Institute of Advanced Engineering and Science

International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.2, No.1, February 2012, pp. 28~39
ISSN: 2089-3337

Content-centric Information Protection in Cloud Computing

Christopher C. Lamb*, Gregory L. Heileman*
* Department of Electrical and Computer Engineering, University of New Mexico

Article Info

Article history:

Received Oct 14th, 2012
Revised Oct 25th, 2012
Accepted Nov 06th, 2012

Keyword:

Security
Cloud Computing
Usage Management
Information-centric Networks

ABSTRACT

Information security has become increasingly important as organizations migrate systems to third-party infrastructure providers. Once migrated, however, previously transparent network topologies, information paths, and systems infrastructure became more opaque. This loss of control when coupled with storage of corporate and personally sensitive information lead to significant increases in potential vulnerability. In this paper, we present experimental evidence demonstrating the feasibility of using content-centric networks with integrated policy-based usage management. We describe a nation-spanning content network developed within Amazon and Rackspace infrastructures and collect performance statistics to show the suitability of various confidentiality strategies in these kinds of large heterogeneous systems. In doing this, we first consider the current state of the art in network information security as well as some of the shortcomings of current designs, and propose a taxonomy of network-enabled usage-control architectures that can solve sensitive information transmission problems. We then close with a description of our content-centric network, a discussion of our experience using this system to manage real-time sensitive information flow over commercial cloud systems, and experimental evidence demonstrating the feasibility of the approach.

Corresponding Author:

Christopher C. Lamb,
Department of Electrical and Computer Engineering,
University of New Mexico,
MSC01 1100, 1 University of New Mexico, ECE Bldg., rm 125, Albuquerque, NM 87131-0001.
Email: cclamb@ece.unm.edu

1. INTRODUCTION

Current enterprise computing systems are facing a troubling future. As things stand today, they are too expensive, unreliable, and information dissemination procedures are too slow. Current approaches to partitioning information are unable to migrate to cloud environments. Additionally, the current approach of controlling information by controlling the underlying physical network is not cost effective and is therefore unsustainable. These problems leave large governmental and commercial organizations that must protect highly sensitive data in a very vulnerable position, one in which they cannot continue doing what they have done, but cannot migrate to what everyone else is doing in order to gain efficiencies (1). In many cases networks containing sensitive data are separated from other internal networks to enhance data security at the expense of productivity, leading to decreased working efficiencies and increased costs (2). Information delivery without regard for underlying infrastructure exposes that information to unnecessary risk as breaking encryption becomes easier and easier. Content-centric routing with a variety of delivery options is a flexible solution to these problems.

Federal, military, and healthcare computer systems are prime examples of these types of problematic distributed systems, and they demonstrate the difficulty inherent in implementing new technical solutions. These types of systems need to be re-imagined in order to take advantage of radical market shifts in computational provisioning. New approaches to networking and information management present possible solutions to these kinds of problems by providing distributed information-centric approaches to data management and transfer (3) (4). Cloud systems certainly provide strong economic incentives for use, leading to cost savings and increased flexibility, but they also have distinct disadvantages as well that must be addressed before highly secure environments can realize these benefits (5).

How to address these issues is an open research question. Organizations ranging from cloud service providers to the military are exploring how to engineer solutions to these problems, and to more clearly understand the trade-offs required between selected system architectures (6). Within this paper, after reviewing the current state of the art in secure systems, we describe specifically how information can be better protected when transiting dynamic networks while still providing timely access to needed information. We present a specific taxonomy of development that demonstrates how to migrate from current to future systems, and describe our experience with our own information-centric overlay prototype. The specific contributions of this work include our taxonomy, our approach to applying information-centric security in dynamic networks, our experimental results supporting our approach, and the application of our ideas to not only current cloud-based systems but to information and content-centric networks as well (7) (8) (9) (10).

1.1. Current Solutions

The Unified Cross Domain Management Office (UCDMO) supports efforts to develop solutions to enable unfettered but secure information flow. The National Security Agency set the standard in this area initially. In 2009, at a conference sponsored by the UCDMO, Booz Allen Hamilton (BAH) and Raytheon presented alternative notional architectures contrasting with current NSA-influenced approaches (11) (12) (13) (14). These cross-domain solutions are intended to enable sensitive information to flow both from a higher sensitivity domain to a lower sensitivity domain, and from lower to higher as well. They generally act over both primary data (say, a document) and metadata over that primary data.

The NSA conducted initial work in this area. Their standard-setting efforts culminated in a reasonable conceptual system architecture, using groups of filters dedicated to specific delineated tasks to process sensitive information (12).

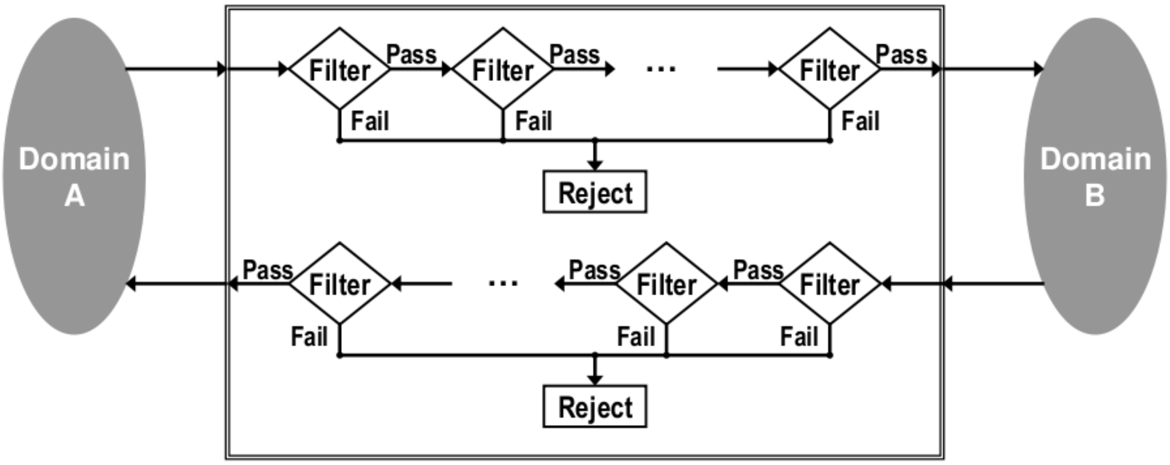

Figure 1. Original Cross-Domain Notional Architecture

In the scenario portrayed in Figure 1, Domain A could very well be a private cloud managed by the U.S. Air Force, while Domain B might be a public operational network of some kind shared by coalition partners in a joint operation. A system user attempts to send a data package consisting of a primary document and associated metadata from Domain A to Domain B. At some point, that submission reaches a guard, which contains at least one filter chain. Each filter chain then contains at least one filter. Individual filters can execute arbitrary actions over a submitted data package and have access to any number of external resources as required. At any point, a filter can examine the data package and reject it, at which point it will frequently wait for human review. If a filter does not reject a data package, it passes that package onto the next filter or submits it for delivery to Domain B.

In recent years, the NSA has extended the legacy system architecture for cross-domain information sharing to exploit service-oriented computing styles (12). Visualized in Figure 2, this model incorporates more modern conceptual elements and component architectures.

We see on the left the Global Information Grid, or GIG. On the right, we have the Distributed Service-oriented Cross Domain Solution, or DSCDS. The GIG is not a truly open system – rather, it is a loosely coupled collection of computational services handing data at a variety of levels of sensitivity, federated to provide stakeholders timely access to relevant information (11). The DSCDS is essentially the embodiment of the NSA's cross-domain vision applied to service oriented computing. This model fuses various technology choices with previous cross-domain thinking.

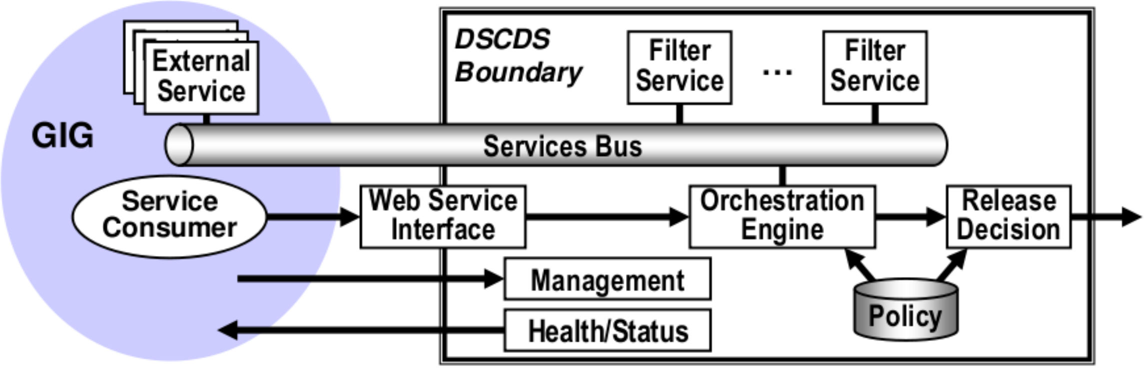

Figure 2. Service-centric Cross-Domain Notional Architecture

Indicative of this more modern system design, we have a variety of services and service consumers attached to a common service bus within the GIG. Within the DSCDS, we have groups of filters implemented as services inspecting transferred data when moved over the bus. Finally, all of this interaction is managed by a management interface and controlled by an orchestration engine accessing a centralized group of policies. Note that here we have begun to access a common policy repository for various types of security metadata regarding primary data elements.

In the past few years, Raytheon has offered a new model for cross-domain use influenced by the NSA service-oriented model (14). The model in Figure 3 is more grounded in the actual technical environment this kind of solution would be embedded within. In this figure the Non-secure Internet Protocol Router Network (NIPRNet) is one domain, and the Secret Internet Protocol Router Network (SIPRNet) is the other. NIPRNet is the lower security domain (lowside), and SIPRNet the higher security domain (highside). This particular view shows the motion of data from the high side to the low side. Here, a data request is submitted from SIPRNet first to the XML Security Gateway which calls into the Orchestration Engine for policy validation. The Orchestration Engine then coordinates calls into a Policy Repository as well as to a collection of external Support Services. Once rectified against these elements, the request is passed into the Cross Domain Guard that routes the request into the Unclassified Enclave in NIPRNet. Here, the request is passed directly through the lowside XML Security Gateway, without rectification, onto the Service Provider. The response from the Service Provider is then passed back to the requester via the inverse path. This model begins to use a centralized policy repository, just as the NSA Service Model. It also uses a single cross domain guard to transfer information from both the highside to the lowside, and vice-versa.

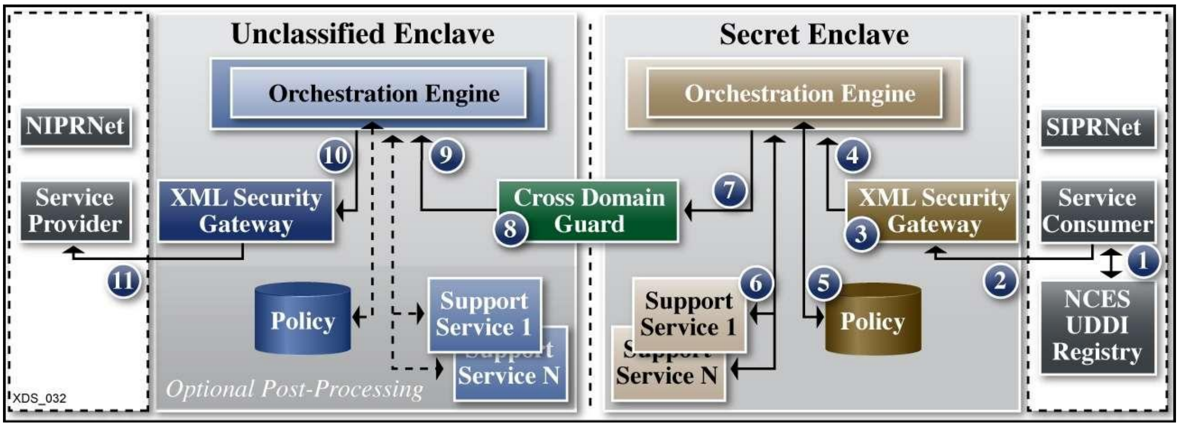

Figure 3. Raytheon’s Notional Architecture

BAH submitted a competing model, also in 2009 (13). In fact, both Raytheon and BAH presented their models under competitive contract to the UCDMO at the same conference, so the domain application is not coincidental.

Figure 4 embodies BAH's thinking with respect to cross-domain information management. In this case Domain A is a high security domain, and Domain B is a low security domain. Data flows from the highside to the lowside through the cross-domain management system. While not as detailed as the Raytheon proposal, this approach does have similar elements. For instance, the data first travels from Domain A into the Interface Segment for Domain A, similar to the secret enclave used in the Raytheon model. From there, it moves into the CI Segment, which in turn submits the transferring data into the Filter Segment. From there, the package is moved into the Interface Segment for Domain B, and then onto Domain B. The Administrative Segment provides management and oversight of the system as a whole. Note the absence of specific policy-centric elements. This system is reliant on specific policy-agnostic content filters.

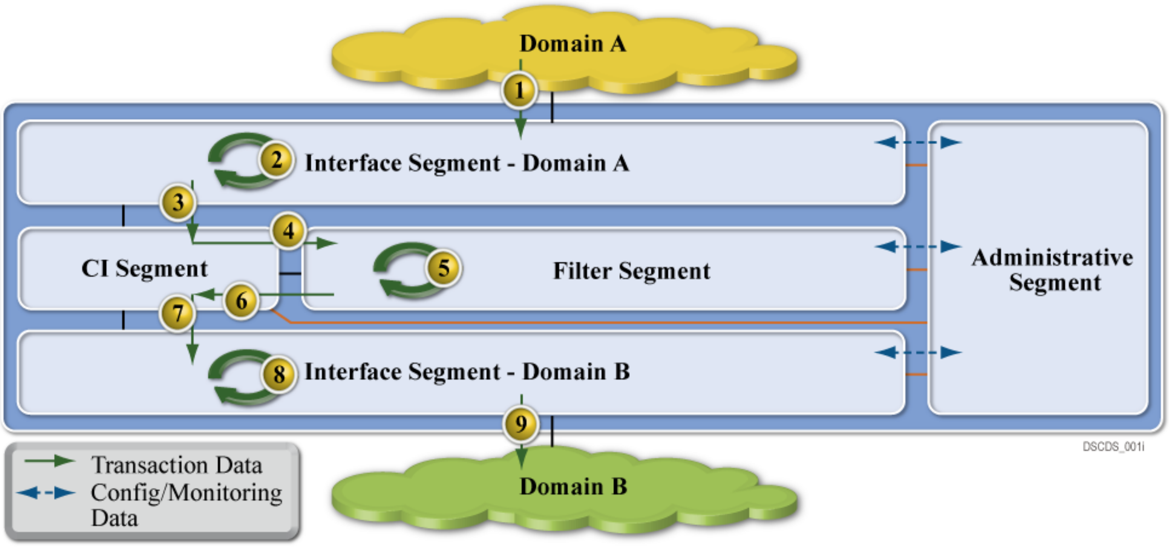

Figure 4. Booz | Allen | Hamilton’s Notional Architecture

The two cross-domain solutions described previously have clear similarities, and work in this area has not progressed far beyond the initial notions of how these kinds of systems should work. Most solutions, for example, still use some kind of filter chaining mechanism to evaluate whether a given data item can be moved from a classified to an unclassified network. Both NSA models used filters explicitly, as did the BAH model. They all use a single guard as well, a sole point of security and enforcement, providing perimeter data security, but nothing else from a system-theoretic perspective. The physical instantiations of these models are locked by operational policy to a single classification level. Users cannot, for example, have Top Secret material on a network accredited for Secret material.

Future desired systems will provide decentralized policy management capabilities, infrastructural reuse, the ability to integrate with cloud systems, and security in depth. Policy management will need to be decentralized and integrated within the fabric of the system. The system will be more secure and resilient as a result, better able to control information and operate under stressful conditions. Multi-tenancy can lower costs and increase reliability, and an appropriately secured system facilitates integration of computing resources into multi-tenant environments. The ability to handle multi-tenant environments and to reliably secure both data at rest and data in motion leads to computational environments deployable in cloud systems. Finally, systems must operate under a wide variety of conditions, including when they are under attack or subject to compromise (15).

This work introduces the notion of usage management embedded in the delivery network itself. It also provides an analysis of the challenges and principles involved in the design of an open, inter-operable usage management framework that operates over this kind of environment. Besides referencing the material we have covered to portray the current state of the art, the analysis includes application of well-known principles of system design and standards (16) (17) (18), research developments in the areas of usage control (19) (20), policy languages design principles (21), digital rights management (DRM) systems (22), and interoperability (23) (24) (25) (26) (27) towards the development of supporting frameworks.

2. PROPOSED SOLUTIONS AND TAXONOMY

A clear taxonomic organization of potential steps in approaching finer-grained policy-based usage management helps in describing the difficulties inherent in developing potential solutions and also aids in planning system evolution over time. In Table 1 we describe four distinct types of integrated policy-centric usage management systems. Of these four, only the first two levels are represented in current system models.

Table 1. Taxonomic Elements

Name	Description
ϕ	The initial level of this taxonomy, these systems have a single guard without policy-based control
α	These systems have begun to integrate policy-based control
β	Systems that have begun to integrate policy-based control with router elements
γ	Systems that have integrated policy-based control with routing and computational nodes

In this taxonomy, it is not required that systems pass through lower levels to reach higher ones. This taxonomy represents a continuum of integration of usage management controls. Systems can very well be designed to fit into higher taxonomic categories without addressing lower categories. That said, it should be noted that many of the supporting infrastructural services, such as identification management or logging and tracing systems, are common between multiple levels. The taxonomy itself starts with the current state, integrating policy evaluation systems into the network fabric gradually, moving away from filters, then by adding policy evaluation into all network nodes.

The ϕ classification in Table 1 consists of systems like the initial NSA and BAH notional models. These systems consist of two distinct domains, separated by a filter-centric single guard. The initial NSA system model is clearly of this type, separating two domains with a guard using filter chains. The BAH model is also of this type, using filter segments to evaluate data packages transmitted between interface segments attached to specific domains.

In these kinds of systems, specific rules regarding information transfer and domain characterization are tightly bound to individual filter implementations. They are based on *a priori* knowledge of the domains the guard connects, and therefore are tightly coupled those domains. Furthermore, the filter elements are standalone within the system, in this classification, not availing themselves of external resources. Rather, they examine information transiting through the filter based purely on the content of that information. The set of filters that could be developed and deployed within the guard are unlimited. Developers could easily create a filter that inspects and possibly redacts the sections within the document, rather than passing or not passing the entire document through the guard. Indeed, if we assume general processing capabilities within the guard, that is, Turing completeness, then this guard can be made as powerful as any solution we can derive for implementing a cross-domain solution (CDS). Thus the computational power of the guard is not the issue. The real issues are the benefits that can be gained by distributing the capabilities intelligently within the networked environment.

Next, the α overlay classification contains systems that have begun to integrate policy-centric usage management. Both policies and contexts are dynamically delivered to the system. The dynamic delivery of context and policies allows these types of systems more flexibility with policy evaluation. The α category begins to integrate policy-centric management rather than using strict content filtering. In this case, we again have at least two domains, Domain A and Domain B, though we could potentially have more. The ϕ type systems require domain specific information to be tightly coupled to the filter implementations. Separating the permissions, obligations, and other constraints from the filters and incorporating them into a specific separate policy entity frees the guard from this coupling and provides additional flexibility to the system. The guard can continue to use filters to process data. These filters however are now more generic and decoupled from the specific domains the guard manages. The choice of using a specific filtering model rather than some other kind of construct is a design detail level to implementers. That said, individual filters can be remarkably different but still need to understand the ontologies over which specific licenses are defined rather than specific content semantics. The policy repository is key to the implementation and differentiation of this taxonomic category. This repository can be implemented as a separate repository keyed into using a data artifact's unique URI, for example. It could also represent a policy sent in tandem with a data artifact in a data package. The policy repository may be implemented as an external service, and as such, represents the first such external service explicitly used in this taxonomy. Other external services may well exist in this type of architecture and be used to adjudicate information transfer decisions as well.

The β taxonomic category begins to integrate policy-centric processing with router elements in the network. While this work is centered on using overlay technology to illustrate and implement these concepts, it is important to note that this kind of distributed policy-centric processing could very well be distributed into the physical routing fabric of a given network by extending Software Defined Networking systems such as OpenFlow (28). In this model we can also host multiple domains as a result of flexible policy-based content examination. Each domain hosts a network of some kind, though that hosted network could very well be a degenerate network of a single system. Each network hosted in a domain is hierarchical, with specific computational nodes embodied by workstations, tablet computers or mobile devices, and routing points embodied by routers or switches of some kind.

We have started to penetrate into the routing fabric of the network by doing content evaluation at router points. Content-based switching networks have been successful in other domains, and such techniques can be used here to provide policy evaluation capabilities (29). It should be noted that certain types of traffic are easier to evaluate than others; for example, HTTP requests and responses are easier to examine than TCP packets. When examining TCP packets, systems generally require additional context to select an appropriate packet window (e.g. the number of packets cached for examination). HTTP traffic does not usually require this level of complexity.

This migration of policy evaluation into the routing fabric provides for enhanced data security and better network management, especially if part of a network is compromised. Now that policy decisions can be made at the router level in a given network, we are starting to have network security in depth rather than simple perimeter protection. This not only provides the ability for additional information protection, but also allows for different compartments holding information at different need-to-know levels to be created ad-hoc under different routing segments. In cases of network compromise, this type of dynamic policy enforcement can also allow for quick node excision as well.

Finally, the γ compartment has integrated policy evaluation with compute and routing nodes. In this case, policies can be evaluated against content at all network levels --- nodes emitting requests, nodes fielding requests, and all routing elements in between. The policy repository in this architecture is supplying services to all computational elements in both domains. This provides increased granularity with respect to data compartmentalization by integrating information security into each network element. At this point, the network can create compartments of single nodes, while previously in β level systems compartments could only be created under specific routing elements. At this level, of the taxonomy we can also provide services revoking data access based on policy evaluation decisions. Individual node exclusion is possible as well. β classified systems could excise network elements under specific routers by dynamic policy application. In this case we can apply the same functionality to individual compute nodes. For example, if a networked device like a smart phone is compromised, that device can be quickly removed from access or used to supply misinformation.

The various levels of the taxonomy vary primarily with respect to the inclusion of policy-based usage management and information-centric structure. The ϕ type systems are not structured with distributed use in mind, nor do they use policy-centric management. Conversely, γ type systems are both purely policy oriented and completely distributed. As systems move through the various levels of the taxonomy they gradually move from one side of the spectrum to another. Distributed usage management structures, hierarchical or otherwise, gradually migrate into the network beginning with β systems. Policy orientation is injected into the architectures starting with α systems and moves into the network fabric in parallel with information-centricity.

3. RESULTS AND EXPERIENCE

Our research so far has focused on the development of a proof-of-concept system that allows us to simulate each of the policy-centric taxonomic categories and provides the capability for obtaining performance and reliability measures over transmitted content. Our current system can emulate α, β, and γ architectures and various confidentiality strategies implemented in tandem with policy-centric usage management. We can extract performance measures based on the rate of information flow as needed.

As part of our research effort, we have created and deployed baseline system images in both Amazon's Elastic Compute Cloud (EC2) and Rackspace Servers infrastructures. We have also created and exercised deployment, configuration, and logging systems to enable distributed monitoring and centralized reporting. Overall, we currently have 22 nodes, including two test nodes, running with two distinct providers geographically dispersed across the continental United States. This leads to a distinct requirement for a centralized system with distributed access for initial configuration information as well as logging and auditing. We have implemented this infrastructure using Amazon's Simple Storage Service (S3), accessible from both Rackspace and Amazon hosted virtual machines.

The specific technical components are EC2, S3, Rackspace Servers, and GitHub. Both EC2 and Rackspace nodes are Ubuntu virtual machines, albeit at different versions, as we run Ubuntu version 11.04 in Rackspace and Ubuntu Version 12.04 in Amazon's infrastructures. These systems are provisioned with Git, Ruby, the Ruby Version Manager (RVM), and supporting libraries. They all run as micro-instances or equivalent, and are bootstrapped with the appropriate project information to begin to participate as an information network node. While EC2 and Rackspace Server infrastructures are infrastructure-as-a-cloud (IaaS) offerings supporting virtual machine instances of various types, Amazon S3 is a simple key-value store. Running with REST semantics over HTTP, S3 stores arbitrary documents associated with specific keys in buckets. In this way, we can store the global configuration of a specific overlay network in a single location from which every node can access information with respect to their pending role and needed configuration information. Likewise, all overlay network state can also be saved to centralized buckets for later analysis. Finally, Github is a centralized source code repository used to share code between all participating nodes. Prior to each content network instantiation, each node checks the repository for updates, and downloads them if they exist. All data saved within S3 is serialized in a text-based data serialization language known as YAML, a widely supported hierarchical data representation. We use Capistrano to manage and initialize overlay nodes, which allows us to bootstrap different configurations of networks from a single command-and-control node simply and efficiently.

The unique strength of this system is enabling dynamic distributed content control, enabling information redaction, protection, and secure routing. Information retraction involves quickly removing a user's access to sensitive data. Redaction addresses simple data removal, while protection would operationally involve applying encryption. Finally, secure routing would provide the ability to send data over a more secure link if such a link is available and required.

In this system information retraction involves changing the execution context such that access for a given user, perhaps even on a specific device, is removed. This context then propagates through the information network and attached clients. This is useful when a given user, say a coalition partner, is suddenly considered compromised and can no longer be allowed access to sensitive information. Likewise, a specific user's system may likewise be compromised and be forbidden access to specific information.

Information redaction is used when a user does not have authorization for a specific section of content, generally within a larger document. In these cases, that information and related policy metadata are simply removed from any query responses. Likewise, information protection also addresses specific subsections of information in a larger document, but unlike redaction, a user is in these cases authorized to access information, but one of the links over which the information must travel is not authorized to transmit specific sensitive information. In these cases that information can be encrypted with appropriately strong encryption to allow for more secure information transmission.

Finally, secure routing use directly addresses the ability to select communication links based on information content. In these situations, a network has more than one path over which to return content. Furthermore, these multiple paths have different characteristics providing different levels of service. The system, based on rules contained in a policy and the current context can then select communication links of different security levels when returning content.

We use attribute-based control in these scenarios, in which we make access decisions based on the attributes of a requesting user or link rather than defined roles or groups. User attributes support defined policy elements. Not every policy attribute has a corresponding user attribute, as not all policy attributes are associated with users. Some are associated with the user's environment, like operating system or device.

In the scope of this project, we use a Ruby-based domain specific language (DSL) to describe policies. In larger heterogeneous deployments, a standards-based alternative like XACML would be more suitable. This project however is not focused on developing a complete policy specification language, but rather on using one in a very dynamic environment. XACML, for example, is a very large and complete standard that would require a significant investment of effort to implement. It can also tend to be verbose. A simple DSL focused on our specific needs is a more efficient.

3.1 Experimental Results

Experiments using our inter-cloud systems yield promising support for this approach. Our experiments show only a slight degradation of information availability as a result of our network permeated security approach, with redaction and encryption demonstrating the smallest degradation. Rerouting-based approaches have the most performance degradation as a result of secondary infrastructure initialization and use. Redaction and rerouting have the largest negative effects on integrity

In these tests, we used a simulated γ-categorized system. This is the kind of system that organizations like the UCDMO have identified as the final goal state of their work, systems that incorporate policy-centric management in the fabric of systems and networks (12). The kind of components required to do this kind of policy-based content-sensitive evaluation do not currently exist, and components of these kinds of systems are only now beginning to emerge. Systems like OpenFlow, when they have stronger hardware support, can begin to provide some of these capabilities. OpenFlow enabled systems are not yet common or widely used however, and though they do provide the needed control for these kinds of systems, the do not supply the necessary policy interpretation and evaluation. As a result, this experimental work was conducted over an HTTP overlay network, at the application layer. Using a document-focused protocol makes content evaluation simpler as well, as systems can evaluate all content when it transits a network rather than maintaining a buffer of content required when processing packet-level communications.

In order to develop a stronger perspective on network performance, we measure delivery times from three separate nodes. One node is hosted in Comcast's infrastructure (a large local Internet Service Provider), one at Amazon, and another at Rackspace. The tested network has four levels. The first level has a single router node. The next level has two routers, both connected to the router in the first level. The third level contains four routers, two attached to each of the routers at the level just above. Finally, the fourth level contains nodes, distributed so that two level three routers have three nodes, one level three router has two nodes, and the last level three router has four nodes. The first three levels are essentially a binary tree. We query the network from five different locations. We query the node that contains the content requested directly (the home node). We then query a node under the same router as the home node (the peer node). Next, we query a node under a different router, but connected to the same second level router (the neighbor node). Finally, we query two nodes on the other side of the network (the distant (1) and (2) nodes). We query each node 50 times in each simulation, for a total of 250 queries per simulation.

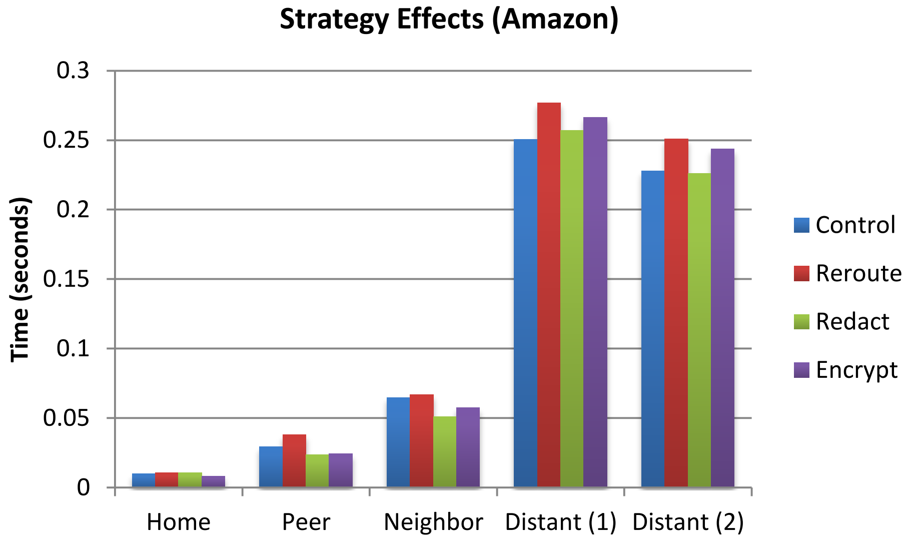

Figure 5. Amazon Timing Results

The goal of this experimental work was to characterize the issues around specific confidentiality strategies in these kinds of networks. The specific strategies addressed were redaction, rerouting, and protection (via encryption), and we evaluated these strategies from the perspective of confidentiality, integrity, and availability. We measured confidentiality via the control used to protect information. Removing information entirely provided the highest measure of protection but is akin to unplugging a computer to improve its cyber-security posture. Routing information through a more secure channel is the next approach, followed by sensitive information protection via strong encryption. We use 256-bit AES-CBC encryption scheme in our current work. We measured availability by the delivery of information and the time required to ensure information delivery, measured by end-to-end network performance. Integrity is a function of the alterations to the information required for secure delivery in the tested scenario. Unaltered information has the highest integrity, followed by information that is still complete but protected via encryption, information that has been divided and rerouted, and finally information that has had content redacted. Though we can specify combinations of strategies in a given network, as we specify strategies by network node, in our experiments we use a single strategy in each network to more clearly attribute strategy performance impacts. We used identical policies in each simulation to ensure the same amount of required usage management actions, limiting the effects on availability to the approach rather than differing policy.

We also ran a control simulation that did not incorporate any usage management to provide a performance baseline.

Figure 5 shows our performance results from our Amazon testing node. The access times for the content from the home, peer, and neighbor nodes were by far the smallest. As the testing node was hosted in the same datacenter as these three nodes, that was to be expected. The access times for both distant nodes was, however, surprisingly high. With that in mind, the overall trend for response times is sensible however, with access time increasing as the requesting node is farther away from the content in the information network. Queries from distant nodes need to traverse five information routers, while home, peer, and neighbor nodes only traverse one, two and three, respectively. Also surprising was the finding that rerouting was generally more expensive from an availability perspective than encryption-based approaches. This is likely attributable to the costs associated with attaching to the external SMTP server, hosted at Google, used as the out-of-band communications channel. Also evident is remarkable performance variability. Control data was collected at different times than experimental data, and infrastructural demands seem to have driven the control data availability to be less than that of other, managed approaches. Overall, this evidence of variable performance due to external provider demands leads to the conclusion that overall, the availability costs of the various approaches are in fact negligible.

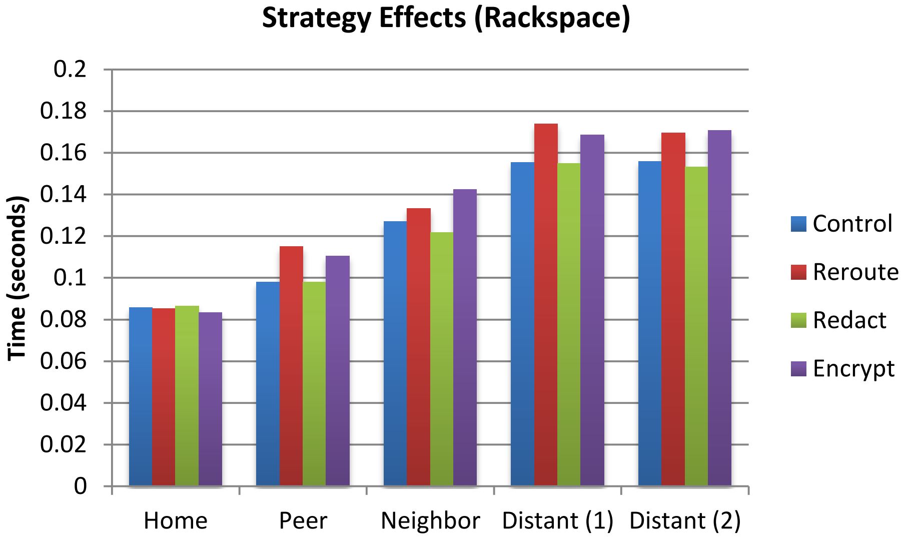

Figure 6. Rackspace Timing Results

Figure 6 shows similar results to Figure 5. Here, the query times are much higher for the home and peer nodes, but actually lower for the distant nodes. In this case, the content is still hosted in Amazon's infrastructure, but the testing node is at Rackspace. As a result, the longer response time for content from the home node is to be expected. Queries to distant nodes are actually shorter than the previous calls into distant nodes from Amazon. This stems from the fact that the distant nodes are both hosted at Rackspace. This locality shortens the round trip distance for a request. Previously, from Amazon, a content request would need to travel from Amazon's east coast data centers to the Rackspace data center in Dallas, then back to the east coast for content, then back to Dallas, then back to the east coast. In this test, the request only travels from Dallas to the east coast, and back. Nevertheless, the overall performance profile is sensible, reflecting the expected shorter latency between home, peer, and neighbor nodes when compared to distant nodes. Similar to amazon, we again have cases when the control latency is higher than experimental latency, indicating some amount of infrastructure performance variability. In Figure 6 however, we see that overall encryption and rerouting impact performance more than redacting, as we would expect. Rerouting again has high overall impact, likely as a result of using Google's remote SMTP services.

Figure 7 Shows performance results measured from Comcast. Interestingly, they show significant variability when accessing nodes hosted at Amazon, and more predictable performance when accessing nodes in Rackspace's infrastructure. The overall variability does not follow the expected pattern of shorter response times when accessing content from nodes close to that content, except in a few cases. This illustrates the kind of performance variability one can expect from an external service provider.

Integrity impacts are the result of approach rather than platform. Redacting content destroys information integrity, as information is removed and not delivered to requesters. Encryption maintains integrity the best of the three alternatives as information, even though encrypted, is still delivered, and delivered in the context of the query response at that. Rerouting is better than redaction, in that sensitive information is still delivered, but worse than encryption, as it is not delivered within the response context and is sent out-of-band. Simulations removed sensitive information from the information network and dispatched it to a user's email address via SMTP over TLS when the selected strategy was rerouting. This impacts information availability, as email delivery times can be highly variable. In our experiments, delivery could take anything from a few seconds to a few minutes.

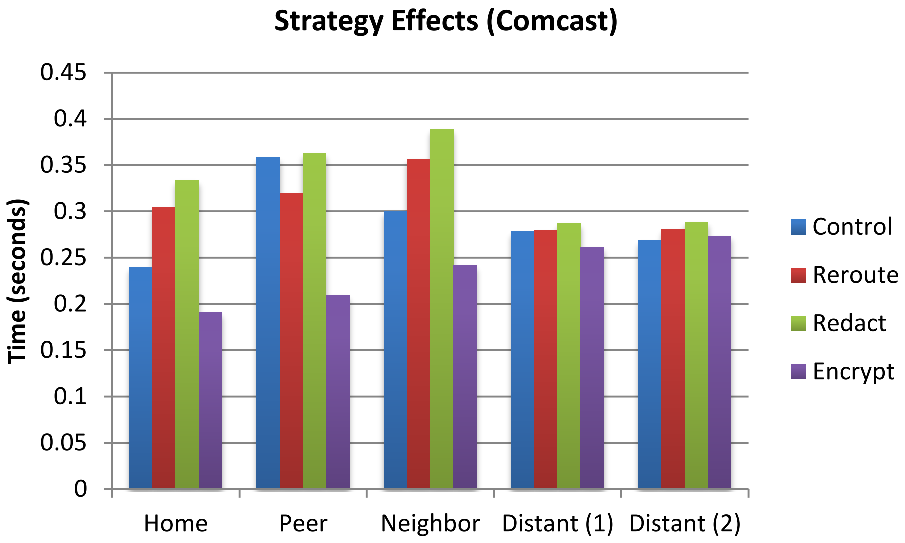

Figure 7. Comcast Timing Results

Confidentiality is likewise impacted primarily by approach and not by infrastructure. Redacting sensitive content provides the best confidentiality protection, as sensitive content is simply not exposed. Encryption is likely the worst solution from a confidentiality perspective as content encryption is a delaying tactic against a determined, well-resourced adversary. Rerouting may be better or worse than encryption as an approach, depending on the confidentiality of the out-of-band channel. If the security of that channel can be guaranteed, then it is likely a better approach. If, on the other hand, the security of that channel is more variable or difficult to ascertain, encryption may be a more reliable approach.

Overall, our results show that, from a performance perspective, the rerouting approach fares the worst, but only slightly. Both our results from Amazon and Rackspace, in Figures 5 and 6, show encryption as generally taking the second largest performance hit.

Table 2. Approach Evaluation Summary

	Redaction	**Rerouting**	**Encryption**
Confidentiality	3	2	1
Integrity	0	1	3
Availability	3	1	2

Furthermore, network effects have a much larger impact on performance than information protection approaches. Note that when queried from Amazon or Rackspace, the home node timing results are very close to uniform. Queries from Comcast, however, are much more varied, indicating more highly variable quality of service within the Comcast network. This is also supported by the gross distribution of response times. Within both the Amazon and Rackspace networks, the farther a queried node is from the content requested, the worse the performance, as expected. Comcast's network has a much more uniform information network response time overall as the processing time of the information network simulation is overshadowed by the highly varied performance of Comcast's physical network. Availability is surprisingly uniform across all confidentiality strategies, showing little impact on end-to-end processing times. Rerouting strategies show the most degradation, though that performance degradation is less than general network performance variation.

Table 2 shows the overall results of our experiments and analysis with respect to various possible approaches to securing information transiting content networks, on a scale of zero to three, with three the highest and zero the lowest scores, respectively. Not surprisingly, there is no clear best approach. Rather, decisions with respect to which approach to choose for given content is highly dependent on the sensitivity of the content as well as integrity and availability requirements.

4. CONCLUSIONS AND FUTURE WORK

The work described in this paper presents bounds under which to select specific confidentiality strategies for protecting information in content networks. We first described the state of the art of this kind of information protection in content networks, and introduced the current accepted protection architectures sponsored by the UCDMO. We then presented a related taxonomy of increasing information protection, describing their advantages and disadvantages and how they could be implemented. Next, we described our current customizable experimental framework for evaluating various confidentiality strategies. We closed with a description of and the motivation for our experiments over these networks, the results of these experiments, and analysis of those results. All simulation code is freely available via Github.

Overall, confidentiality strategy had little impact on information availability. Redaction, rerouting, and encryption all performed within similar bounds. Of these three approaches, redaction damaged information integrity the most, followed by rerouting, and then encryption, depending on the security of rerouting infrastructure. Redaction provided the most confidentiality, followed by rerouting, and then by encryption (as encrypted content is generally at best a delaying tactic given enough time for cryptanalysis). Based on these results, rerouting is likely the best general solution, depending on the existence of a secondary secure channel. Less sensitive information can still be delivered via encryption, especially if that information is only sensitive within a given time window. Very sensitive information can be redacted, but due to the related damage to integrity, this is only an attractive option when confidentiality is of the utmost importance.

At this point, our information network implementation has integrated three different configurable strategies for information protection, and routes information via an overlay network using HTTP. Longer term, this project will expand to both incorporate public-key encryption protocols and software defined networking (SDN) capabilities to provide physical control of information routing. We intend to provide public-key encryption capabilities via an integrated public key infrastructure providing additional privacy and non-repudiation abilities for the network and SDN capabilities via integration with OpenFlow. Shorter term goals include inclusion of different modes of operation, so that the network can support both request/response and publish/subscribe modes of operation, and more robust development so the system can run as a commercial grade security-on-demand service.

REFERENCES

1. Tallon PP. Understanding the dynamics of information management costs. Commun. ACM. 2010 May 1; 53(5): p. 121-125.
2. U.S. Department of Defense. Trusted Network Interpretation Environments Guideline USA: U.S. Department of Defense; 1990.
3. U.S. Department of Defense. Chief Information Officer, U.S. Department of Defense. [Online].; 2007 [cited 2012 September 29. Available from: http://dodcio.defense.gov/Portals/0/Documents/InfoSharingStrategy.pdf.
4. Hoover JN. Informationweek. [Online].; 2011 [cited 2012 September 29. Available from: http://www.informationweek.com/news/government/cloud-saas/229401646.
5. Pearson S, Benameur A. Privacy, Security and Trust Issues Arising from Cloud Computing. In Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on; 2010. p. 693 -702.
6. U.S. Air Force. SBIR/STTR Interactive Web Site. [Online].; 2011 [cited 2012 September 29. Available from: http://www.dodsbir.net/sitis/archives_display_topic.asp?Bookmark=41198.
7. Koponen T, Chawla M, Chun BG, Ermolinskiy A, Kim KH, Shenker S, et al. A data-oriented (and beyond) network architecture. SIGCOMM Comput. Commun. Rev. 2007 October 1; 37(4): p. 181-192.
8. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL. Networking Named Content. In Proceedings of the 5th international conference on Emerging networking experiments and technologies; 2009; New York, NY, USA: ACM. p. 1-12.
9. Ain Mea. D2.3 – Architecture Definition, Component Descriptions, and Requirements. Deliverable. Publish-Subscribe Internet Routing Paradigm; 209.
10. Ghodsi A, Koponen T, Rajahalme J, Sarolahti P, Shenker S. Naming in Content-oriented Architectures. In Proceedings of the ACM SIGCOMM workshop on Information-centric networking; 2011; New York, NY, USA: ACM. p. 1-6.

11. U.S. Department of Defense. Department of Defense Global Information Grid Architectural Vision. Informational. U.S. Department of Defense; 2007.
12. U.S. National Security Agency. Distributed Service Oriented Architecture (SOA)- Compatible Cross Domain Service (DSCDS) DSCDS Overview. In Unified Cross Domain Management Office Conference; 2009.
13. Booz | Allen | Hamilton. Distributed Service Oriented Architecture (SOA) Compatible Cross Domain Service (DSCDS). In Unified Cross Domain Management Office Conference; 2009.
14. Raytheon Corporation. Raytheon DSCDS Intro. In Unified Cross Domain Management Office Conference; 2009.
15. Ross R. Next Generation Risk Management. Unified Cross Domain Management Office. 2009.
16. Clark DD. The design philosophy of the DARPA Internet Protocols. SIGCOMM Comput. Commun. Rev. 1995 January 1; 25(1): p. 102-111.
17. Blumenthal MS, Clark DD. Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world. ACM Trans. Internet Technol. 2001 August 1; 1(1): p. 70-109.
18. Clark DD, Wroclawski J, Sollins KR, Braden R. Tussle in cyberspace: Defining Tomorrow's Internet. In SIGCOMM; 2002; Pittsburg, PA: ACM. p. 347-356.
19. Park J, Sandhu R. The UCON ABC Usage Control Model. ACM Trans. Inf. Syst. Secur. 2004; 7(1): p. 128-174.
20. Jamkhedkar PA, Heileman GL, Lamb CC. An Interoperable Usage Management Framework. In Proceedings of the Tenth ACM Workshop on Digital Rights Management; 2010; Chicago.
21. Jamkhedkar PA, Heileman GL, Martinez-Ortiz I. The Problem with Rights Expression Languages. In Proceedings of the Sixth ACM Workshop on Digital Rights Management; 2006; Alexandria, VA. p. 59-67.
22. Jamkhedkar PA, Heileman GL. The Role of Architecture in DRM Vendor Economics. In Satish D. Digital Rights Management: An Introduction.: ICFAI University Press; 2009.
23. Jamkhedkar PA, Heileman GL. DRM as a Layered System. In Proceedings of the Fourth ACM Workshop on Digital Rights Management; 2004; Washington, DC, USA: ACM. p. 11-21.
24. Heileman GL, Jamkhedkar PA. DRM Interoperability Analysis from the Perspective of a Layered Framework. In Proceedings of the Fifth ACM Workshop on Digital Rights Management; 2005; Alexandria, VA, USA: ACM. p. 17-26.
25. Koenen RH, Lacy J, MacKay M, Mitchell S. The Long March to Interoperable Digital Rights Management. Proceedings of the IEEE. 2004: p. 883-897.
26. Coral Consortium. Coral Consortium Whitepaper. [Online].: Coral Consortium; 2006. Available from: http://www.coral-interop.org/main/news/Coral.whitepaper.pdf.
27. Marlin. Marlin Architecture Overview. [Online].; 2006. Available from: http://www.marlin-community.com.
28. Openflow. Openflow - Enabling Innovation in Your Network. [Online].; 2011 [cited 2012 June 1. Available from: http://www.openflow.org.
29. JBoss. JBoss ESB. [Online].; 2011 [cited 2012 January 1. Available from: http://www.jboss.org/jbossesb.
30. Perez GM, Clemente FJG, Skarmeta AFG. Building and Managing Policy-Based Secure Overlay Networks. In Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on; 2008. p. 597-603.

BIOGRAPHY OF AUTHORS

Mr. Lamb is a principal scientist at Sandia National Laboratories in Albuquerque, New Mexico and a member of the ECE Informatics Laboratory at the University of New Mexico. Mr. Lamb has nearly twenty years of experience with systems development, ranging from custom microcontroller-centric devices to nation spanning distributed systems. An active contributor to the research community via talks, conference participation, and journal contribution, his primary research interests focus on increasing the security posture of distributed, connected computer systems via new approaches to information storage, processing, and transmittal.

Faculty leader of the Department of Electrical and Computer Engineering’s Informatics Laboratory, Greg Heileman also serves as Associate Provost at the University of New Meixco. A senior member of the IEEE, his research interests are in information security, digital rights management, game theory and machine learning. During 1998 he held a research fellowship at the Universidad Carlos III de Madrid, and in 2005 he held a similar position at the Universidad Politécnica de Madrid. He is the author of the text Data Structures, Algorithms and Object-Oriented Programming, published by McGraw-Hill in 1996, and has more than 100 peer-reviewed publications.