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

Enabling Secure Role Based Access To Sensitive Healthcare Data

Junguk Cho, David Johnson, Makito Kano, Kobus Van der Merwe and Brent Elieson
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
  – Plus non vocational use
    ● Browsing the web, social networking etc.
Motivation cont.

- Problem:
  - Same individual, using same device potentially using several of these applications simultaneously
  - Applications have very different security and performance constraints:
    - Healthcare records: stringent regulatory privacy and security requirements
    - In-home patient monitoring: different privacy and security needs + reliability and soft real time guarantees
    - Web use: no impact on core healthcare applications
  - Devices are increasingly mobile (tablets, laptops, smartphones)
    - Often not part of managed and trusted enterprise environment
Motivation cont.

• Current approaches, combinations of:
  – Device scans when new devices attach to network
  – Run applications on application servers with thin clients on devices
  – Complex network and server access control polices

• Inadequate:
  – Device with up-to-date patch levels might still contain malware
  – Application servers with thin clients constrain the type of applications that can be used
  – Access control policies only deal with access. Provide no protection once data is accessed
Motivation cont.

• Problem generalizes to broad range of access to sensitive data
• Different sets of regulations/practices
  – Protected health information (PHI)
    • HIPAA regulations
  – Student educational records
    • FERPA regulations
  – Federal government work
    • FISMA regulations
  – Business requirements
    • PCI DSS regulations
  – Institutional requirements
    • IRB regulations
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

• Healthcare app:
  – dynamic app specific context
  – app and data contained in this end-to-end context
Threat Model

• Concerned with security and performance of health care applications used from variety of devices in a health care environment

• Assume healthcare applications can be trusted
  – different from conventional threat model where device needs to be protected against untrusted applications

• Specific concerns:
  – Unauthorized access
    • role based authentication and policies
  – Data leakage
    • end-to-end application containment
  – Resource guarantees
    • context based resource allocation with preemption
  – Denial of service
    • resource guarantees plus separation of resources
SeaCat Architecture:
Endpoint Containment

- Uses lightweight containers
  - Linux containers
- All applications execute in containers:
  - move “regular apps” into default container
- Only SeaCat Trusted Daemon left in root namespace
SeaCat Architecture:
Endpoint Containment

- SeaCat Trusted Daemon manages containers:
  - Set default container up: apps unaware that anything changed
  - Use Overlay FS to restrict container storage accesses
  - Dynamically create secure app container(s)
SeaCat Architecture: Endpoint Network Containment

- **SeaCat Trusted Daemon:**
  - Manages endpoint SDN domain
- **Single switch domain:**
  - Sets up context for default apps
  - Sets up context for secure apps: based on interaction with enterprise SDN
SeaCat Architecture:
Enterprise Network Containment

• SeaCat Server:
  – **Manages enterprise SDN domain**
    • Sets up context for secure apps
    • Includes SDN-enabled WiFi
  – **Interacts with SeaCat trusted daemon in endpoint**
    • Instructs trusted daemon to start secure container
    • Coordinates SDN across domains
SeaCat Architecture: Putting it all together

- Enterprise network treats each mobile endpoint as semi-trusted SDN domain
- Secure app user: authenticates using “normal” single-sign-on (SSO) technology
  - **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
      - Ties to enterprise SDN context
- App and data remains in this secure end-to-end context
- When app exits:
  - Complete context is destroyed
SeaCat Workflow/Interaction
SeaCat Workflow/Interaction
SeaCat Workflow/Interaction
SeaCat Workflow/Interaction
SeaCat Workflow/Interaction
SeaCat Workflow/Interaction
SeaCat Workflow/Interaction
SeaCat Demo

• Mobile endpoint:
  – Linux WiFi-enabled tablet
  – With SeaCat Trusted Daemon:
    • Container and SDN management

• Enterprise network:
  – SDN enabled WiFi access point
    • Tallac Networks
    • Virtual APs
    • Mapped to OpenFlow switch
  – Rest of enterprise SDN emulated in a Mininet instance

• SSO:
  – Uses Shibboleth SSO
  – SeaCat (Service Provider) to realize SeaCat functionality

• Medical application:
  – OpenMRS (Medical Record System)
Status

• Have working prototype...

• Looking for partners to do a trial deployment...