• SOFTWARE ENGINEERING and ANALYSIS
  Study and design of methodologies that support the development and maintenance of complex software systems

• SECURITY AND PRIVACY
  • malware (e.g., virus, trojans, backdoors…)
  • disclosure of sensitive data in the internet
  • network security (protocols)
  • web security (code injection, cross site scripting)
  • intellectual property protection (software piracy)
  • software integrity
  • …
Security
SW Protection and Malware
SW Protection

Untrusted Host

MATE attack
The Value of SW Protection

Tamper Detection

Tamper Resistance

Hide algorithms and computations

Hide internal data

Hide proprietary information

Identify piracy

Protect the intellectual property
SW Protection Techniques

- Tamper Detection
- Tamper Resistance
- Tamper-proofing
- Code Obfuscation
- SW Watermarking

- Identify piracy
- Protect the intellectual property
- Hide proprietary information
- Hide algorithms and computations
- Hide internal data
SW Protection Techniques

- Tamper Detection
- Tamper Resistance
- Tamper-proofing
- Code Obfuscation
- SW Watermarking
- Protect the intellectual property
- Identify piracy
- Hide algorithms and computations
- Hide internal data
- Hide proprietary information

General Semantics-based framework

TODO
the number of Android malware families added in 2015 grew by 6%, compared with the 20% growth in 2014

the number of Android malware variants added in 2015 grew by 40%, compared with the 29% growth in 2014

[Symantec 2016]
* clone detection
* software forensics
* plagiarism detection
* tamper detection
* software birth-marking
* malware detection
* vulnerability detection
In order to identify malware variants, plagiarized code, tampered code and vulnerabilities in different application, we need to extract semantic models!

Develop a semantics-based similarity analysis!
Similarity Analysis

R.E.H.A. - General Workflow

Niccolò Marastoni

Feature Extraction → Models → Similarity Matching → Behaviors

apk

DB → Models
Similarity Analysis

Family Analysis Workflow

R.E.H.A.

Feature Extraction → Models → Clustering → Behaviors

Features → DB → Clusters
• Static analysis for the detection of race condition on Concurrent Java programs.

• Verification and inference of locking policies (@GuardedBy and @Holding annotations) and relative formal semantics

• Verification and inference of thread-confinement properties (@UiThreada and @WorkerThread annotations)
Hyperproperty: Property that can be verified on sets of execution instead of on single executions.
Hyperproperty Verification

**Hyperproperty:** Property that can be verified on sets of execution instead of on single executions.

Characterize a framework for verifying hyperproperties
SQL injection

SQL injection example

query = “SELECT Username, UserID, Password
FROM Users WHERE
Username = ‘bob’; DROP Users--
‘AND Password = ‘ ‘”

Our idea for releasing **secured** applications
// Retrieve the ID for a camera
int cameraId = ...;

// Create an obfuscated string
// containing the method call
String obfuscated = “AoApBeBnBA”;  
String deobfuscated = obfuscated.replaceAll(“[AB],”);  

Class<?> klass = Class.forName(  
“android.hardware.Camera”  
);  

// Retrieve and invoke the method
Method method = klass.getMethod(  
deobfuscated,   
Integer.class  
);  

Camera camera = (Camera) method.invoke(cameraId);  

x := ...
...
manipulation of x
...  
Eval (x)
Thanks