Monitoring-Based System for E2E Security Auditing and Enforcement in Trusted and Untrusted SOA & Privacy-Preserving Data Dissemination and Adaptable Service Compositions in Trusted and Untrusted Cloud

7 October 2015

Bharat Bhargava
Purdue University
Collaboration with NGC

Weekly meeting to install software at NGC server, respond to NGC client’s questions and advance research based on vision of Donald D Steiner and integrate work at MIT (Harry Haplin) and Purdue (Patrick Eugster).

Researchers at NGC

Leon Li

Jonathan E Fulkerson

Sunil Lingayat

Jason C Kobes
Outline

- Problem Statement
- Benefits of Proposed Research
- State of the Art
- Year 6 Final Report
  - Methodology
  - Results
  - Impact
  - Demo
- Year 7 Proposal
  - Technical Approach Overview
  - Distributed Service Monitoring for Adaptability and Resiliency
  - Active Bundles for Secure Data Dissemination
  - Proposed Deliverables
  - Collaboration
Problem Statement

• When clients interact with a cloud service, they expect certain levels of Quality of Service (QoS) guarantees, expressed as service performance, security and privacy policies. Controlling compliance with service level agreements (SLAs) requires continuous monitoring of services in an enterprise, as sudden changes in context can cause performance to deteriorate, if not result in the failure of a whole composition. To provide optimal performance in an enterprise cloud architecture under varying contexts, we need context-awareness and adaptation mechanisms for SOA and cloud service domains.

• Cloud platforms are vulnerable to increasingly complex attacks that could violate the privacy of data stored on them or shared with web services, which is especially detrimental in case of mission-critical operations. In order to mitigate these problems, cloud enterprise systems need to integrate proactive defense mechanisms, which provide increased resiliency by treating potentially malicious service interactions and data sharing before they take place.
There is a need for novel techniques to
- **Monitor** service activity
- Discover and report service **misbehavior**
- Share information across domains on security threats using a **unified model**
- Ensure **security** and **privacy** of data in SOA and clouds

If a service is compromised or misbehaves, the service monitor should
- **Discover** malicious activity
- Provide **feedback**
- Take **remedial actions**
- **Adapt** according to changes in **context**
Benefits of Proposed Research

This research proposes a novel method of dealing with security and adaptability problems in SOA and cloud:

- Provides **enforcement of security policies** in addition to auditing capability
- Offers a **proactive** security solution by detecting **anomalies** across individual service interactions and at the domain level
- Ensures **uninterrupted service** even under attacks and failures
- Provides a **transparent** mechanism for service monitoring that can be used in standard web service frameworks
- Provides context-based **adaptable** data dissemination **independent** of third party data and policy management
Applications and Demonstration of Prototypes

- Medical data dissemination
- Cross-domain data dissemination based on clearance levels of top secret, secret, classified and unclassified
- UAV images
State of the Art

- Industry-standard cloud systems such as Amazon EC2 provide coarse-grain monitoring capabilities (e.g. CloudWatch)
- Splunk (expensive log management and analysis tool) GrayLog, Kibana:
  - Provides capabilities to store, search and analyze big data gathered from various types of logs on enterprise systems
  - Enables organizations to detect security threats through examination by system administrators
  - Disadvantage: Requiring human intelligence for detection of threats; need to be complemented with automated analysis and accurate threat detection capability to provide increased resiliency

- DataSafe: Software-hardware architecture supporting confidentiality throughout data lifecycle
  - Special architecture limited to well-known hosts
- Encore: Sticky policies to manage privacy of shared data across domains
  - Prone to TTP related issues, sticky policies vulnerable to attacks from malicious recipients
• Service monitor intercepts all client-service/service-service interactions.
• Unified security architecture for SOA and cloud with components for:
  • *Service trust management*
  • *Interaction authorization* between different services
  • *Dynamic service composition* based on changes in context
  • *Secure data dissemination* using active bundles
2014/2015 Methodology

- A novel secure data dissemination approach
  - Self-protection mechanism for data
  - Policy-based access control
  - Independent of third party data and policy management
  - Ability to operate in external environment
  - Agnostic to policy language and evaluation engine

- A novel service monitoring approach for adaptability
  - Distributed service monitoring with request interceptors placed at every service domain
  - Service interactions logged in local database
  - Service trust values maintained at central database
  - Efficient dynamic service composition reconfiguration algorithm
2014/2015 Results Overview

- Low overhead of dynamic service composition
  - Experiments:
    - Dynamic composition time vs. number of service categories
    - Dynamic composition time vs. number of services
- Low overhead of data protection with active bundles
  - Experiments:
    - AB size vs. number of policies
    - AB interaction time vs. number of policies
    - Tamper resistance overhead
- Demo implementations based on simulations
- Integration discussions with two other NGCRC projects & demo at NGC TechFest (June’15)
Dynamic Service Composition Experiment 1:

- Dynamic service composition overhead is especially important in time-critical settings
- **Task**: Dynamic composition of business process involving varying number of service categories each with 3 services to choose from, with a single SLA constraint
- **Measurement**: Response time overhead of dynamic service composition for different number of service categories involved in composition
- **Setting**: Central service monitor on Amazon EC2 m3.medium instance (1 vCPU, 3.75 GB memory)
- **Conclusion**: Composition time is not affected significantly by the number of service categories (database access time dominates response time) and composition overhead is reasonable for most settings.

![Composition time vs. number of service categories graph]
Dynamic Service Composition Experiment 2:

- **Task**: Dynamic composition of business process involving 3 service categories, with a single SLA constraint

- **Measurement**: Response time overhead of dynamic service composition for different number of services to choose from for each category

- **Setting**: Central service monitor on Amazon EC2 m3.medium instance (1 vCPU, 3.75 GB memory)

- **Conclusion**: Composition time is not affected significantly by the number of possible services in a category and composition overhead is reasonable for most settings.
Active Bundle Experiments

• Measurements
  – Experiment 1: Growth in AB size with increase in the number of policies
  – Experiment 2: Growth in AB and Service interaction time with increase in the number of policies
  – Experiment 3: Tamper Resistance overhead in AB execution

• Variations
  – AB versions
    • ABx – XACML-based policies and WSO2 Balana-based policy evaluation
    • ABxt – ABx with tamper resistance capabilities
    • ABc – JSON-based policies and JAVA-based policy evaluation
    • ABct – ABc with tamper resistance capabilities
    – Number of AB policies

• Environment
  – Amazon EC2 C3 Large and XLarge instances

• Data collection
  – 5 runs of each experiment
  – 100 requests per run
Experiment 1: AB Size vs. Number of policies

• Observations
  – Linear growth in AB size with increase in number of policies for all versions
  – Tamper resistance adds a slight overhead to AB size (< 2 KB)
  – 79% reduction in policy size (0.79 KB) with JSON-based policies
    • Additional reduction of 8.5 KB with Java-based policy engine
Experiment 2: AB-Service Interaction Time vs. Number of policies

- Observations
  - Linear growth in interaction time with increase in policies for ABx and ABxt
    - Use of XACML-based policies and external library (WSO2 Balana) for policy evaluation
    - Evaluation of XACML policies involve the traversal of XML policy and request trees
  - Constant growth in interaction time with increase in policies for ABc and ABct
    - Use of JSON-based policies and Java code for policy evaluation
    - Highly optimized Java code evaluation
Experiment 3: Tamper Resistance Overhead

• Observations
  - Tamper resistance has higher overhead for XACML policies
    • Digest calculation of XACML policies involves the traversal of XML policy and request trees
    • Digest calculation of JSON policies takes less time due to smaller policy size
Comprehensive security auditing and enforcement architecture for trusted & untrusted cloud

- Continuous monitoring of service SLA and policy compliance
- Efficient mechanism to dynamically reconfigure service composition based on the system context/state (failed, attacked, compromised) and resiliency requirements
- Privacy-preserving data sharing approach for client-to-service and service-to-service interactions
- Increased client control over monitoring of cross-domain data sharing
- Proactive treatment of potentially malicious service invocations and data sharing with services
- Compatibility of solution with industry standard SOA and cloud frameworks
Demonstration

• Code repository: https://github.com/rohitranchal/pd3
• Active Bundles tutorial: https://www.cs.purdue.edu/homes/bb/ngcsoa_webpage/PD3_TechFest_Tutorial_ReleaseOK.pdf

• Demo videos:
  1. https://dl.dropboxusercontent.com/u/79651021/AgileDefenseDemo.mp4: Surveillance mission scenario with unmanned combat air system; dynamic service composition reconfiguration based on changing context and SLA requirements
  4. https://dl.dropboxusercontent.com/u/3723150/AB-Insecure.mp4: AB created in insecure browser and sent to the cloud for execution
Privacy-Preserving Data Dissemination and Adaptable Service Compositions in Trusted and Untrusted Cloud (September 2015 – August 2016)

*: service performance & security parameter values
+: summary service health data, cyber threat information (STIX)
Technical Approach Overview

- Service monitor intercepts all client-service/service-service interactions.

- The approach aims to provide a unified security architecture for trusted and untrusted clouds by integrating components for:
  - Service trust management
  - Interaction authorization between different services
  - Anomaly detection based on service behavior
  - Dynamic service composition
  - Secure data dissemination using active bundles
  - Unified threat information sharing using a common language
• Each service domain has a monitor that tracks interactions among the services in the domain and outside the domain.

• Local service monitors gather performance and security data for each service, logged in the local database of the monitors and mined using unsupervised machine learning algorithms based on multivariate time-series data analysis to detect deviations from normal behavior.

• Local analysis results are reported to a central monitor using a common language (STIX).

• Central monitor uses information from local monitors to update trust values of services and reconfigures service compositions to provide resiliency against attacks and failures.

• Threat intelligence data gathered from multiple domains are analyzed by central monitor and stored as intelligence feeds in the form of active bundles. Services/users can then search over these intelligence feeds to take actions for protecting against attacks and limit the propagation of threats.

• Detection of service failures and/or suboptimal performance triggers restoration of optimal behavior through replication of services and adaptable live migration of services to different platforms.

• Privacy-preserving dissemination of data between services is also achieved using active bundles. Likewise, data services in the cloud utilize active bundles for protected data storage that enforces fine-grain security policies associated with the usage of the data items when authorizing access. Interaction logs of active bundles form part of the data utilized by the central monitor during threat analysis.
Agile Defense, Antifragility, and Adaptability

Goals:

– Provide Resiliency (withstand cyber attacks, sustain and recover critical functions)
– Provide Antifragility (increase in capability, resilience, and robustness as a result of mistakes, faults, attacks and failures)
– Replace anomalous services with reliable versions
– Reconfigure service orchestrations in response to anomalous service behavior
– Swiftly adapt to changes in context:
  • Services may be in trusted or untrusted environments (e.g. public clouds)
  • Choose services in orchestration to comply with SLA requirements (e.g. response time, latency, security level etc.) based on context (e.g. trust may be important in one context, response time in another)
  • Choose data dissemination policy based on context (coarse-grain access in untrusted cloud, fine-grain access in trusted domain)
– Ensure continuous availability assuming system is under constant attacks and failures
Agile Defense and Adaptability

Components:

- Monitor service status and determine action
  - Update service health status in case of significant deviations from normal behavior
  - Create service backup in case of suspicion of anomaly
  - Re-deploy service in case of complete failure

- Dynamic service reconfiguration based on changes in context
  - Adapt priorities (e.g. response time vs. level of detail/accuracy)
  - Adapt constraints (e.g. trust levels of all services > $T$ for critical mission)
  - Dynamically replace failed services in composition with healthy ones to avoid complete restart of business process

- Controlled sharing of data
  - Determine data dissemination based on the requirements and authorization level of the subscriber
  - Limit data disclosure based on trust level of subscriber
  - Control dissemination based on changes in context (e.g. emergency, attack, etc.)
Machine Learning for Anomaly Detection

• Supervised learning:
  – Requires training data that is labeled (i.e. class labels such as normal operation, DoS attack, injection attack etc.) corresponding to each observed input (e.g. CPU usage > X, number of authentication failures > Y etc.).
  – This is the working principle of signature-based anomaly detection techniques, i.e. particular inputs (observation sequences) are mapped to particular classes of attacks known in advance.
  – Main shortcoming is requiring large amount of training data and providing accurate results only previously observed in the system

• Unsupervised learning:
  – The model is not provided with the class labels during training.
  – The training data consists of a set of input vectors without any corresponding target values (class labels).
  – The goal may be to discover groups of similar examples within the data, where it is called clustering.
  – For service behavior, unsupervised learning with security and performance parameters can be used to find clusters such as periods of high/normal/low memory usage etc. These clusters can then form the observation sequences for algorithms such as Hidden Markov Models. Any values that fall outside these clusters will be detected as outliers and help detection of anomalies.
## Machine Learning Techniques for Anomaly Detection

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means Clustering</td>
<td>Low complexity</td>
<td>Sensitive to initial assignment of centroids and noisy data</td>
</tr>
<tr>
<td>Expectation-Maximization (EM) Meta Algorithm</td>
<td>Model can be easily changed to adapt to different distributions</td>
<td>Converges slowly in some cases</td>
</tr>
<tr>
<td>One-Class Support Vector Machine (SVM)</td>
<td>Can handle very high-dimensional data, usually has high accuracy</td>
<td>Requires high memory and CPU time, requires both positive and negative examples</td>
</tr>
<tr>
<td>Unsupervised Neural Network</td>
<td>Has learning capability, does not require reprogramming</td>
<td>Requires high processing time for large networks</td>
</tr>
<tr>
<td>Self-Organizing Map</td>
<td>Capable of high dimensionality reduction</td>
<td>Time consuming</td>
</tr>
<tr>
<td>Hidden Markov Models (HMM)</td>
<td>Representative of the time-based relations and states that services go through during their lifecycle</td>
<td>Have scalability issues</td>
</tr>
</tbody>
</table>
Anomaly Detection Approach

- Statistical analysis of multivariate time-series data collected by service monitors to detect significant deviations from normal behavior
- Adjusts service threat levels based on duration, extent & type of anomalies
- Correlation of time-series data from multiple services allows for detection of bigger threats (affecting the whole domain, collaborative attacks etc.)
- Ability to detect zero-day attacks as opposed to signature-based models

Diagnosis: Anomaly affecting S1
Response: Replace S1

Diagnosis: Anomaly affecting whole domain
Response: Re-deploy all services in different domain / switch to backup versions in different domain
Anomaly Detection Approach (cont.)
## Performance and Security Parameters for Anomaly Detection

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Cloud service related</th>
<th>Data service related</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of requests/sec</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Bytes downloaded/sec</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Bytes uploaded/sec</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Total error rate</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CPU utilization</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Memory utilization</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Number of authentication failures</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Number of connections</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Number of connection failures</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Number of disk reads/writes</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Network latency</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Service response time</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Disk space usage</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Throughput</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Number of database connections</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Service/cluster health status</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Unsupervised Learning Approach for Service Anomaly Detection

• Training:
  
  Input: Matrix $V_{d \times t}$ of service performance record
  
  $d$: number of performance parameters
  
  $t$: number of time points observed

  Cluster each set of performance parameter values using K-means algorithm

• Testing (system operation):

  for each service interaction log
  
  measure distance of performance parameter values to each cluster, assign time point to closest cluster

  if latest interaction does not belong to any cluster
  
  raise anomaly signal
TotalADS for Software Anomaly Detection

We will investigate possible usage of and integration with the TotalADS [1] software anomaly detection system:

- Developed by Prof. Wahab Hamou-Lhadj et al. at Concordia University
- Framework to automatically diagnose anomalies from software systems in operation
- Supports execution traces and logs in CTF, XML and text format
- Supports live anomaly detection using trace streaming along with real-time training and testing
- Supports variety of trace inspection views for forensic analysis such as control flow of processes, resource usages etc.
- Developed in Java and provides extensible interfaces in Java to add new algorithms, trace formats and views
- Winner of “People’s Choice Showcase Award” at the IBM Center of Advanced Studies Conference
Benefits:

- Increased availability of services (measured by up-time and throughput)
- Increased performance by obviating the need to restart invocations of service compositions with failed/attacked services (measured by total response time)
- Increased security and flexibility in service composition based on priorities and constraints in a specific environment (measured by success of avoiding attacks)

Costs:

- Dynamic service composition time cost
- Cost to maintain central service monitor
- Service response delay due to monitoring
- Increased resource usage in service domain
- Overhead of re-deploying service in same/different domain
Privacy-Preserving Publish/Subscribe Cloud Data Storage and Dissemination

Architecture of an Application for an NGC customer (by Dr. Leon Li, NGC)
• Public cloud for storing and sharing intelligence data from various sources
• Use of public cloud requires data stored to be encrypted
• Components:
  – Collection agent to gather intelligence feeds
  – CryptDB storing encrypted data and providing SQL query capability over encrypted data
  – Subscription API to provide methods for authorized access to data
• Incoming data may have different access and usage policies
• In order to ensure secure access to and usage of data, the policies specified at the source must be evaluated and enforced at all times including storage, access and usage
• Active bundles provide the requires policy-based controlled data disclosure capability
Active Bundle

- Active Bundle (AB) approach for secure data dissemination
  - Self-protecting data encapsulation mechanism
  - Provides secure cross-domain information exchange
- Sensitive data
  - Encrypted data items
- Metadata
  - Access control and operational policies
- Virtual Machine
  - Protection mechanism (self-integrity check)
  - Policy evaluation, enforcement and data dissemination
Active Bundle Solution Components

• Service authentication can be based on
  – Password, Certificate, Biometric, PKI

• Service request authorization is based on policy evaluation
  – Flexible policy specification based on access control models such as Attribute/Role-based

• Key management for data disclosure
  – Key inclusion (prone to attacks)
  – Centralized key management service (use of trusted third party for key storage and distribution)
  – Distributed key management that splits the keys into shares using threshold secret sharing and uses a Distributed Hash Table (DHT) to store the shares and reconstruct the key by retrieving minimum threshold number of shares (unsuitable for real-time interactions in a service environment)
  – Dynamic key derivation based on the unique information generated in AB execution control flow steps only if the service is authenticated and authorized

• Tamper Resistance
  – Correct data dissemination depends on the correct execution of AB control flow steps
  – Verify the integrity of the execution steps to ensure there is no difference from the original code (using secure one-way hash function)
  – Derive keys based on digests of AB execution steps and their resources
  – Any modification of AB changes the digest resulting in incorrect key derivation
CryptDB integration

- CryptDB stores keywords (in encrypted form) for locating ABs.
- CryptDB is a proxy to a database server
  - It encrypts query
  - It never releases decryption key to database
- When database is under attack, only ciphertext is revealed
  - Database has anonymized-schema with encrypted table and column names
  - Database has encrypted data
- Even when CryptDB is compromised, leakage of data is restricted to the data for currently logged in users.
- Status: setting up with node.js and MySQL is complete.
Isolated AB Execution

- Recipient may be reluctant to execute AB => we support the isolated execution of AB by means of Docker.
  - Docker is based on Linux container which is light-weight virtual machine
  - When AB arrives at recipient machine, one virtual machine is created and AB is copied into that virtual machine.
  - AB can be executed inside virtual machine. Only the result returns to host machine
Key Generation during AB Creation

• An AB Template is used to generate new ABs with data and policies specified by a user
  – An AB Template includes the implementation of the invariant parts (monitor) and placeholders for customized parts (data and policies)
• User specified data and policies are included in the AB Template
• AB Template is executed to simulate the interaction process between an AB and a service requesting access to each data item of AB
• The information generated during the execution of different AB modules and the digests of these modules and their resources (such as authentication (authentication code, CA certificate that it uses), authorization (authorization code, applicable policies, policy evaluation code)) are collected and aggregated into a single value for each data item
• The value for each data item is input into a Key Derivation module (such as SecretKeyFactory, PBEKeySpec, SecretKeySpec provided by javax.crypto library)
• The Key Derivation module outputs the specific key relevant to the data item
• This key is used to encrypt the related data item
Key Derivation during AB Execution

- AB receives access request to a data item from a service
- AB authenticates the service and authorizes its request
- The information generated during the execution of different AB modules and the digests of these modules and their resources (such as authentication (authentication code, CA certificate that it uses), authorization (authorization code, applicable policies, policy evaluation code)) are collected and aggregated into a single value for each data item
- The value for each data item is input into the Key Derivation module (such as SecretKeyFactory, PBEKeySpec, SecretKeySpec provided by javax.crypto library)
- The Key Derivation module outputs the specific key relevant to the data item
- This key is used decrypt the requested data item
- If any module fails (i.e. service is not authentic or the request is not authorized) or is tampered, the derived is incorrect and the data is not decrypted
Context-Sensitive Data Disclosure

• Perfect data dissemination not always desirable
  – Example: Confidential business data shared within an office but not outside

• Context-sensitive AB evaporation
  – AB evaporates in proportion to their “distance” from their owner

• “Closer” subscribers trusted more than “distant” ones

• Illegitimate disclosures more probable at less trusted “distant” guardians

• Different distance metrics
  – Context-dependent
Examples of Distance Metrics for Trust

- Examples of one-dimensional distance metrics
  - Distance ~ business type

- Multi-dimensional distance metrics
  - Security/reliability as one of dimensions

If a bank is the original guardian, then:
  -- any other bank is “closer” than any insurance company
  -- any insurance company is “closer” than any used car dealer
Example of Controlled Data Distortion

• Distorted data reveal less, protects privacy
• Examples:

<table>
<thead>
<tr>
<th>accurate data</th>
<th>more and more distorted data</th>
</tr>
</thead>
<tbody>
<tr>
<td>250 N. Salisbury Street West Lafayette, IN</td>
<td>Salisbury Street West Lafayette, IN</td>
</tr>
<tr>
<td>250 N. Salisbury Street West Lafayette, IN [home address]</td>
<td>250 N. University Street West Lafayette, IN [office address]</td>
</tr>
</tbody>
</table>
P2P Secure Data Sharing in UAV Network

Dissemination Policy:

- If $2.5 > \text{trust} \geq 2.2$ => blur level = 20
- If $2.2 > \text{trust} \geq 2.0$ => blur level = 40
- If $2.0 > \text{trust}$ => blur level = 80

trust = 2.4
trust = 2.1
trust = 1.5
P2P Secure Data Sharing in UAV Network

Dissemination Policy:

- If context = emergency => contrast = 0.4
- If 2.5 > trust ≥ 2.2 => contrast = 0.2
- If 1.8 > trust => contrast = 0.1

trust = 2.0
context = emergency

trust = 2.4

trust = 1.7
Success Criteria

- Resilience against service attacks
- Service uptime
- Resilience against data privacy violations
- Resilience against data tampering attacks
- Runtime performance of distributed service monitoring
- Successful detection of service anomalies
- Active bundle search scalability in cloud
- Successful deployment of monitoring framework on industry-standard cloud infrastructures
# Milestones & Deliverables

## Milestones:

<table>
<thead>
<tr>
<th>Task</th>
<th>Q1 (Sept-Nov)</th>
<th>Q2 (Dec-Feb)</th>
<th>Q3 (Mar-May)</th>
<th>Q4 (Jun-Aug)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration of active bundles with NGCRC projects</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed monitoring system setup</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extension of AB with stateful execution, incremental dissemination, situation awareness</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Development of anomaly detection models</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Experiments and Demo</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Milestones & Deliverables

Deliverables:

Software:
• Active Bundle Generator
• Active Bundle Handler
• Active Bundle Monitor
• Distributed Service Monitor
• Anomaly Detection Module

Documentation:
• Source code
• Deployment and user manuals
• Reports characterizing performance of proposed solution
• Publications on comprehensive results of proposed privacy-preserving data dissemination and adaptable service composition approach
Collaboration & NGC Project Integration

Service request → Browser

- Insecure browser
  - Analyze request source based on W3C standards
  - Send active bundle to cloud for execution
  - Active Bundle
    - Data access request
    - Encrypted search
      - Filtered/lower quality data
      - Filtered search results

- Secure browser
  - Authentication
    - CAC
      - Trust = X + Y
      - High trust
    - PIN
      - Trust = X
      - Low trust

Service Domain
- Service X
  - Data request
  - Active Bundle

UNTRUSTED → TRUSTED
Secure End-to-End Information Flow in Cloud

1. Client sends request to the service using browser and shares data by means of Active Bundle (AB)

2. Service checks the request source (secure or insecure browser)
   - Based on W3C Crypto standards

3. Service executes AB in Cloud if created by an insecure browser

4. Service interacts with AB and requests data

5. AB behaves differently under different contexts
   - Full data dissemination based on service authorization/trust level
   - Context-based partial data dissemination based on insufficient authorization level
   - No data dissemination for unauthorized access/attacks

6. Cross-domain information exchange with trustworthy/untrustworthy subscribers
   - Data dissemination is done on a “need to know” basis by limiting the disclosure of decryption keys
   - Incremental disclosure of keys based on increase in the “need”
• Cyber 2.0 (with Dr. Sunil Lingayat):
  – The service monitor interfaces together with administrator provided policies will allow for automated responses to threats and dynamic routing, as well as notification, of threats across the system as a whole.

• CURATE 2 (with Jason Kobes)

• Citadel EBS (with Jonathan E. Fulkerson)
  – Missile defense

• Cyber Exportable Risk Management Techniques (CERMT)
  – Will provide benefits in terms of risk management models under various contexts
CURATE 2 Objectives & Values

- Real time access to continue to do mission
- Computing on encrypted data
- Limiting data leakage/disclosure
  - Active bundles keep data private and uncompromised
  - Only legitimate users get data to the extent they are authorized
  - If subscriber misuses data, owner is notified and misuse will be limited
  - Data can be made available on multiple devices, but it will be based on authentication (security clearance) of devices and users
- Advanced integration with honeypots
  - Give false data to malicious users
  - Benign users get real data with an assurance mechanism they understand
  - If unsure about users, give partial view with opportunity to earn more trust
- Business value:
  - Capable of being implemented in many environments
  - Allowing access to information that was not available
  - Lower cyber insurance premiums
  - Stakeholders may find intrinsic value in the technology
References


5. Protecting PLM Data throughout their Lifecycle, R. Ranchal, B. Bhargava, 9th QSHINE.


THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

CYBERSECURITY RESEARCH Consortium