STAAF
An Efficient Distributed Framework for Performing Large-Scale Android Application Analysis

OWASP AppSec USA
Thursday, September 22, 2011
Allow Me to Introduce Myself

Ryan W Smith
VP Engineering at Praetorian

- OWASP DFW Chapter Leader (2011)
- Active member of The Honeynet Project (2002– )
- 8+ years of work with DoD, Intelligence Community, Federal/State/Local governments, and Fortune 500 companies
WE ACT AS TRUSTED ADVISORS WHO HELP ORGANIZATIONS BETTER UNDERSTAND AND MINIMIZE OVERALL RISK ACROSS I.T. ASSETS, SO THEY CAN FOCUS ON WHAT’S IMPORTANT - THEIR CORE BUSINESS.

Software Security
Evaluate your application’s security over its entire development lifecycle

Security Research
Leverage outside expertise to solve advanced problems

Infrastructure Security
Measure the overall strength of your company’s security program

Security Training
Learn online, and in a classroom environment, from the experts

www.praetorian.com
Your World, Secured
Presentation Roadmap

• STAAF (Overview)
• Background
• STAAF (Deep Dive)
• Results
• Future Work
• Conclusions
What can STAAF do for you?

Observation #1:
There are a lot of Android app analysis tools freely available

BUT:
They’re typically designed for single app analysis

STAAF leverages the power of these tools as modules,
And adds efficiency, scalability, data mgmt and sharing
What can STAAF do for you?

Observation #2:
Higher value analysis can be attained by analyzing large numbers of applications over long periods of time

SOLUTION:
Reduce the time and complexity for an analyst to process large numbers of apps

Goal
Analyze 50k apps in less than 2 days and make the extracted data readily available to analysts
What can STAAF do for you?

Minimize analysts’ effort to extract meaningful results from a large number of applications
What is STAAF

SCALABLE
TAILORED
APP ANALYSIS
FRAMEWORK

STAAF
SCALABLE TAILORED APP ANALYSIS FRAMEWORK
What is STAAF

SCALABLE
TAILORED
APP ANALYSIS
FRAMEWORK
What is STAAF

SCALABLE
TAILORED
APP ANALYSIS
FRAMEWORK
What is STAAF

SCALABLE
TAILORED
APP ANALYSIS
FRAMEWORK

HACKER SHIELDS ON FULL POWER
I’LL USE SOME MAGIC
What is STAAF
What STAAF is NOT

- STAAF is not a stand alone application
- STAAF is not only a malware detection or anti-virus engine
- STAAF is not an application collection tool

STAAF is a problem agnostic app analysis framework
Presentation Roadmap

• STAAF (Overview)
• Background
• STAAF (Deep Dive)
• Results
• Future Work
• Conclusions
Android’s Open App Model

- Low barrier to entry
- Apps hosted and installed from anywhere
- All apps are created equal
- No distinction between core apps and 3rd party apps
- Accept apps based on:
  1. Trust of the source
  2. Permissions requested
“Legitimate” Monitoring Apps

- Ad/Marketing Networks
- Social Gaming Networks
“Not-So-Legitimate” Permission Use

**SMS Trojan**
- Link to site hosting rogue app for “free movie player”
- Sends 2 Premium SMS messages to a Kazakhstan number (about $5 per message)

**Gemini**
- Repackaged apps in Chinese markets
- Sex positions and MonkeyJump2 are known examples
  - Central C&C
  - Exfiltrates unique device identifiers
  - Downloads and Install New Apps (with permission)

**DroidDream**
- Approx. 50 Malicious apps in official market
  - Central C&C
  - Exfiltrates unique device identifiers
  - Downloads additional code modules
Presentation Roadmap

- STAAF (Overview)
- Background
- **STAAF (Deep Dive)**
- Results
- Future Work
- Conclusions
STAAF Workflow

Step 0: STAAF components initialized
STAAF Workflow

**Step 1:** Users sends APKs to be processed
STAAF Workflow

Step 2: Coordinator checks database for previous results and logs new instance data for each APK
**STAAF Workflow**

**Step 3:** Coordinator sends new APKs to the file repository service
STAAF Workflow

**Step 4:** Coordinator sends tasking orders to the task queue
STAAF Workflow

**Step 5:** Elastic computing nodes pull tasks from their designated task queue
STAAF Workflow

**Step 6:** Elastic computing nodes pull in the APK and related information
STAAF Workflow

Step 6: After processing the elastic computing nodes push out processed files and analysis results
STAAF Workflow

Step 7: When all tasking is complete elastic computing nodes notify the coordinator
Task Modules

- Can be registered dynamically
- Task-Oriented
  - High level
    - What % of apps use permission X
    - What is the most common libraries used
  - Mid level
    - Extract Permissions
    - Extract static URLs
    - Extract Methods Called
  - Low level
    - Extract manifest
    - Extract Dex bytecode
Deduplication of Effort

- All Intermediate data are cached for later use
  - Extract and convert manifest to ASCII
  - Extract Dex and convert to Smali and Java
  - Compute the control flow graph from the Dex
- Libraries and shared resources must only be processed once
- Apps must only be processed once by each module, ever

Small savings matter at large scales
Distributed Data Sharing

• Sharing app samples is just the beginning
• Share the entire process:
  – Raw Application
  – Extracted Resources
  – Raw Data
  – Processed Data
• Or set specific limits on what data is shared
Presentation Roadmap

• STAAF (Overview)
• Background
• STAAF (Deep Dive)
• **Results**
• Future Work
• Conclusions
## Time Trials

### STAAF Performance Tests

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECU</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

![Graph showing performance across different number of nodes and ECUs]

**Achieved 50k apps in ~7 hours**

*Extrapolated from shorter tests*
### STAAF Performance Tests

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECU(s)</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

“One EC2 Compute Unit (ECU) provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.” -Amazon
## Time Trials

### STAAF Performance Tests

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECU(s)</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

STAAF is bound by both CPU and database throughput.
## Time Trials

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECUs</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

By using distributed, local databases STAAF achieves a significant time performance increase.
# Time Trials

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECU</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

Using by adding multiple processors with local databases, we achieve near linear scalability.
By simply increasing the CPU capacity to 5 ECUs, we achieve the same performance as four 1 ECU nodes.
Time Trials

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECU</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

Once again, using a central database fails to achieve linear performance gains.
## Time Trials

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECUs</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

By using distributed, local databases we once again achieve near linear performance gains.
# Time Trials

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECU</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

By increasing CPU capacity, number of processing nodes, and number of databases, we decreased processing time by 14.5x.
Time Trials

<table>
<thead>
<tr>
<th>#</th>
<th>Time</th>
<th>Apps</th>
<th>ECU</th>
<th>Nodes</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2h25m</td>
<td>500</td>
<td>1</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>2</td>
<td>2h00m</td>
<td>500</td>
<td>1</td>
<td>2</td>
<td>Central</td>
</tr>
<tr>
<td>3</td>
<td>1h56m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>4</td>
<td>0h36m</td>
<td>500</td>
<td>1</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>5</td>
<td>0h36m</td>
<td>500</td>
<td>5</td>
<td>1</td>
<td>Central</td>
</tr>
<tr>
<td>6</td>
<td>0h28m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Central</td>
</tr>
<tr>
<td>7</td>
<td>0h10m</td>
<td>500</td>
<td>5</td>
<td>4</td>
<td>Local</td>
</tr>
<tr>
<td>8</td>
<td>0h27m</td>
<td>1722</td>
<td>5</td>
<td>5</td>
<td>Local</td>
</tr>
<tr>
<td>9</td>
<td>1h19m</td>
<td>9349</td>
<td>5</td>
<td>10</td>
<td>Local</td>
</tr>
</tbody>
</table>

Larger tests confirm that STAAF continues to scale linearly

1722 Apps
Compute Time
0h27m

9349 Apps
Compute Time
1h19m
Initial Results :: Permissions Requests

53,000 Applications Analyzed
- Android Market: ~48,000
- 3rd Party Markets: ~5,000

Permissions Requested
- Average: 3
- Most Requested: 117

- Location Data: 11,929 (24%)
- Read Contacts: 3,636 (8%)
- Send SMS: 1,693 (4%)
- Receive SMS: 1,262 (4%)
- Record Audio: 1,100 (2%)
- Read SMS: 832 (2%)
- Process Outgoing Calls: 323 (1%)
Additional Results :: Shared Libraries

53,000 Applications Analyzed

- **Android Market:** ~48,000
- **3rd Party Markets:** ~5,000

- com.admob 38% (18,426 apps)
- org.apache 8% (3,684 apps)
- com.google.android 6% (2,838 apps)
- com.google.ads 6% (2,779 apps)
- com.flurry 6% (2,762 apps)
- com.mobclix 4% (2,055 apps)
- com.millennialmedia 4% (1,758 apps)
Permissions Are Not a Good Indicator

Malware only needs a single permission
Presentation Roadmap

• STAAF (Overview)
• Background
• STAAF (Deep Dive)
• Results
• Future Work
• Conclusions
STAAF’s Future

• Build a publically available user interface
• Provide a dashboard with global stats
• Further Tune database performance issues
• Build more complex analysis modules
  – Static data flow analysis
  – Dynamic sandbox analysis
• Expose a public module interface through UI
Presentation Roadmap

• STAAF (Overview)
• Background
• STAAF (Deep Dive)
• Results
• Future Work
• Conclusions
Final Thoughts

• STAAF is a system of systems and services, not an application
• STAAF enables large scale Android application analysis
• STAAF is problem agnostic and can be tailored to answer many analytic questions
• STAAF augments the capabilities of the analyst, it does not replace them
• STAAF achieves scalable performance increases by increasing computer nodes/power
Q&A

IS YOUS DELICIOUS?

I HOPES SO