IOWA STATE UNIVERSITY

Secure Code Changes

Lotfi ben Othmane and Moataz Abdelkhalek

Iowa State University, USA

2018 Annual Computer Security Applications Conference
December 6, 2018
San Juan, Puerto Rico

Problem

- Companies develop and deploy frequent releases of their software.

- PB: How to maintain the security of the software when the code changes?

```
public class MyClass
{
public void Method_A()
    {  // Do Something
        FileIOPermission myPerm =
            new FileIOPermission(PermissionState.Read);

        myPerm.Demand();
        // Do Something
    }}
```

Change to:

Unrestricted

Current solutions

- Companies use two common approaches:
  - Perform full security assessment of the software in each new release.
  - Use keywords, e.g., encrypt, secure, hash.
  - Use notations and peer-review of all the changes by the senior developers .

- Both approaches are:
  - Impractical
  - Time-consuming
  - Expensive
  - Cause conflicts based on the different understandings of software architecture.

Research Question

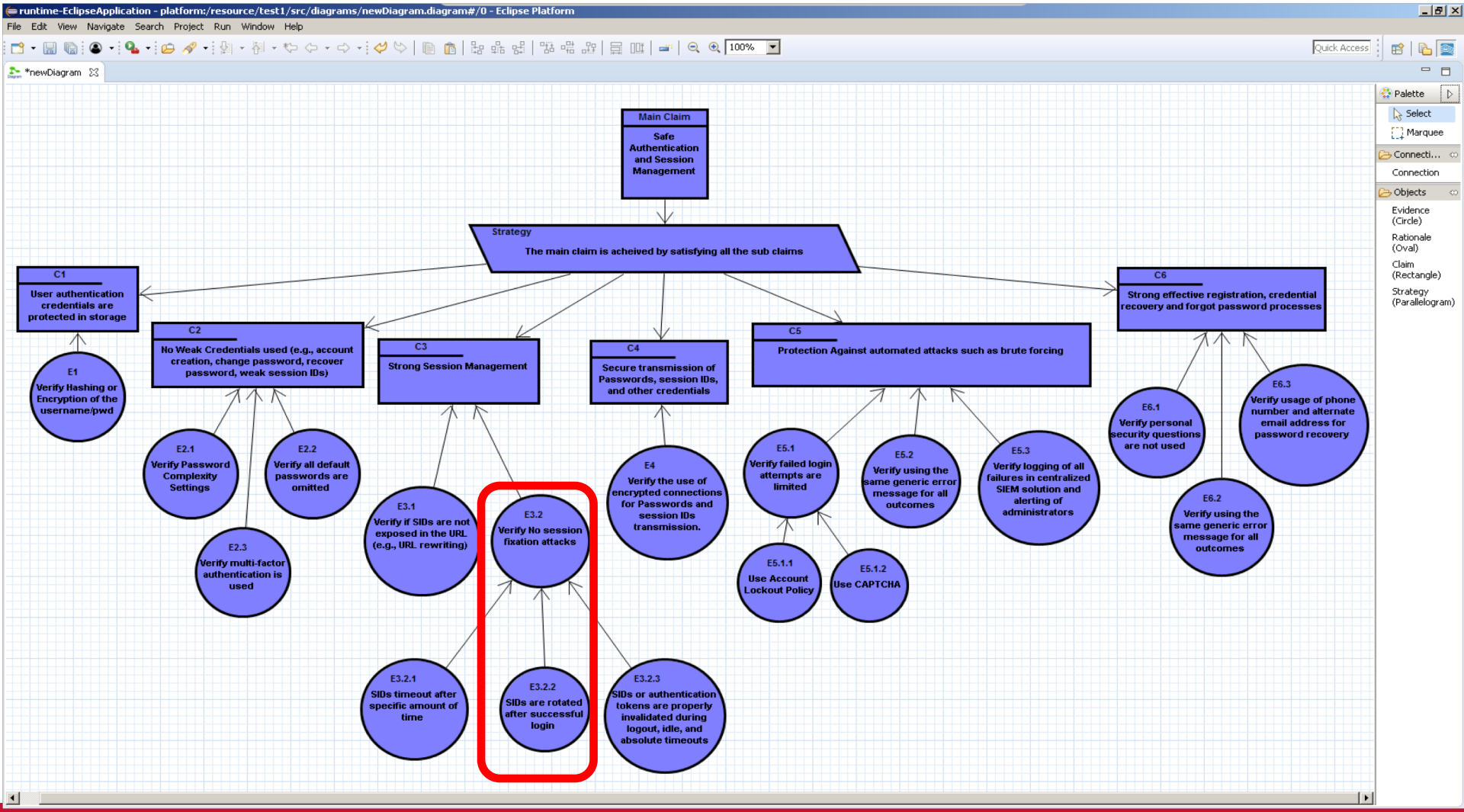
How to trace the impacts of code changes on the security of a given software?

Approach

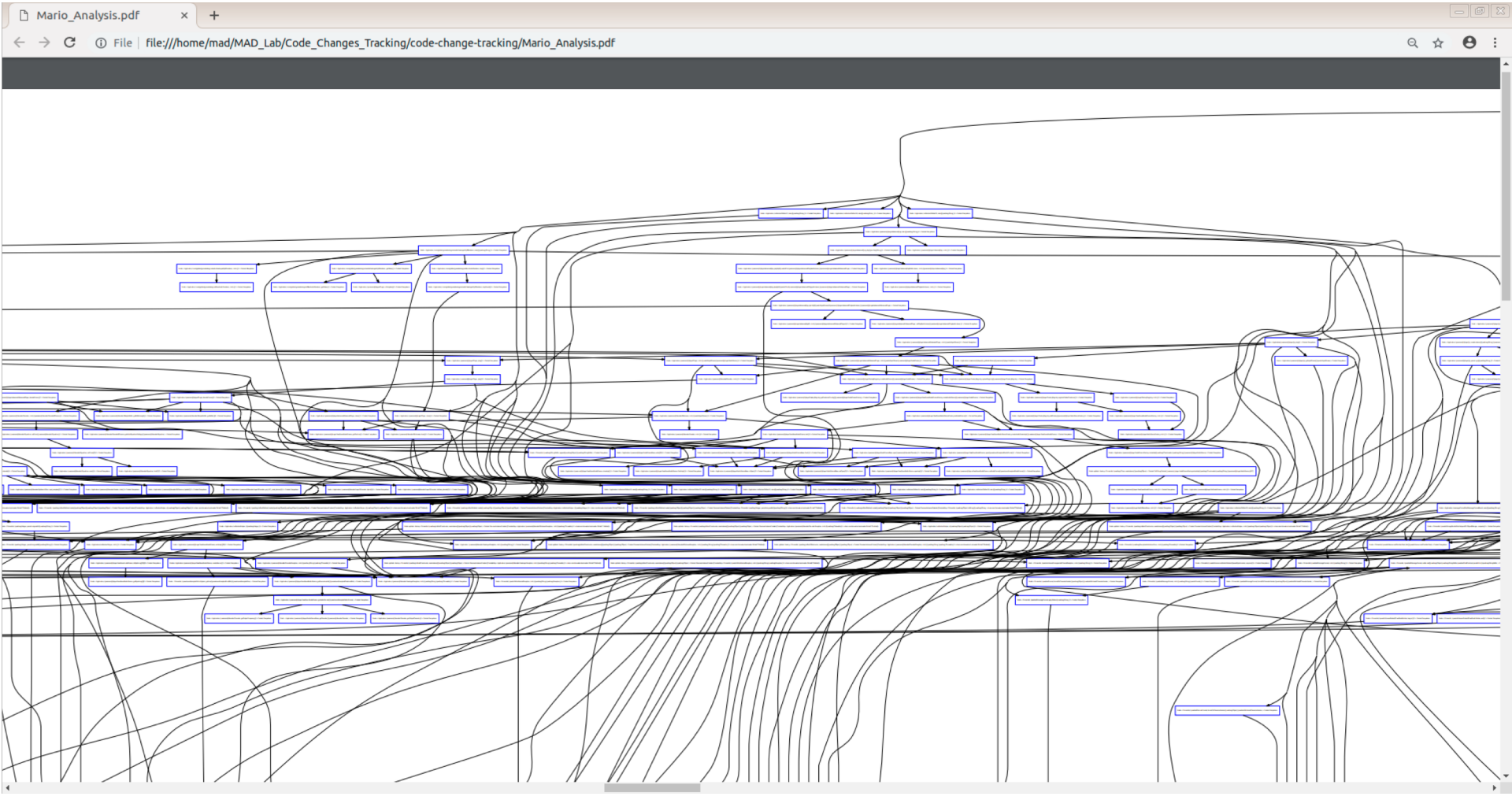
1. Model the security assurance of the software using security assurance cases.
2. Associate code parts to security claims/requirements.
3. Related code changes to attack surface entry points.

=> Relate assurance case elements to code changes.

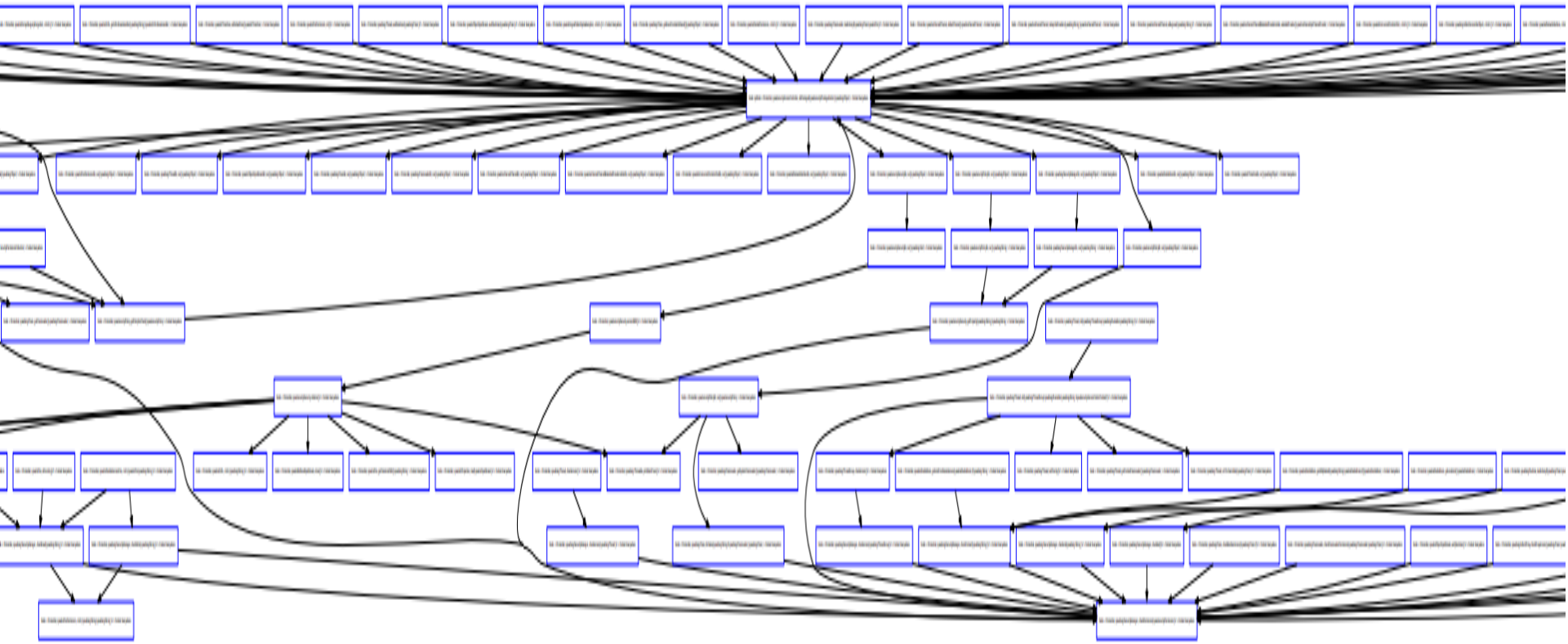
Security Assurance Case



Example of Call Graph -- WALA



Call Graph – Security Functions



Example Claim – Strong Session Management

1. Verify if SIDs are not exposed in the URL (e.g., URL rewriting)
2. Verify No session fixation attacks
3. SIDs timeout after specific amount of time
4. SIDs are rotated after successful login
5. SIDs or authentication tokens are properly invalidated during logout, idle, and absolute timeouts.

Related nodes

- compiere/process/SessionEndAll, main
- compiere/process/SessionEndAll, clinit

Any Comments!

Thank you

Lotfi Ben Othmane
othmanel@iastate.edu

Example of Call Graph -- WALA

```
digraph "DirectedGraph" {
graph [concentrate = true];center=true;fontsize=6;node [ color=blue,shape="box"fontsize=6,fontcolor=black,fontname=Arial];edge [ color=black,fontsize=6,fontcolor=black,fontname=Arial];
   "Node: synthetic < Primordial, Ljava/security/AccessController, doPrivileged(Ljava/security/PrivilegedAction;)Ljava/lang/Object; > Context: Everywhere" [ label="Node: synthetic < Primordial, Ljava/security/AccessController, doPrivileged(Ljava/security/PrivilegedAction;)Ljava/lang/Object; > Context: Everywhere"]
   "Node: synthetic < Primordial, Ljava/security/AccessController, doPrivileged(Ljava/security/PrivilegedExceptionAction;)Ljava/lang/Object; > Context: Everywhere" [ label="Node: synthetic < Primordial, Ljava/security/AccessController, doPrivileged(Ljava/security/PrivilegedExceptionAction;)Ljava/lang/Object; > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager, checkPermission(Ljava/security/Permission;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager, checkPermission(Ljava/security/Permission;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/net/URL, checkSpecifyHandler(Ljava/lang/SecurityManager;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/net/URL, checkSpecifyHandler(Ljava/lang/SecurityManager;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/AccessController, checkPermission(Ljava/security/Permission;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/AccessController, checkPermission(Ljava/security/Permission;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager, checkPackageAccess(Ljava/lang/String;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager, checkPackageAccess(Ljava/lang/String;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager, checkAccess(Ljava/lang/Thread;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager, checkAccess(Ljava/lang/Thread;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager, checkRead(Ljava/lang/String;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager, checkRead(Ljava/lang/String;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager, checkWrite(Ljava/lang/String;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager, checkWrite(Ljava/lang/String;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/Thread, init(Ljava/lang/ThreadGroup;Ljava/lang/Runnable;Ljava/lang/String;JLjava/security/AccessControlContext;Z)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/Thread, init(Ljava/lang/ThreadGroup;Ljava/lang/Runnable;Ljava/lang/String;JLjava/security/AccessControlContext;Z)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/SecureRandom, getSeed(I)[B > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/SecureRandom, getSeed(I)[B > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/SecureRandom, <init>()V > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/SecureRandom, <init>()V > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/SecureRandom, getDefaultPRNG(Z[B)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/SecureRandom, getDefaultPRNG(Z[B)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager$1, run()Ljava/lang/Object; > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager$1, run()Ljava/lang/Object; > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/AccessControlContext, checkPermission(Ljava/security/Permission;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/AccessControlContext, checkPermission(Ljava/security/Permission;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager, checkConnect(Ljava/lang/String;I)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager, checkConnect(Ljava/lang/String;I)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/PrivilegedActionException, toString()Ljava/lang/String; > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/PrivilegedActionException, toString()Ljava/lang/String; > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager$1, run()Ljava/lang/String; > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager$1, run()Ljava/lang/String; > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager, checkLink(Ljava/lang/String;)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager, checkLink(Ljava/lang/String;)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/lang/SecurityManager, checkExit(I)V > Context: Everywhere" [ label="Node: < Primordial, Ljava/lang/SecurityManager, checkExit(I)V > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/Security, getProperty(Ljava/lang/String;)Ljava/lang/String; > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/Security, getProperty(Ljava/lang/String;)Ljava/lang/String; > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/Security, <clinit>()V > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/Security, <clinit>()V > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/AccessControlContext$1, run()Ljava/lang/Object; > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/AccessControlContext$1, run()Ljava/lang/Object; > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/Security$1, run()Ljava/lang/Object; > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/Security$1, run()Ljava/lang/Object; > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/AccessControlContext$1, run()Ljava/lang/Void; > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/AccessControlContext$1, run()Ljava/lang/Void; > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/Security$1, run()Ljava/lang/Void; > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/Security$1, run()Ljava/lang/Void; > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/Security, access$000()V > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/Security, access$000()V > Context: Everywhere"]
   "Node: < Primordial, Ljava/security/Security, initialize()V > Context: Everywhere" [ label="Node: < Primordial, Ljava/security/Security, initialize()V > Context: Everywhere"]
   "Node: synthetic < Primordial, Ljava/security/AccessController, doPrivileged(Ljava/security/PrivilegedAction;Ljava/security/AccessControlContext;)Ljava/lang/Object; > Context: Everywhere" [ label="Node: synthetic < Primordial, Ljava/security/AccessController, doPrivileged(Ljava/security/PrivilegedAction;Ljava/security/AccessControlContext;)Ljava/lang/Object; > Context: Everywhere"]
```

Graphviz Dot ▾ Tab Width: 4 ▾ Ln 268, Col 2 INS