Review of Mobile Traffic Monitoring Paper
Haystack: A Multi-Purpose Mobile Vantage Point in User Space

Abbas Razaghpanah (Stony Brook University) Narseo Vallina-Rodriguez (ICSI)
Srikanth Sundaresan (ICSI) Christian Kreibich (ICSI / Lastline) Phillipa Gill (Stony Brook University)
Mark Allman (ICSI) Vern Paxson (ICSI / UC Berkeley)
Traffic Monitoring in Mobile Device

• Packet capture in Android or iPhone smartphone!
• How?
  • Root your smartphone???
  • Just install a mobile app
    • tPacketCapture,
• There is a possibility of implementing a user-layer app to capture packets on the smartphone!!!
  • Due to Android API
Problem

• Given a proxy API on the smartphone
• Implement
  • Proxy TCP/UDP/IP protocol stack
State-of-the-Art

- Haystack
  - ICSI--UC Berkeley and IMDEA Networks in collaboration with UMass and Stony Brook University

- AntMonitor: A System for Monitoring from Mobile Devices
  - UC Irvine

- ReCon: Revealing and Controlling PII Leaks in Mobile Network Traf
  - Northe

- Mobile Apps
  - Google Play. Packet Capture
  - Google Play. tPacketCapture
What is this “Haystack” Mobile App?

• “Application-layer” tcpdump in Android
  • Tcpdump
    • Capture packet and inspect the payload
    • Usually need “root” privilege
  • Mobile devices hacking is difficult
    • “rooting” is not popular to “average joe” users
  • Tcpdump as a mobile app!!!

• Why do we need the traffic monitoring app?
  • Many security and privacy incidents on the mobile devices
  • Monitor privacy leakage

• Android implementation
  • Mobile app in Java
Haystack Mobile App

Haystack monitoring state: ON

International Computer Science Institute (ICSI)
Berkeley, CA
http://www.icsi.berkeley.edu
System Design

- Traffic Analyzer (TA)
  - Intelligence Service, Aho-Corasick Parsers

- Forwarder
  - TLS Proxy
Ethical Considerations

• Best case
  • Do not deal with the private information
  • Collect the necessary information

• IRB at UC Berkeley
  • Need ethical consideration in the research

• SSL Decryption
  • User agreement
  • CA
Forwarder state machine

- **Forwarder**
  - TCP/UDP proxy
    - Split TCP/UDP connection
  - App ↔ Haystack ↔ Internet
- **Tunneling interface**
  - App ↔ Haystack
- **Nio interface**
  - Haystack ↔ Internet servers

**Figure 2:** Haystack’s Forwarder state machine. It controls read/write operations and transitions between tun interface, Java NIO socket, and sleep states. The idle count variable increments when both tun and NIO do not succeed, i.e., there is nothing to read. Each read operation from the tun interface potentially becomes a write operation for a NIO socket and vice versa.
TLS Interception

• Man-in-the-middle (MITM) proxy on the TLS transaction
• Need self-signed Haystack CA
  • User agreement
• Decryption
Dealing with failed TLS interception

• Failure of TLS proxy
  • Strong security policy used in app
    • E.g., facebook, google
    • Certificate pinning

• Bypass proxy
Traffic Analyzer

• Packet analysis
  • Parsing TLS, HTTP, DNS
• Off-path analysis
• Application and entity mapping
• Tracking DNS transaction
  • Non-HTTP flow: QUIC, HTTPS
Testbed and Measurement Apparatus

• Nexus 5
• 5 GHz 802.11n link (wireless access point)
• Simple UDP and TCP echo packets
• max idle cycles, idle sleep, max readtun, max readnio
CPU load

- max_idle_cycles
  - ok
- idle_sleep has impact on CPU
  - Less than 10~25ms
- Optimal idle_sleep
  - 100ms

Figure 3: Haystack’s CPU overhead for different max_idle_cycles and idle_sleep configurations. The horizontal line indicates the aggregated average CPU load of all apps running on the background for reference.
Power consumption

• Monsoon Power Monitor
• worst case
  • max_idle_cycles : 100
  • idle_sleep = 1ms
• 3-9% power usage increase

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Power(mW) Mean/SD</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>1,089.5 / 125.9</td>
<td>+3.1%</td>
</tr>
<tr>
<td>Idle (Haystack)</td>
<td>1,123.8 / 150.4</td>
<td></td>
</tr>
<tr>
<td>YouTube</td>
<td>1,755.3 / 35.5</td>
<td></td>
</tr>
<tr>
<td>YouTube (Haystack)</td>
<td>1,914.4 / 16.1</td>
<td>+9.1%</td>
</tr>
</tbody>
</table>

Table 2: Power consumption of Haystack when max_idle_cycles is 100 cycles and idle_sleep is 1 ms in different scenarios. The percentage indicates the increase when running Haystack.
Throughput of Haystack
TLS Performance in Haystack

(a) TLS session establishment time.

(b) TLS download speeds.
Summary

- Traffic monitoring for security in mobile device
- Need user space tool
  - Do not use “rooting”!

- Android
  - Local VPN Java class by Google

- iOS
  - Network extension library by Apple