THE UNIVERSITY OF UTAH

SeaCat: an SDN End-to-end Application Containment ArchitecTure

Enabling Secure Role Based Access To Sensitive Healthcare Data

Junguk Cho, Makito Kano, Brent Elieson, David Johnson and Kobus Van der Merwe

Funded by NSF award CNS-1343713

Motivation

Motivation

- **"Everything" is networked**
  - Nearly all business applications assume network availability
- **Also true in healthcare**
  - Accessing patient records
  - Remote diagnoses and consultation
  - In-home monitoring
  - Healthcare analytics
  - Plus "regular" vocational applications
    - HR/payroll functions, accessing domain specific literature

Problem

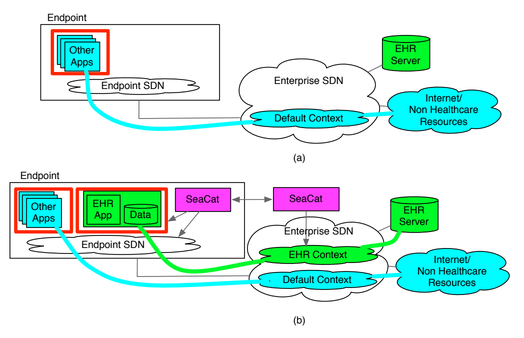
- **Individuals act in different roles**
  - Often using same device
- **Apps have different security and performance constraints**
  - Healthcare records: stringent privacy and security requirements
  - In-home patient monitoring: privacy, security needs + reliability and soft real time guarantee
- **Devices increasingly mobile**
  - Often unmanaged and untrusted
- **Generalizes to broad range of sensitive data access/management**
  - HIPAA, FERPA, FISMA, PCI-DSS

Current Approaches

- **Scan device when attaches to network**
  - Device with up-to-date patch levels might still contain malware
- **Thin clients**
  - Application servers with thin clients constrain the type of applications that can be used
- **Complex network and server access control polices**
  - Access control policies only deal with access
  - No protection once data is accessed

SeaCat Approach

- **Combine SDN and application containment:**
  - End-to-end application containment
- **Treat mobile device as "semi-trusted" SDN domain**
  - Inter-domain SDN interaction to tie in
- **Non-healthcare apps:**
  - Default context: endpoint container and separate network
- **Healthcare app:**
  - Dynamic app specific context
  - App and data contained in this end-to-end context



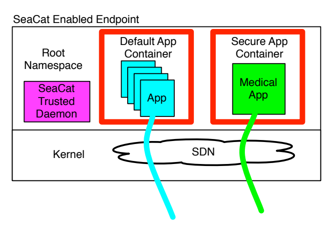
Threat Model

- **Concern: security and performance of health care applications**
  - Including apps on mobile devices
- **Assume healthcare applications can be trusted**
- **Specific concerns:**
  - Unauthorized access
  - Data leakage
  - Resource guarantees
  - Denial of service

SeaCat Architecture

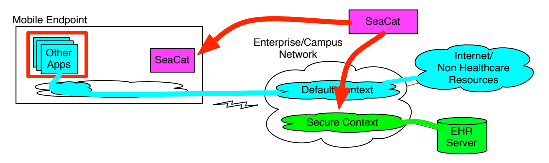
Endpoint Containment

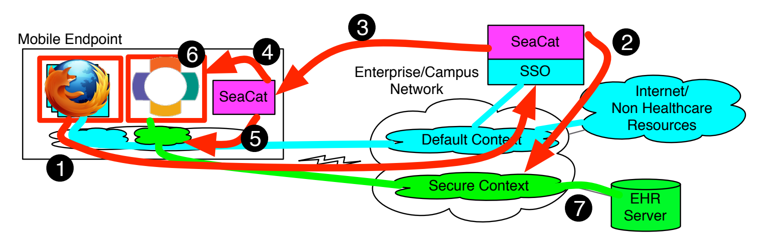
- Uses lightweight containers
- "Regular apps" in default container
- Minimize trusted computing base in root namespace
- SeaCat Trusted Daemon:
  - **Dynamically creates secure app container(s)**
  - **Manages endpoint SDN domain**



Enterprise Network Containment

- SeaCat Server:
  - **Manages enterprise SDN domain**
  - **Interacts with SeaCat trusted daemon in endpoint**





End-to-end

- Mobile endpoint: semi-trusted SDN domain
- **SeaCat server integrated with SSO**
  - Successful authentication triggers:
    - Creation of app specific SDN context in enterprise
    - Signaling to endpoint SDN to:
      - Create secure container
      - Create endpoint app specific SDN context
- App and data remains in secure context
- When app exits:
  - Complete context is destroyed