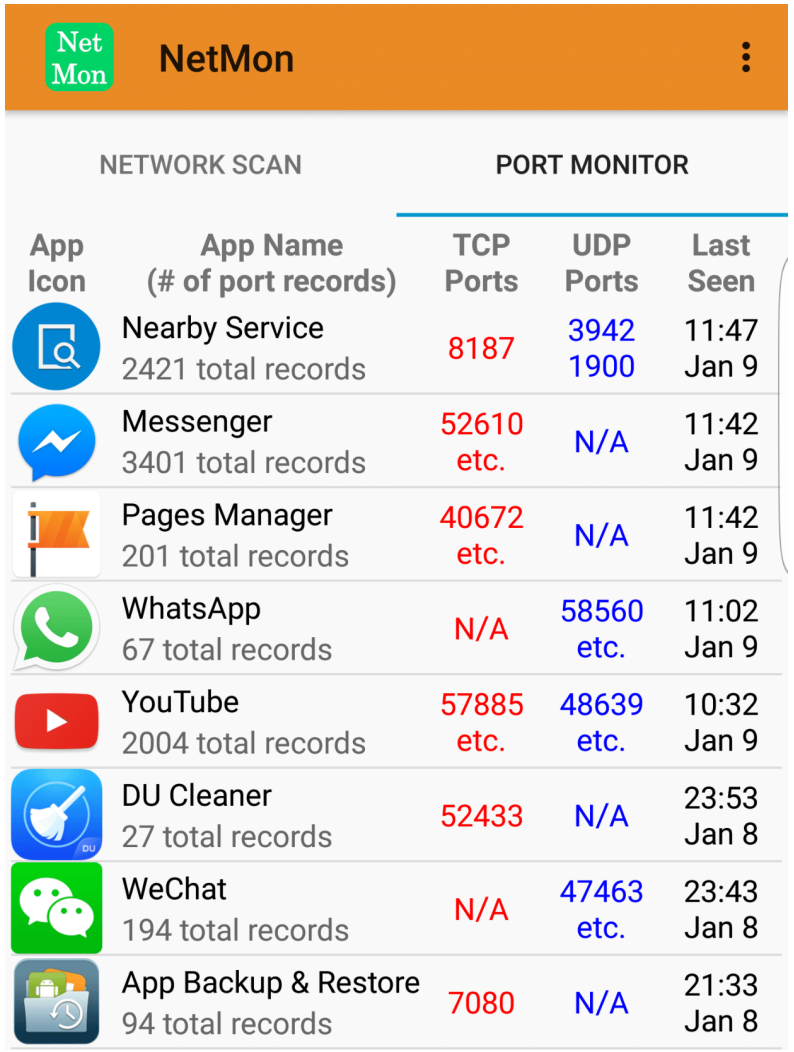
Issues:
dynamic code loading, complex implicit flows, and code obfuscation.

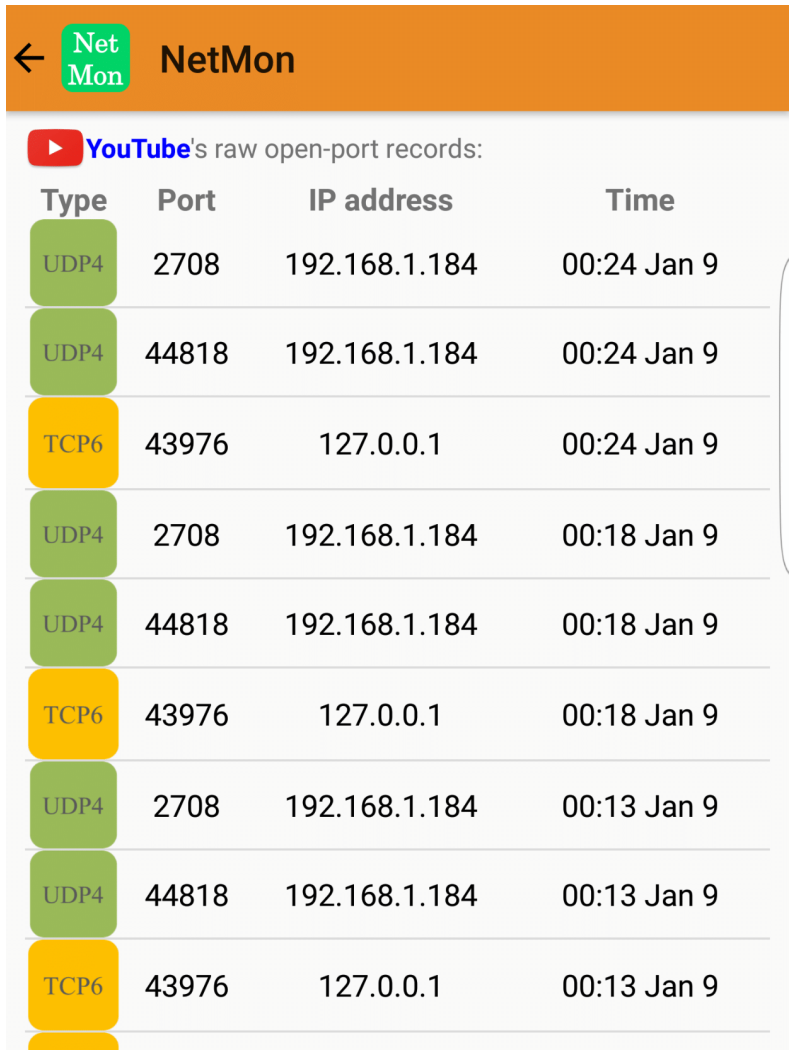
Difficult to recognize random port numbers

Crowdsourcing Discovery

Leverage users' interaction with their smartphones to monitor open ports

NetMon: On-device Open Port Monitoring

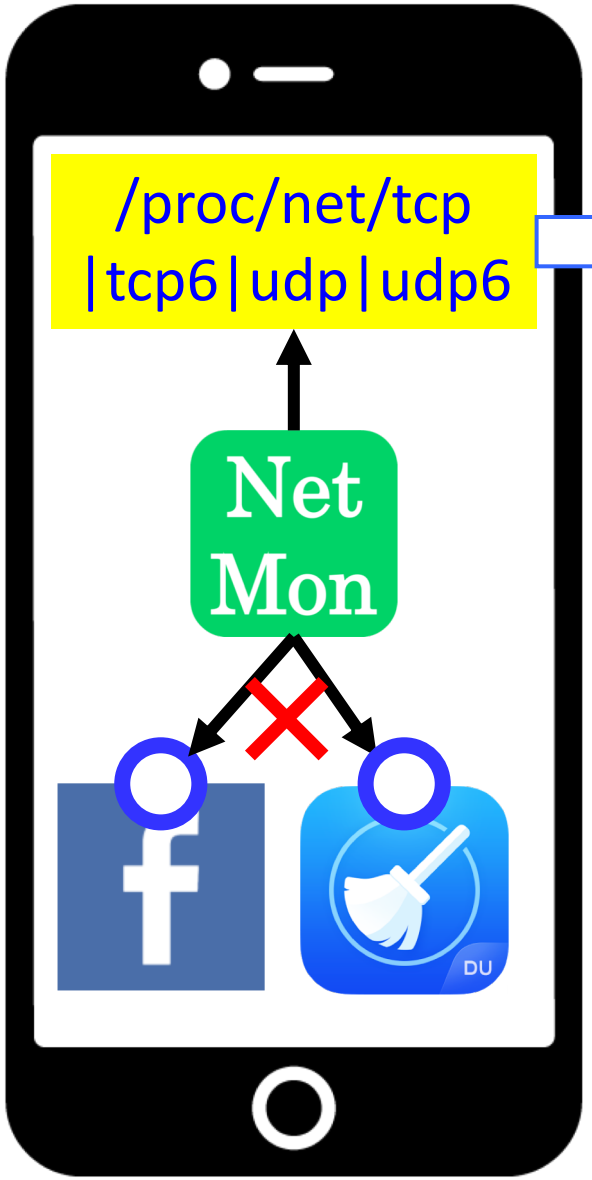




Available on Google Play since October 2016

https://play.google.com/store/apps/details?id=com.netmon

Port Monitoring Mechanism



$ cat /proc/net/tcp **(accessible also on the latest Android 8 and 9)**

```
  sl  local_address                         remote_address                        st tx_queue
rx_queue tr tm->when retrnsmt   uid
0: 00000000000000000FFFF00000100007F:9AE0
00000000000000000000000000000000:0000 0A 00000000:00000000
00:00000000 00000000 10156
1: 00000000000000000FFFF00000100007F:EC22
00000000000000000000000000000000:0000 0A 00000000:00000000
00:00000000 00000000 10272
2: 00000000000000000FFFF00002600040A:E8EA
00000000000000000FFFF00006B72662F:01BB 06 00000000:00000000
03:00001279 00000000     0
3: 00000000000000000FFFF00002600040A:84B0
00000000000000000FFFF00005FC2D9AC:01BB 08 00000000:00000001
00:00000000 00000000 10015
```

Periodically analyze proc with minimal overhead

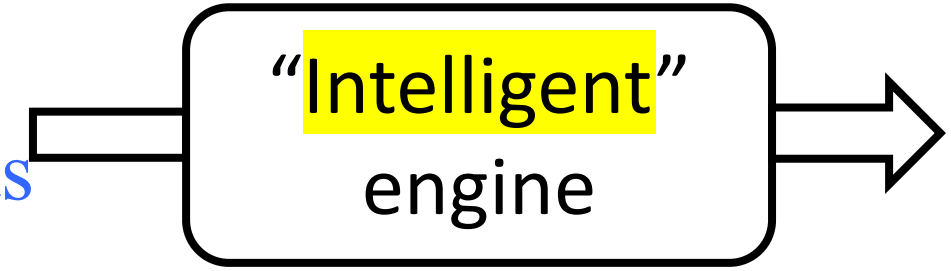
Server-side Open-Port Analytic Engine

UID	App	Type	IP	Port	Time
U1	Netflix	UDP4	0.0.0.0	1900	T1
U1	Netflix	UDP4	0.0.0.0	**39798**	T1
U2	Netflix	UDP4	0.0.0.0	1900	T2
U2	Netflix	UDP4	0.0.0.0	**32799**	T2
.....					
Ux	Netflix	TCP4	0.0.0.0	9080	Tx
Uy	Netflix	TCP4	0.0.0.0	9080	Ty

App	Type	IP	Port
Netflix	TCP4	0.0.0.0	9080
Netflix	UDP4	0.0.0.0	1900

App	Type	IP	Port
Netflix	UDP4	0.0.0.0	Random

Raw port monitoring records

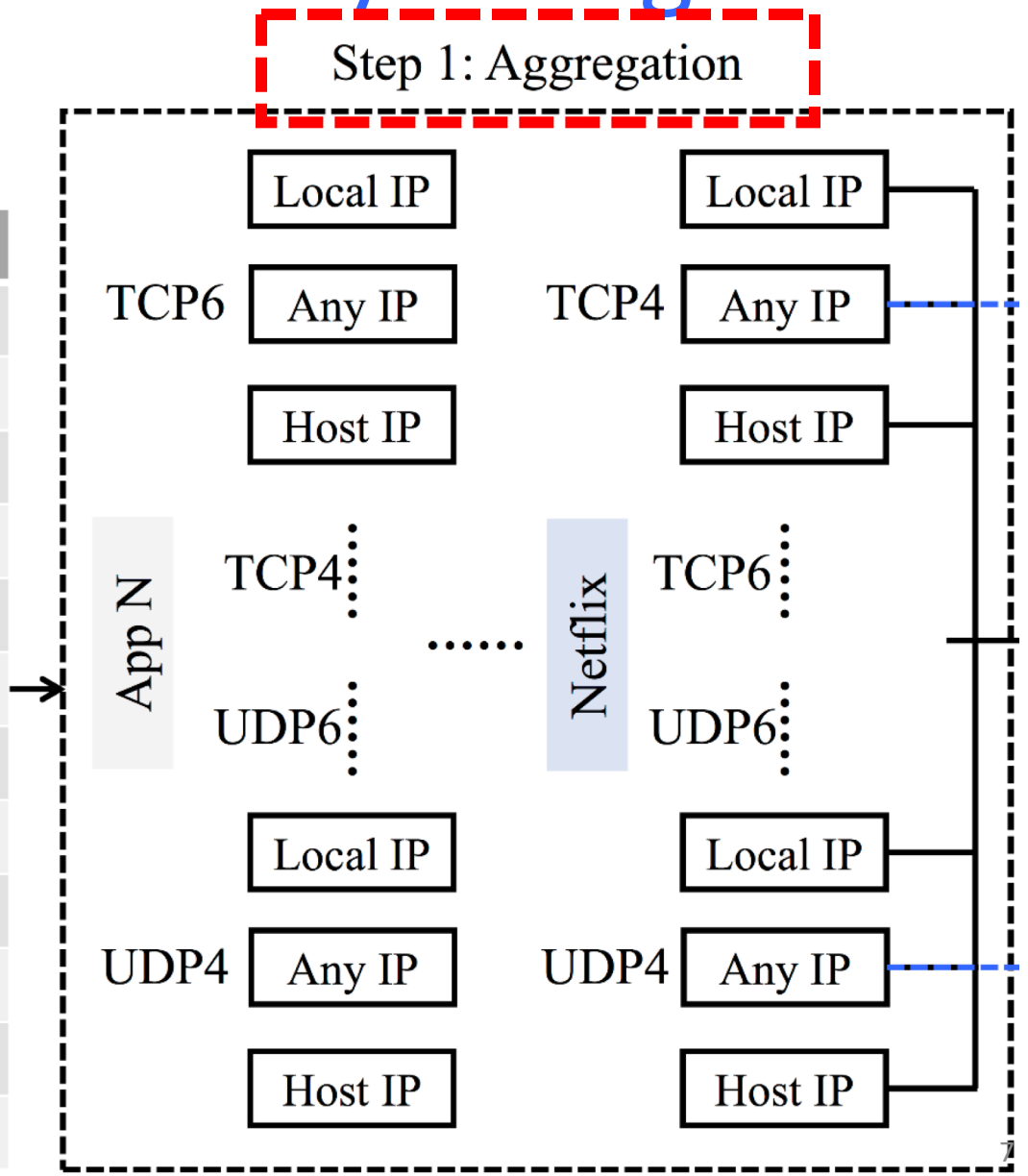


Per-app open ports

Server-side Open-Port Analytic Engine

Raw port monitoring records from crowdsourcing
(Using Netflix's TCP4/UDP4 ports as examples)

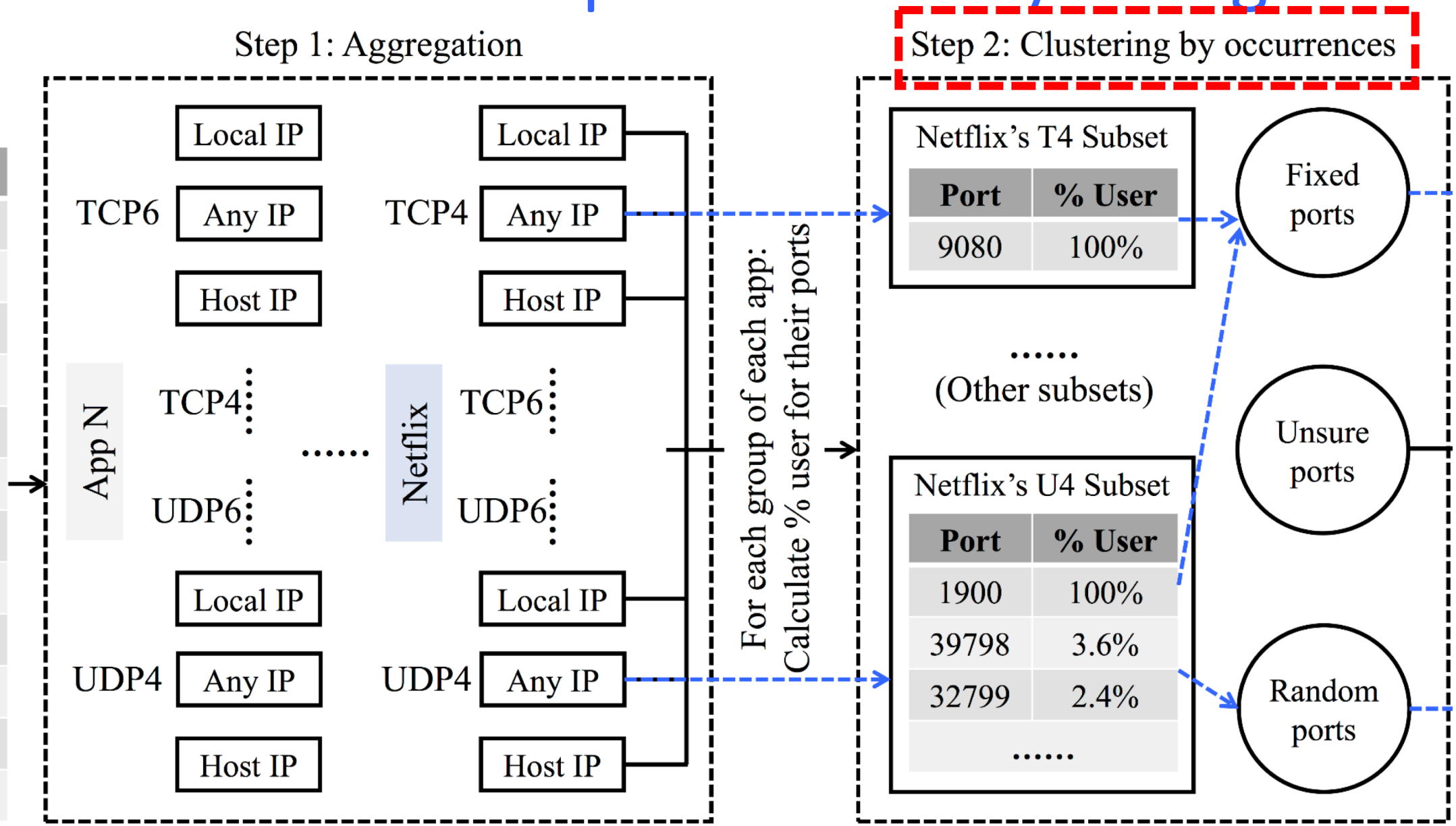
UID	App	Type	IP	Port	Time
U1	Netflix	UDP4	0.0.0.0	1900	T1
U1	Netflix	UDP4	0.0.0.0	39798	T1
U2	Netflix	UDP4	0.0.0.0	1900	T2
U2	Netflix	UDP4	0.0.0.0	32799	T2
.....					
Ux	Netflix	TCP4	0.0.0.0	9080	Tx
Uy	Netflix	TCP4	0.0.0.0	9080	Ty
.....					
Un	App N	UDP6	127.0.0.1	51663	Tn
Uo	Facebook	TCP6	127.0.0.1	46467	To
Up	WeChat	TCP4	192.x.x.x	9014	Tp
Uq	WeChat	TCP4	10.20.x.x	9014	Tq



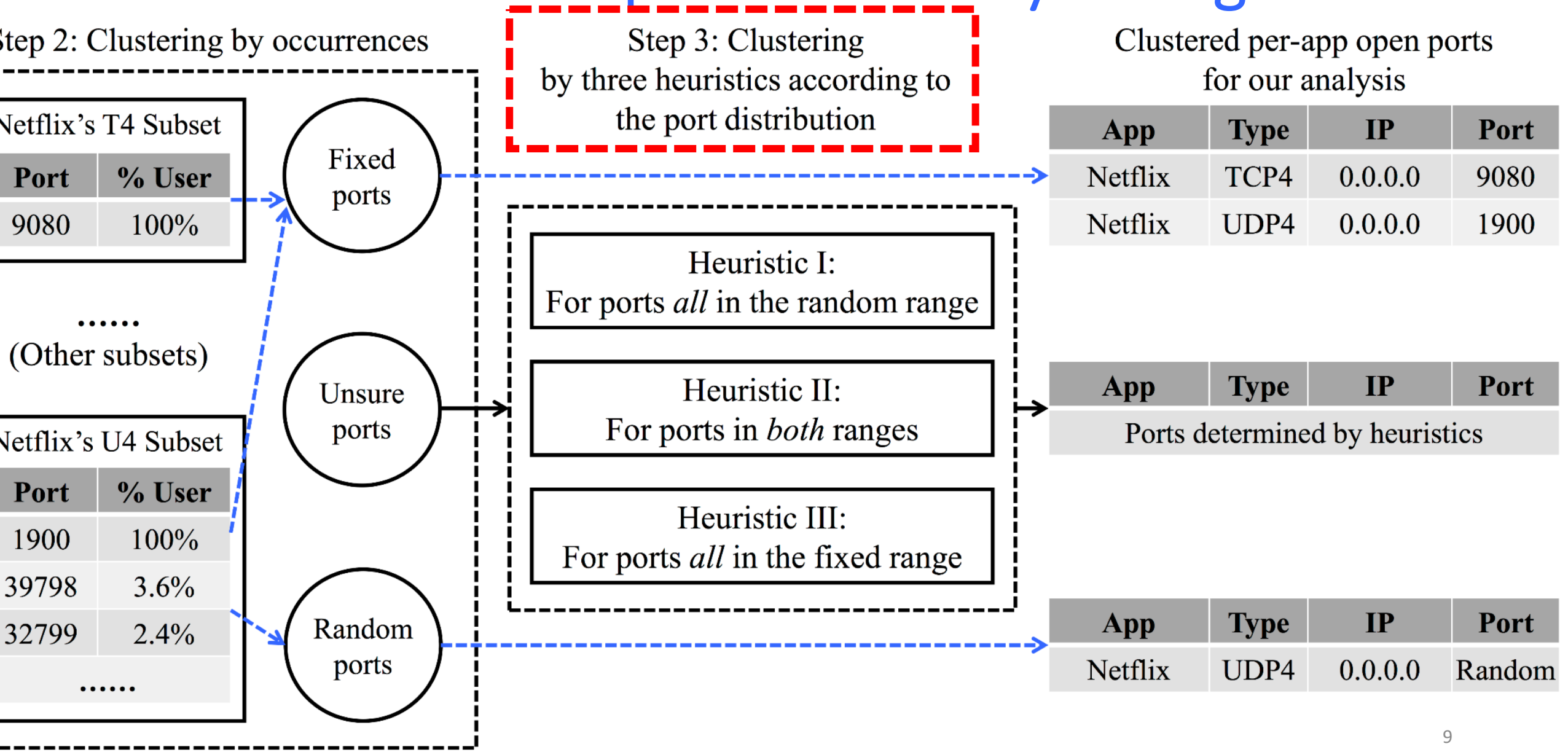
Server-side Open-Port Analytic Engine

urcing
ples)

Time
T1
T1
T2
T2
Tx
Ty
Tn
To
Tp
Tq



Server-side Open-Port Analytic Engine



Crowdsourced Open Port Results

- The ten-month data:
  - 3,293 user phones from 136 different countries
  - 26% are from US, while diverse for others
- 40M port monitoring records:
  - 2,778 open-port apps
  - And their 4,954 open ports

- The effectiveness:
  - Discovered 2,284 apps with TCP open ports, **vs.** 1,632 apps detected in state-of-the-art research [EuroS&P'17].
  - In a controlled set of apps with TCP open ports, 25.1% of them use dynamic or obfuscated codes for open ports.

- The pervasiveness:
  - Correlated with top 3,216 apps from Google Play, 492 of them are with open ports.
  - **Pervasiveness: 15.3%.**

Open Ports in 925 Popular Apps

Category	App Name	Type	IP†	Port
Social	Facebook	TCP	L	Random
	Instagram	TCP	L	Random
	Google+	TCP	H	Random
		TCP	L	Random
	VK	TCP	H	48329
		TCP	L	Random
Communication	Messenger	TCP	L	Random
	WeChat	TCP	H	9014
	Skype	TCP	H	Random
		TCP	L	Random
	Chrome	TCP	L	5555
	Firefox	TCP	H	8080
		TCP	L	Random
		UDP	H	1900
Video Players or Music & Audio	YouTube	TCP	H	Random
		TCP	L	Random
	GPlay Music	TCP	L	Random
		UDP	H	1900
	Spotify	TCP	H	Random
	Amazon Music	TCP	L	Random
		TCP	H	Random
Tools	Google Play Services	UDP	H	2346
		UDP	H	5353
	Google	TCP	H	20817
	Clean Master	TCP	L	Random
	360 Security	TCP	L	Random
	Avast	TCP	H	20817
		TCP	L	Random
Productivity	Google Drive	TCP	L	Random
	Cloud Print	UDP	H	5353
	ES File Explorer	TCP	H	42135
		TCP	H	59777
		TCP	L	Random
		UDP	H	5353
Entertainment	GPlay Games	TCP	L	Random
	Netflix	TCP	H	9080
		UDP	H	1900
		UDP	L	Random
	Peer Smart Remote	TCP	L	Random
		UDP	H	5353
Games	Plants vs. Zombies 2	UDP	H	24024
	Asphalt 8	TCP	H	7940
	Solitaire	TCP	L	Random
	Sonic Dash	TCP	L	Random

Open Ports in 755 Built-in Apps

Table 2: Top ten device vendors that include open-port apps.

Vendor	#	Top Five Open Port Numbers					
Samsung	186	UDP:	5060	68	1900	6100	6000
		TCP:	5060	6100	6000	7080	8230
LG	75	UDP:	68	1900	19529	5060	39003
		TCP:	5060	59150	59152	8382	39003
Sony	69	UDP:	68	1024	1900	1901	-
		TCP:	5000	5900	5001	9000	30020
Qualcomm	42	UDP:	68	5060	1900	32012	-
		TCP:	5060	6100	4000	4500	4600
MediaTek	26	UDP:	68	5060	50001	50002	50003
		TCP:	5060	50001	-	-	-
Lenovo	25	UDP:	68	5060	50000	50001	52999
		TCP:	2999	5060	50001	55283	39003
Motorola	21	UDP:	68	32012	16800	-	-
		TCP:	2631	20817	-	-	-
Huawei	13	UDP:	68	1900	8108	-	-
		TCP:	-	-	-	-	-
ASUS	13	UDP:	68	5353	11572	11574	-
		TCP:	2222	5577	8258	8282	8990
XiaoMi	11	UDP:	68	1900	5353	-	-
		TCP:	6000	8081	8682	-	-

More than half of these built-in apps contain UDP open port 68.

One quarter (175 apps, 23.2%) have TCP/UDP port 5060 open.

41 Samsung and 16 LG models modify some Android AOSP apps to introduce port 5060.

- TCP port 6000 in Xiaomi Browser
- UDP port 19529 in LG's 18 apps

While crowdsourcing is effective in discovering open ports,
it does not reveal the code-level information
for more in-depth understanding or diagnosis.

Open Port Diagnosis via Static Analysis

SDK?

1

```
// API #1-#3
ServerSocket(int port);
ServerSocket(int port, int backlog);
ServerSocket(int port, int backlog, InetAddress addr);

// API #4-#6
SSLServerSocket(int port);
SSLServerSocket(int port, int backlog);
SSLServerSocket(int port, int backlog, InetAddress addr);

// API #7-#9
//class ServerSocketFactory:
createServerSocket(int port);
createServerSocket(int port, int backlog);
createServerSocket(int port, int backlog,InetAddress addr);

// API #10-#11
//ServerSocket socket = new ServerSocket();
socket.bind(SocketAddress addr);
socket.bind(SocketAddress addr, int backlog);
```

2 Insecure parameters?

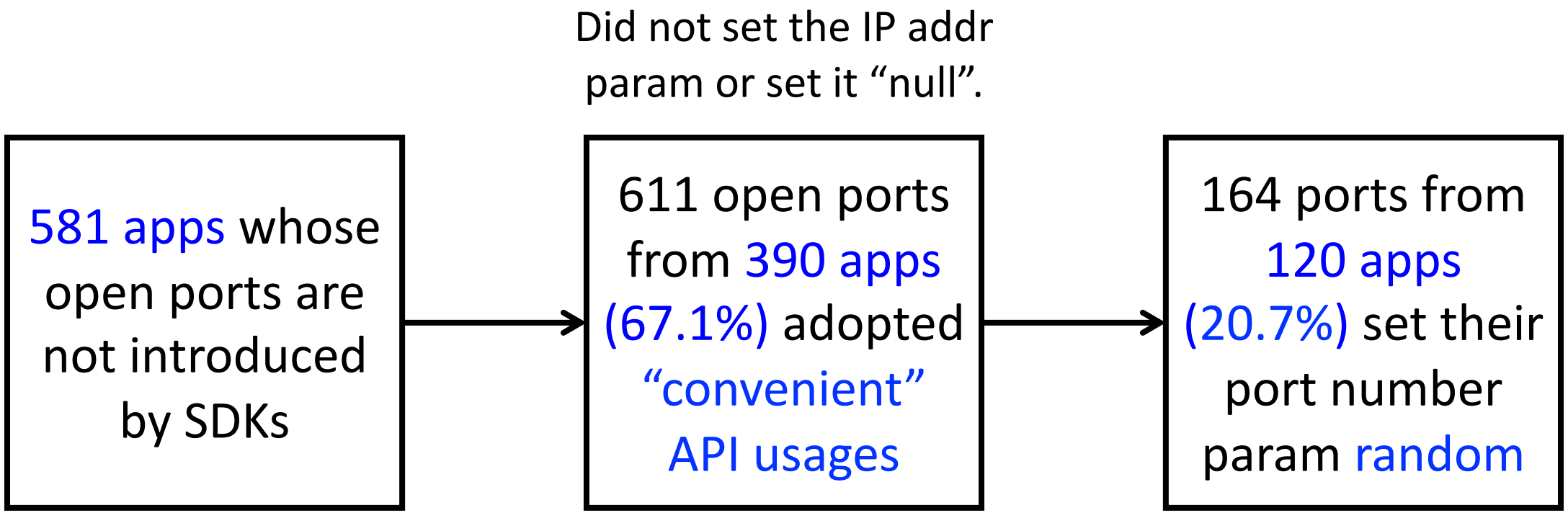
Listing 1: All `ServerSocket` constructor APIs.

Diagnosis I: Open-Port SDKs

- Out of the 1,520 open-port apps:
  - 61.8% are solely due to SDKs;
    Facebook SDK is the major contributor.
  - 13 open-port SDKs detected:

SDK	Pattern	#
Facebook Audience Network SDK [17]	Class='com.facebook.ads.%', Tag=ProxyCache, Ip=127.0.0.1, Port=0, Backlog=8	897
Yandex Metrica SDK [38]	Class='com.yandex.metrica.%', Port=29009|30102	28
CyberGarage UPnP SDK [14]	Class=org.cybergarage.http.HTTPServer, Ip=getHostAddress(), Port=8058|8059	19
MIT App Inventor SDK [25]	Class=com.google.appinventor.components.runtime.util.NanoHTTPD, Port=8001	19
Tencent XG Push SDK [36]	Class=com.tencent.android.tpush.service.XGWatchdog, Port=RANDOM+55000	13
Corona Game Engine SDK [13]	Class=com.ansca.corona.CoronaVideoView, Port=0, Backlog=8	11
Alibaba AMap SDK [4]	Class='com.amap.%', Port=43689	9
Millennial Ad SDK [24]	Class='com.millennialmedia.android.%', Tag=MillennialMediaAdSDK, Ip=null, Port=0	8
PhoneGap SDK [28]	Class=com.phonegap.CallbackServer, Port=0	6
Titanium SDK [37]	Class=org.appcelerator.kroll.common.TiFastDev, Tag=TiFastDev, Port=7999	6
Aol AdTech SDK [8]	Class=com.adtech.mobilesdk.publisher.cache.NanoHTTPD, Port=RANDOM+9000	6
Apache Cordova SDK [9]	Class=org.apache.cordova.CallbackServer, Port=0	4
Getui Push SDK [18]	Class='com.igexin.push.%', Port=48432|51688, Ip=0.0.0.0	3

Diagnosis II: Insecure API Usages



20.7% (120/581) open-port apps adopt convenient but insecure API usages.

In the last phase of our pipeline,
we perform three novel
security assessments of open ports.

Vulnerability Patterns Identified in Open Ports

ID	Vulnerability Patterns	Representative Apps Affected
P1	No/insufficient checks for information transmission	
P2	No/insufficient checks for command execution	
P3	Crash-of-Service (CoS)	Skype, Instagram
P4	Stealthy Data Inflation	Facebook SDK, Instagram
P5	Insecure Analytics Interface	Weibo, Alibaba & Baidu SDKs

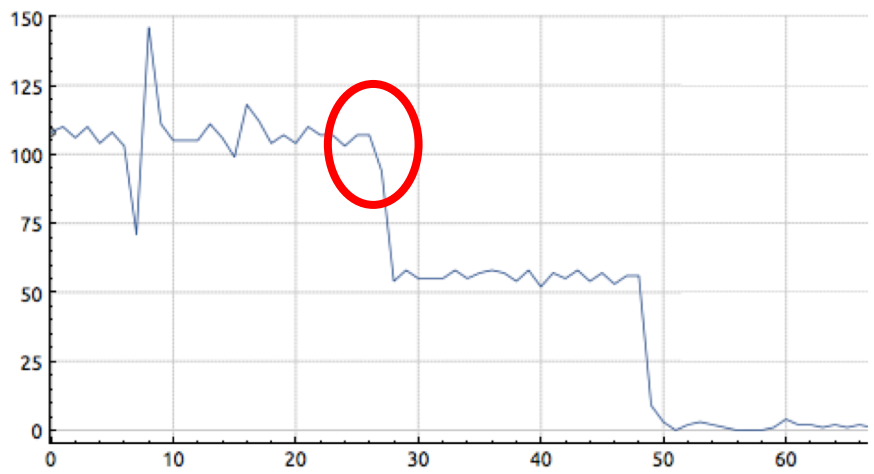
Terminate on-going sessions by sending two UDP packets

Crash Instagram by sending just a HTTP request

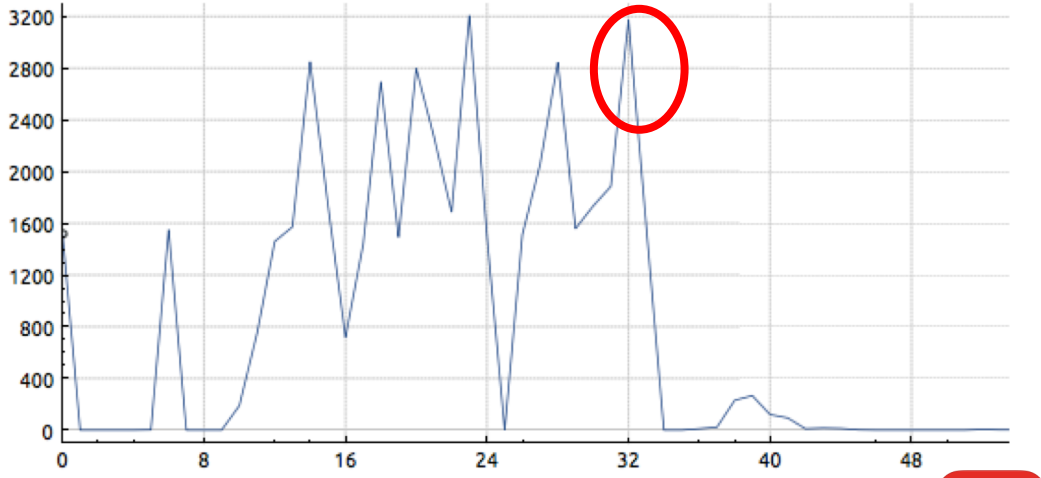
Some open ports are used as **an analytics interface** for their companion websites.

Send a HTTP URL request pointing to a large file, to maliciously **inflate victim apps' cellular data usage** in the background.

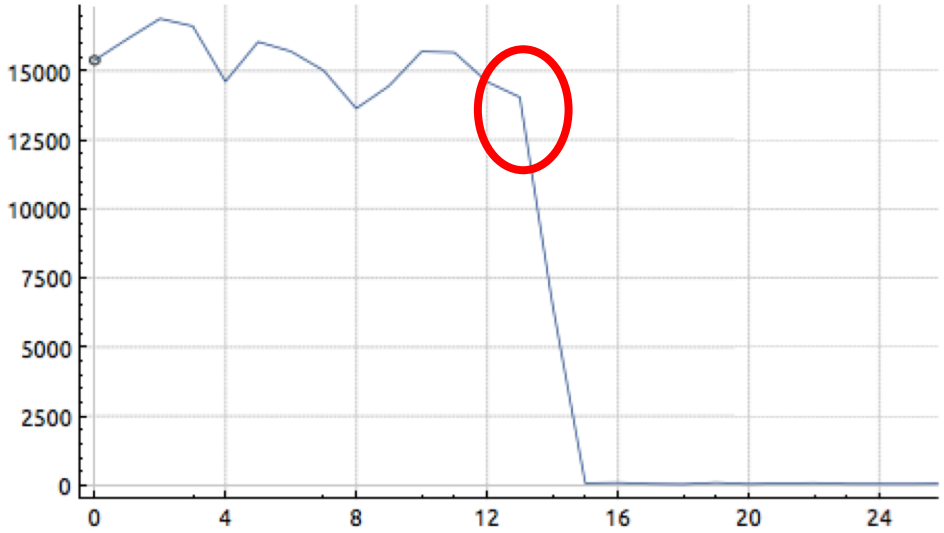
Denial-of-Service Attack Evaluation



(a) WeChat's voice call (DoS at ~26s).

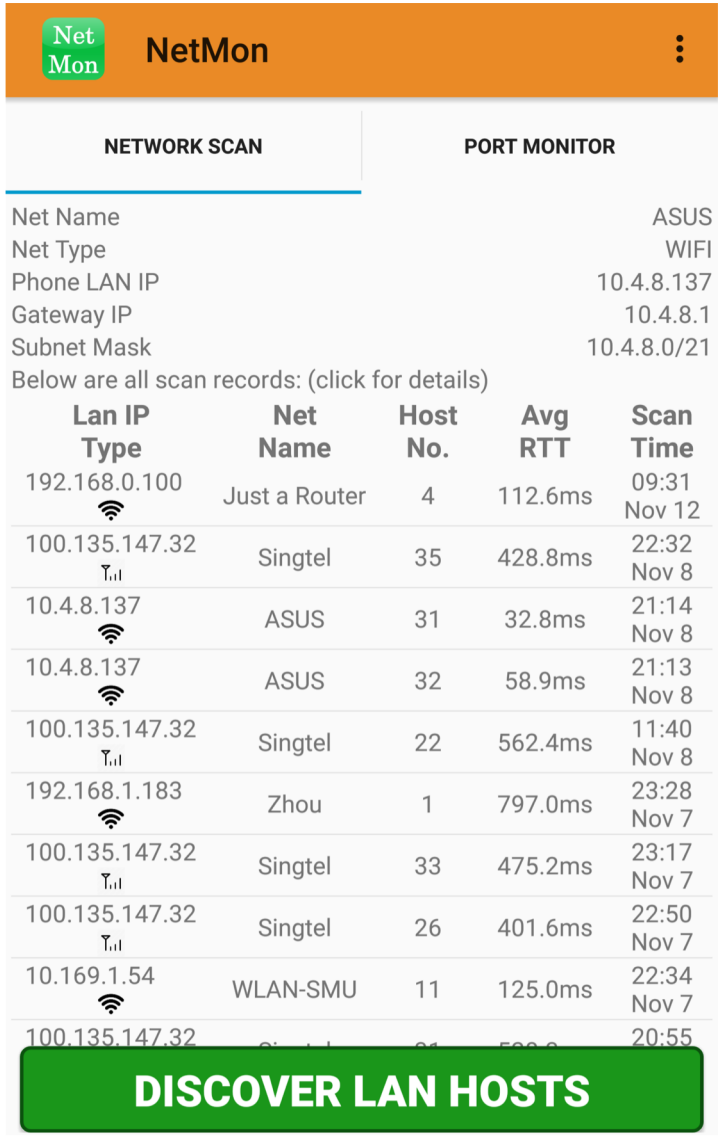


(b) YouTube's video streaming (DoS at ~32s).



(c) AirDroid's file transmission (DoS at ~13s).

Inter-device Connectivity Measurement



Remote open-port attacks require the victim device to be connected (intra- or inter-network).

6,391 network scan traces

224 cellular networks

2,181 WiFi networks

111 (49.6%)

1,823 (83.6%)

Allow intra-network connectivity (in the same network)

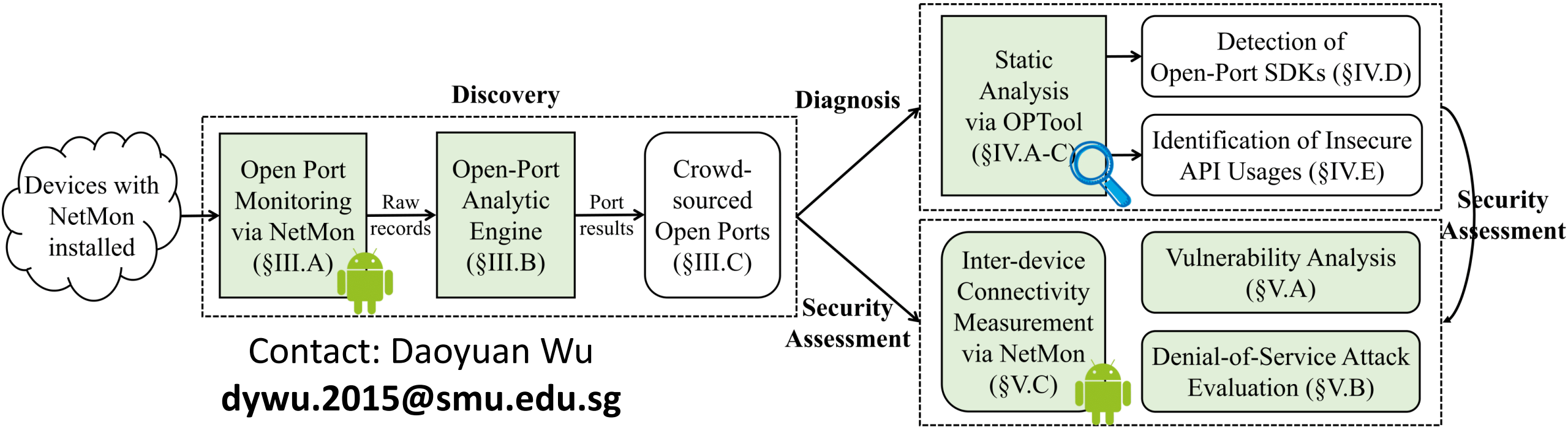
23 cellular

10 WiFi

Allow inter-network connectivity due to using public IP

Conclusion & Takeaway

- We proposed the first open-port analysis pipeline.
- We found open ports in many popular and built-in apps, and also in SDKs.
- We performed comprehensive security assessments:
  - Vulnerabilities in popular apps, DoS experiments, real connectivity measurement.



Contact: Daoyuan Wu

dywu.2015@smu.edu.sg