

# Information Security is Broken, Ineffective at Best.

By Harry R. Haury, CEO, NuParadigm Government Systems, Inc., hhaury@nuparadigm.com  
Prepared for SBIR OSD08-IA5, Michael H. Davis, CIV SPAWAR, IA TPO, TPOC  
July 4, 2009

## Abstract

The state of information assurance, IA, and our ability to protect our systems from attack and external manipulation has steadily deteriorated over the last decade. There are many reasons for this but foremost amongst them are the increasing complexity of platforms, the inclusion of many external uncontrolled vectors such as USB Flashdrives, drivers, executable components, and of course the use of internetworking with a steady erosion of the boundaries between systems, both by design and incidental to the explosion of the Internet. Existing perimeter defenses are inadequate, inspecting security into systems is reactive, addressing only known vulnerabilities. The vast majority of existing IA mechanisms have been designed using a closed hierarchical application model, which no longer applies. What is needed is a new framework for protection that establishes fixed security relationships between persons, computers, applications, distributed components and application communities that can cope with and protect the new virtual boundaries between systems whether found inside a computer or between participants across the Internet.

While this might sound difficult, emerging standards for communicating across these far flung networks are primarily message object or streaming data pipe oriented dramatically reducing the number of places and communication options that need additional protection. This revised model will be used to establish deterministic cryptographically protected connections between systems and participants in the environment. This will be done by creating a series of trusted software and hardware components that can insure that data objects and piped data streams flowing through the networks are what they say they are and in fact come from other trusted components.

These protection systems will operate as a message processing stack inserted between processing nodes as a new type of community boundary enforcement proxy or guard. It is the primary key to this system that these proxy guards will be compatible with existing protocols allowing the framework to support existing and new applications seamlessly. Further, all participants in the framework will require cryptographically bound access control and authorization assertions. In this way, all nodes in a particular community; including computers, humans, virtualized servers, programs, operating platform builds and others can be verified using the principal of determining Validity, Integrity, and Authority (VIA) at each "relevant" intersection of systems.

Surprisingly, this environment will actually become easier to manage and protect than the current one eliminating a number of current protection mechanisms and only requiring the development of a limited number of "new" components (none of which is novel / unexpected in the IA community):

1. Access control proxies for arbitrating the addition of new components, participants or programs into a community,
2. Cryptographically bound loaders controlling the instantiation and identification of all software components across the systems,
3. Data object proxies for the controlling the flow of discrete data objects, code and process invocations across the systems,
4. New standards for implementing highly efficient background movement and updating of certificate, token and key stores including the design of a new type of service oriented certificate/token manager to support community separation; and for implementing composite data and workflow objects.

In this enhanced environment data and streams can be protected and nothing will come into contact with our systems that does not carry specific authority to do so. While the new framework will

eliminate some current IA devices, there will still need to be protections for the network itself using routers, firewalls and other devices (such as intrusion detection systems (IDS) and host based security systems (HBSS), where these protections are typically associated with computer network defense (CND) capabilities, a major element in our collaborative defense in depth / breadth approach). This gives us better situational awareness and finer control of our networks and applications again creating a new framework for trusting our systems in these new widely distributed information sharing environments.

It is imperative that the Department of Defense undertake the development of proof of concept and early standardization efforts of this protection framework to work toward actual field hardening and deployment within an 18 to 36 month timeframe. Efforts to utilize Net Centric systems and web services have put our capabilities at risk but this framework offers an answer to the problem, with minimal development effort, costs and lag time. The risk of not revising our IA approach and moving forward as expeditiously as possible is not acceptable, as we collectively continue to put our systems and data and people at much greater risk, which is growing exponentially. Success will also help the commercial sector secure its infrastructure with a common broadly deployed IA solution architecture.

## Background

Current Information Assurance (IA) approaches cannot effectively protect our systems and data in the evolving ubiquitous net-centric environment – the approach currently being used is extraordinarily complex, designed for a different protection systems design paradigm and in many cases obsolete. While most IA/security subject matter experts (SMEs) would agree that there are serious IA issues, the broader issue is how to step back and fix the problem rather than continuing the current approach of piecemeal patches reacting to security vulnerabilities– adding complexity without addressing the root causes. The aim of this paper is to discuss a new approach which is a significant departure from the current ineffective pattern, exploring a new framework for security in the day of loosely coupled web based systems broadly sharing information across communities creating an environment with bound risk at the application layer.

This more refined approach defines an updated architecture that is simpler, systemically consistent, significantly more flexible, sustainable and most importantly, affordable. This architecture can be thought of as a new type of information or data centric model that will act as an unobtrusive middleware layer allowing it to work with legacy and new systems alike with the minimum possible intrusion into the applications. This paper presents a notional concept as to how to actually build an IA framework centered on a revised reference concept that of an enterprise trust environment that strictly controls privileges in our complex, netted systems. This framework is built from a collection of standardized off the shelf trusted components that plug and play within the framework creating an adequately assured environment tailored to the exact policy, community and application requirements with minimal system modifications. While the approach does mean investing in the development and certification of new methods / approaches to IA, it can be done in conjunction with the current IA/CND architectures, with a relatively small investment. The framework will also replace a myriad of expensive, inflexible components providing a significant overall ROI compared to current IA approaches. But more importantly, provide a system with an actual, known, adequately secure, end-to-end, enterprise, meshed environment.

Otherwise, the Department of Defense (DoD) and Intelligence community (IC) will keep throwing large amounts of money away on an approach that cannot work, creates unsustainable complexity, and is always in a reactive mode - versus investing a relatively small amount of funds to rebuild IA capabilities for the emerging system paradigms of Web Services and Net Centricity as the best way to protect our nation well into the future. As a forewarning, the basis for this framework is a complex analysis of the way systems intersect in the new meshed and networked world, the analysis attempts to develop a complete technical basis for the framework. Yet because it will align perfectly with the systems implementation patterns it will actually be an order of magnitude easier to implement and sustain than current systems, which seem simpler but actually fail for exceedingly complex and sometimes technically obscure reasons. We use as a basis for the scope and boundaries of our approach the response to the question “what really matters” in effective enterprise IA, from the community position paper sponsored and coauthored by SPAWAR, *“Enterprise Software/SOA IA/Security Approach”* (AKA - Clarifying the “Fog of SOA IA”), where our approach described herein both facilitates and simplifies the four major barriers / capabilities called out therein:

1. Specification and enforcement of specific IA standards and their future extensions and options. Whereas in an OA environment it is necessary to have both an overarching IA enterprise architecture (EA) and standards specified (e.g., TV-1/2), those are not sufficient. There must also be an overall profile-based implementation level guidance to make it all interoperable, secure and accredited.
2. End-to-end (E2E) identification, authentication & authorization (IA&A) that enables full spectrum cross domain access control supporting a broader “need to share” coalition and first responder environment, “unanticipated users” and potentially many non person entity (NPE) users such as devices and services. The critical lacking element is an enterprise distributive and transitive trust model and process that spans these mixed domains top-down dynamic policy –especially the Roles and Responsibilities/ Rules of Engagement (R&R/RoE) requirements.
3. Data/content centric security (DCS) assessed and integrated into our enterprise environment in parallel with the more typical “user / application” based security services approach (itself also an emerging art). DCS includes certain security aspects, such as: what metadata fields need to be protected and encrypted; what access control schema should be common for the discovery function (whether public or private); and how crypto-binding should be employed (including the IA metadata).
4. Dynamic policy execution for operations, management, legal, and security (the last being the long-term SOA implementation barrier) understood, assessed and integrated. This technical discipline is a huge and complex endeavor, embroiled in all levels of politics and mismatched expectations - where we do not yet have a federally proposed schema and organization to be socialized, let alone minimally implemented through digital means (though a couple of simple pilots exist).

Does this imply that the rest of the IA suite and capabilities, including all the other “protection” elements like Information Operations (IO) / Computer Network Operations (/CNO) do not matter that much? Not at all, in fact, the foundational IA functions such as CND, crypto/EKMS, PKI/CAC, etc and the hierarchy of IA protections required must be entirely considered to fully accommodate a managed risk

and best value IA environment. Those protections, for the most part, are typically satisfied within the core IT infrastructure often referred to as the common computing environment (CCE). The presumption is that, while the Navy may be behind in implementing improved and updated IA/security technologies and products, the overall IA/security functions are relatively mature and well managed at the DOD/DISA/NSA/GIAP level: UCDMO for CDS, ESSG for CND and NSA for crypto/EKMS and PKI. These IA capabilities should account for the typical foundational threats to include DOS, malware ID and control, and DNS spoofing. However, it should also be recognized that a high-level of configuration management (CM) rigor and especially enforcement (including all the dynamic settings in all the major IA products with baselines, monitoring, reporting, and controlling functions) must be in place and effective or the whole IA/security and risk posture might be minimized or even disabled to the point of ineffectiveness.””

## Introduction

During the course of the OSD SBIR and the investigation of practical means to insure anonymity as required by different missions while maintaining non-repudiation and audit-ability we discovered that the world has shifted regarding cybersecurity and that while these critical functions must be addressed as part of the overall solution, a significant gap exists in a more basic premise between systems - determining trustworthiness as it applies to policy and enforcing the policy in a trusted fashion. Our investigations show that we have a critical need for an enterprise end-to-end trust model, which is standards based and open architecture that can support anonymity and non-repudiation but also addresses directly the ability of one community of interest (CoI), system or participating entity to trust another. Thus managing cross domain access control effectively becomes a critical aspect of the collective approach to providing adequate IA across most systems and environments – as in essence, all transactions have a cross domain requirement, either implicitly or explicitly.

There are a number of new concepts discussed at length in this paper but the premise of this paper is that:

- Current IA processes have become too complex and sap performance unnecessarily,
- The world of Net Centricity, Service Oriented Architectures, and Web Services have rendered many of our existing notions about how to secure systems ineffective, thus obsolete, and
- These current problems arise because of the shift in requirements and the inability to trust participants, the blurring of boundaries between systems and exponential increases in system complexity.
- Given the reasonably well know state of computer network defense (CND) (assuming configuration management is well enforced (it’s not currently)), the IA end-state advocated by this paper is one where a new capability to secure meta-data, authorizations the current state of an object and transmit it along with the associated data within a reasonably assured messaging communications network operating in a fully meshed infrastructure.

Within this context, we propose a new way to build IA into our systems with a comprehensive framework for approaching and solving the difficult problems we have all been wrestling with for the

last ten years as the Internet has emerged. This new model centers on the creation of an enterprise trust environment and the controlling, limiting of privilege in arbitrarily complex intersecting systems of systems. To accomplish this we need:

1. An end to end framework for assuring, distributing and implementing trust.
2. IA systems need to be decomposed into their common basic elements, which are methods of endorsement and obfuscation, data arguments, and workflows implementing application level interoperability. (aka – using IA building blocks with pedigrees)
3. Existing monolithic protocols should be further structurally decomposed and encapsulated so that the components can be strung together in flexible, interchangeable, and dynamic ways.
4. Endorsement carries with it a concept of character, within these new systems understanding and trusting character becomes extremely important. Character is meant in the context of this paper to capture all the meta-concepts we use to describe data today as they may apply to a system of systems. It is the composite of what you know about an object, for example who created it, what authority the creator had, the history or provenance of an object, the community to which it belongs and so forth.
5. Monolithic protocols are then replaced by structured workflow standards that describe how the encapsulated pieces of the overall protocol are put together and in what order.
6. A new approach to building Policy Enforcement Points (PEPs) (along with the associated Policy Decision Points (PDPs)) is also espoused using a new stack-wise approach for all policy tests known as validity-integrity-authority testing, or VIA. Please note, this concept of authority is much broader than that normally discussed and is also critical to both the access control and digital policy execution approach barriers mentioned earlier. This authority is a combination of the required object character necessary to determine if an object should be processed in a particular way and the rule sets governing whether particular object values are consistent with the character of the object. (note: a subset of this model has already been implemented in various incarnations of the authorization Based Access Control (“ZBAC”) model and since access control is governed by authority and not identity in this model it can seamlessly support anonymity with auditability, as required by many systems. The *“ABAC to ZBAC, evolution of access controls”* paper is an implementation approach that is much more efficient, secure and actually works in mixed / cross domain environments- unlike most current IA&A approaches )

In appendix A to this paper we discuss a notional implementation of this framework and offer a discussion of many of the security lapses government, commercial and private entities have experienced and the risk that these vulnerabilities pose to the country in Appendix B. Essentially the magnitude of the problem continues to grow exponentially with increasingly severe and frequent attacks on critical systems worldwide. Nation-state based attacks and well funded cyber criminals can easily penetrate many of our critically important systems. And worse, the current computing infrastructure does not offer an answer to this severe problem. All the while the attacks become more sophisticated and increasingly difficult to detect. Although a somewhat controversial claim, a number of sources across government and industry including congressional testimony from major commercial representatives and

the FBI now report that the problem has now grown to the point that the costs of illicit activities from cyber attacks are estimated to exceed the value of the international drug trade. Current defenses rely on relative obscurity and making your target a harder target than your neighbor's but the problem is that some targets are of high enough value and profile that they cannot be hidden.

The state of our systems is in constant flux with us trying to catch up to unexpected asymmetric attacks, as we plug a hole two more leaks show up. We cannot afford this style of defense (continuously chasing new threats that quickly morph before we can implement mitigations enterprise wide) as that always leaves us at risk to the unexpected and it is unaffordable to fight an asymmetric attacker when the attacker gets to choose one of a few quadrillion ways to come at you. A new approach is critically needed, and it is needed now – ideally based on consequences based enterprise risk management approaches, not just the known threats, which are elusive at best.

While many of these issues, as raised above, are at least beginning to be understood as problems the solutions are lagging far behind. As we worked on the apparent paradox of providing secure and auditable connections while protecting anonymity we happened on a broader and more important generalized solution to the trust problem that we believe will enable us to make our systems adequately safe again while still serving the broader goal of using open systems broadly connected and sharing data. The goal is an IA vision of ubiquitous information dominance empowering commander's intent and ensuring decision superiority (while embracing critical business drivers). Restated in more practical, less technical terms, giving the right access, to the right folks and systems, at the right time, anywhere, anytime, with the appropriate level of assured availability and data quality protection, while also minimizing data loss in an affordable fashion as measured against risk and rewards for the enterprise at an affordable cost.

## **Evolving Systems Context and IA Justification**

The environment consists of and will continue to consist of heterogeneous components with varying levels of protection from external and internal threats. These protections range from air gaps and strict operational controls to sophisticated technological approaches to solving the IA issues our systems face. We can assert the following:

1. Heterogeneous systems and components will always exist in our various networks,
2. There will never be a 'single' standard, prescriptive, approach to IA, as it is not theoretically possible to condense all the systems into a common syntactical framework although portions of the systems will gravitate toward standards as they develop, and
3. IA will always have significant operating and processing overhead by its very nature.
4. With net-centricity, and the shared vulnerabilities it brings, fully assured systems that are 100% protected is neither obtainable nor affordable (e.g., we must dynamically determine "what is good enough security" for the mission at hand...) and understand the long term consequences of opening up access in the heat of battle on long term warfighter decision superiority.

It is best to think of these intersecting spaces as multi-dimensional vectors represented by Venn diagrams, each having overlapping and unique requirements. The key will be to determine what can be done to secure these intersections without violating the IA requirements of the data owners while ignoring or simplifying those dimensions that do not have to be protected to keep risks contained. We need to develop a system wide philosophy about what is good enough based on risk assessments and trading off the cost of successful attack against the cost of protecting the system. These notations by themselves would be an important contribution to the community allowing rational decisions about using one system versus another. It must also be remembered that there are many costs to security, depending on approach:

1. Processing overhead,
2. Latency,
3. Bandwidth absorption,
4. Critical system identification,
5. Mobility impacts,
6. Failsafe and “battleshort” effects,
7. Administrative burdens, and so on.

The trade-off equation should be:

$$\sum_1^n (\text{Cost of any known or unknown but possible, Loss}_i \times \text{The Probability of said Loss}_i) > \sum_1^n (\text{Cost of Securing Against this Loss}_i)$$

Where “i” is any particular risk vector for “n” vectors if it is demonstrated that each vector is fully independent of all other vectors then the equation for protection tradeoff would be independently evaluated for each vector. It doesn’t take much research of today’s headlines to realize the phrase cyber-security is almost an oxymoron. For a variety of diverse and complex technical reasons information technology is at a crossroads where the potential for information sharing and large scale integration have driven the extension of systems into extremely unsafe territory. Without belaboring this obvious fact we need to ask whether the security risks of our systems can be properly managed and mitigated, what has to be done in addition to our current efforts, and what priorities should be applied to addressing these problems. An examination of the current problems and risks to our country are reviewed in Appendix B.

## Major Systemic IA Issues

The primary issues affecting our systems are sometimes lost in the milieu of attacks and breaches we hear about. Many of the problems we are currently encountering originate from just a few basic tenets that were ignored when our current systems were built. These originate from naive assumptions, changes in the basic computing landscape and a failure to appreciate the developing storm of cybersecurity problems. The basic issues that must be addressed are:

1. Identity and Authority to act are critical factors in controlling how an individual and non-person entities (NPE) act within systems, an end to end fabric for appropriately managing and enforcing relationships between requested or inferred actions and their authority to do so must be developed and deployed. These solutions must also be able to interoperate without loss of resolution between systems that is caused by inexact transformations or mappings. This is further complicated by the inappropriate use of identity instead of authorization to open systems access capabilities up to participating entities. (Please Note: in this environment Authorization, becomes the transportable concept that moves between system boundaries whether identity is available, irrelevant or purposefully obscured to maintain anonymity. Identity and authority are separate concepts. By keeping them separate, where possible, the use of cryptographically bound authorization allows the adjudication of access control without identity.)
2. Trusted platforms must exist to properly and provably implement the desired actions across emerging systems. As can be seen from weaknesses of systems to prevent viral attacks, phishing, malware and other direct sabotage of the computing environment, existing off the shelf platforms do not provide an adequately trustworthy environment for enforcing action integrity and preventing unwanted actions from being implemented. Any solution to this problem must be nearly cost free and should be integrated into the computing platforms and operating systems directly. Without a ubiquitous, simple to control and simple to manage environment that is 'user proof' these systems will continue to be susceptible to subversion. While there are certainly limits to what can be done to protect from social engineering attacks, current systems routinely admit users that have no or fraudulent credentials. By creating systems that have detailed information about relationships and privileges the scope of such attacks can easily be limited. Other means will be required to protect from insider attacks or corruption of inside systems.
3. A trusted fabric for disseminating, managing and enforcing complex policy and workflow steps between entities must be developed. The emerging Net Centric and web-based applications have shared components, shared access, and distributed highly eccentric participants, all outside any current concept of centralized control, therefore policy intersections must be able to reflect the nature of the complex interactions of such systems. The interactions must understand the authority of participating entities, the degree of trustworthiness around their current operating context and connections, and the applicable rules governing the entity's behavior within those contexts. The currently envisioned policy frameworks are entirely too flat and inflexible to properly manage these tasks although they are a good start. These policy enforcement points, except in the case of hierarchically bound fixed systems, must be able to dynamically assemble relevant policy and make policy enforcement decisions on the fly, all with minimal performance overhead – that is, an implementable enterprise dynamic policy execution approach must developed and deployed. They also have to be able to be applied to routine Internet access via port 80 and 443 openings in the firewalls. Trusted and un-

corruptible web browsers might be part of a workable solution but something watching what is loaded and will prevent dangerous components from entering would be a strong step forward.

4. A data/content centric security (DCS) scheme for protecting data from disclosure and tampering must be implemented in a way that protects the integrity of the underlying systems and should be interoperable between systems or it will not support the broader goals of information sharing and Net Centricity. The data centric security fabric must therefore be constructed in a fashion that supports flexible evolving needs, can deal with 'virtual' instead of physical boundaries between these emerging systems of systems, and understands computing context and composite policy. This implies a new capability for protecting data outside and within distributed applications, something new under the sun, a pervasive concept of multi-domain data protection similar in architecture to our concepts of multi-level security.

## Urgency of Action

The vulnerability of our critical systems to attack proves it is time to take these issues seriously; meaningful and comprehensive efforts need to be undertaken immediately to protect our networks, systems and data with the highest national priority. While addressing the four major IA gaps listed in the previous section, the focus should be on the development of capabilities that are cost effective, manageable, and massively scalable to provide for the following functional outcomes:

1. Elimination of forgery across the protected systems ranging from packet spoofing to transaction forgery through strongly protected physical guards, filtering devices and high performance cryptography,
2. Creation of cost effective systems that can be trusted to only execute applications and instructions that are known to be authentic and authorized,
3. Creation of intelligent and deterministic filtering and firewall capabilities that block traffic that does not have the appropriate authorization or 'right' to assert the privileges it is asserting. This has to be functional at any layer of the OSI stack including Layer 7,
4. Protection of core routing infrastructure and common services such as DNS and BGP from external attack or compromise, otherwise generally known as the development of a black core network,
5. The trust of identity and authorization at the level of assurance necessary for the applications affected with sufficient granularity to manage privilege authorization appropriately while providing support for inter-domain anonymity when required,
6. Implementation of community separation to allow virtual application networks to control the flow of information between participants and limit the exposure to data exfiltration while still allowing communities to share common services and infrastructure, and
7. Creation of an end to end framework for managing and enforcing trust including data, character, application code, platform integrity, policy dissemination and enforcement.

The bottom line is that systems need to be trusted to do what they are supposed to do, at all times, and our infrastructure has become so vulnerable to compromise that it is already a massive national security issue ranging from ecommerce to the Department of Defense. The core to all these development targets and filling the major architectural gaps articulated above is the development of 'trust' where it is relevant and in the many instances where it can no longer be inferred.

## Net Centric IA: the Trust Problem

### Old Basic Processing Model

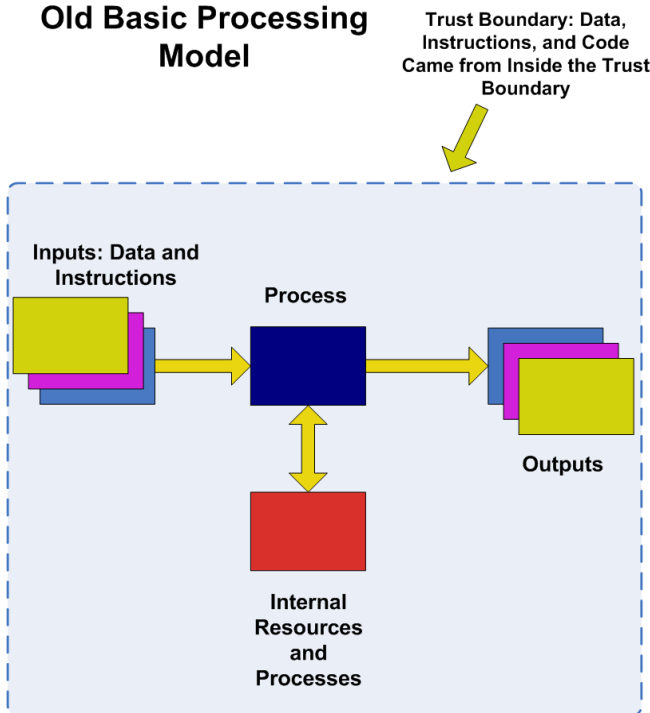


Figure 1

urban setting. Current systems add a whole new layer of IA challenges and complexities on top of the ones with which the community is already dealing. Just as with an insurgent action in the streets of a hostile city, "you should trust nothing that you aren't absolutely sure of: computers, systems, identities, networks, authority, browsers, anything, unless it can be 'adequately' proven.

The basis of these emerging IA problems is really quite simple to understand if the time is taken to examine the fundamentals of how the evolving Net

As we grapple with security in Net Centric and Web Services based systems it is important to explore what is different and why it is that security seems so difficult in this new world of information sharing. An analogy is appropriate here, we have traditionally viewed computer security as a perimeter defense issue but in fact it is more akin to the problems encountered with an irregular insurgency involving unidentified combatants in an

### Evolving Basic Processing Model

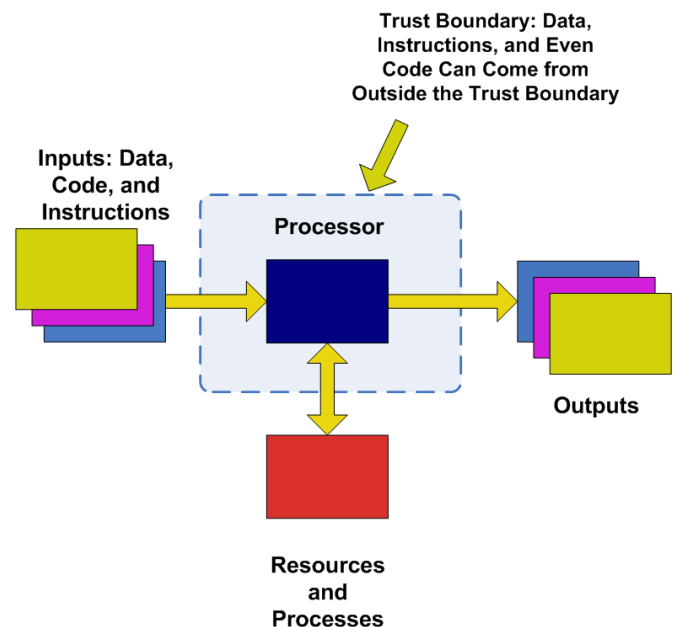


Figure 2

Centric and the Web based systems have changed the IA equation. To clearly see what's different it is useful to examine some simplified models. As seen in Figure 1, the basic processing model was that data or actions were input into the processing block where an arbitrary computing process was applied resulting in outputs or actions of some sort. With this model if we can be sure of everything that is inside the trust boundary the output is assured, we do not worry about the result. Traditionally we have designed isolated systems and built very strong protections against things getting inside the trust boundary that we could not trust or at least control. This

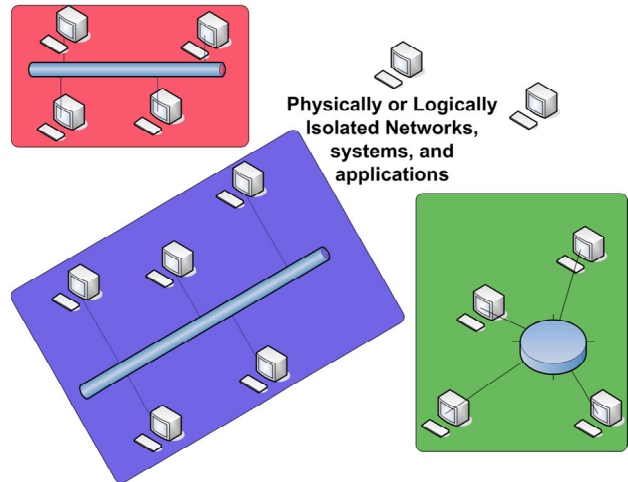


Figure 3: Circa 1989

model has been evolving steadily over the last several decades and has become something totally different, particularly as it applies to applications and networks. The external trust boundaries have been disappearing and our ability to understand what is inside the apparent trust boundary has been significantly diminished. The efficiency and power of larger and larger systems that broadly share information are in direct contradiction to the idea of maintaining such boundaries. A tension exists between the enhanced functionality of the systems and the risks/costs of them. This evolution has continued until many current systems begin to look like the model in Figure 2. The quintessential examples of the evolving basic processing model are the modern Internet Browser and web services. These systems may have a core of stable processing code but they are very permissive with regard to what data, instructions and code crosses the boundary of the respective systems. The prevailing wisdom is to attempt to analyze known attack vectors and patterns and use partial perimeter boundaries like NAT (Network Address Translation) based firewalls, virus scanners and heuristics to

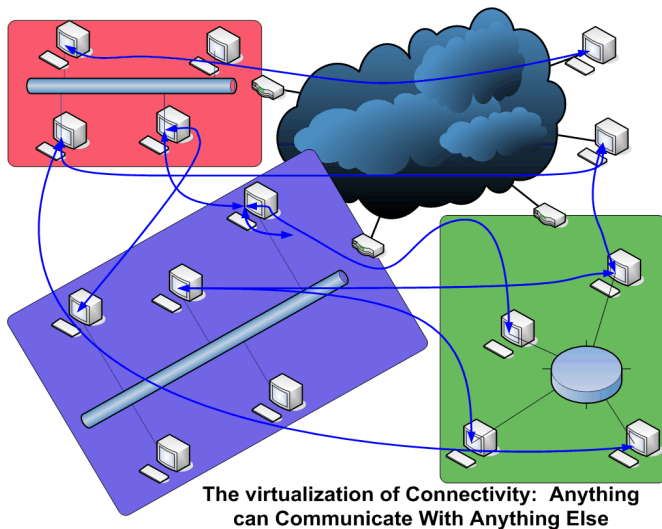


Figure 4: Circa 2009

protect our systems. But the current IA problems arise from the basic permissive and non-deterministic relationships between providers of content and services, and the consumers. Many systems architects and designers were not really ready for the risks because much their training came during periods of high **implicit** trust. Implicit trust is all but dead in today's open systems, soon to be followed by virus scanners looking for yesterday's known mode of attack; at least as these approaches apply to active defense against emerging threats. These mechanisms must be augmented with an end to end layering of deterministic protections if we are to get control of the current security problems and restore our

ability to **trust** systems. The concept of trust gets even more complicated as our traditional boundaries disappear and systems and networks are interconnected. Figures 3 and 4 capture the sense of this change in basic system assumptions at the network level. This picture becomes even more complicated when you introduce concepts of workflow across systems of asymmetric trustworthiness. All systems consist of a workflow between components, we have just taken this to a new extreme with SOA, Net Centricity and web services as shown in Figure 5.

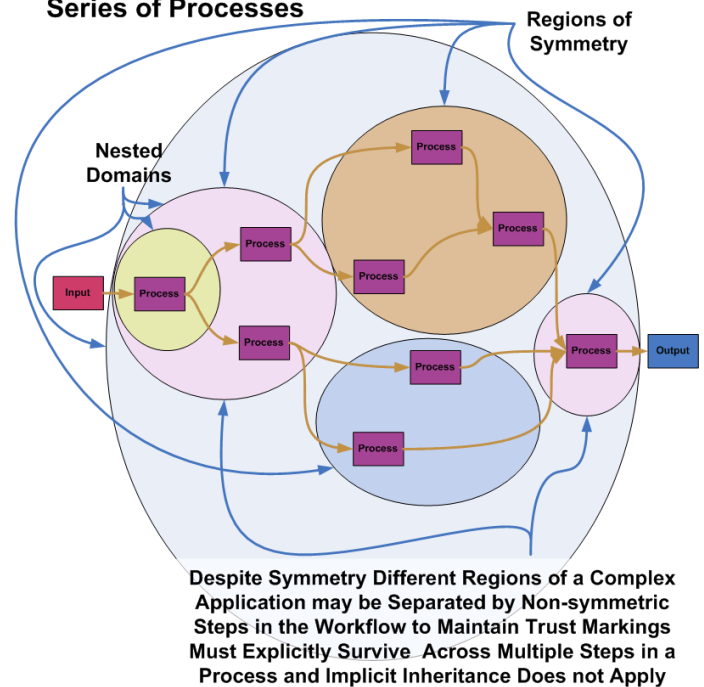
## Trust is Everything

Trust and trustworthiness mean a lot of things in the context of computing, in general, as well as in the specific instances of net centric and web service based systems. Knowing something is well formed, authentically from a known entity and is authorized to intersect the systems in the way asserted explicitly or implicitly is critical. In such systems a series of processes, numerous inputs and outputs are strung together in a workflow, a specific sequence of steps which form a multi-step composite process. For purposes of simplification, it is easier to aggregate portions of a process that exist within the same trust boundary as a single process. It is easy to see from Figure 5 that this trust issue is going to be very complicated, but let's explore it further, "what are trust and trustworthiness"?

To answer the questions we have to go back to the basic process diagram in Figure 1, to trust a process means that we trust, "adequately" by some common taxonomy execution agreements, the data inputs and instructions, the processing code and all of the other layers of services, drivers, APIs, protocol handlers, the processor and, so on. In a multi-step workflow, the output from this process must also be appropriately trustworthy for the next process or processes in the series. These components include hardware, software, and human interaction within the trust boundary.

The security industry has complicated IA unnecessarily with the creation of all sorts of composite processes that are given names and protocols for implementation when in reality there are only a limited number of things that you can physically do with cryptography, there are a number of algorithms for implementing various processes but to understand the nature of the IA challenges in the Net Centric and web services worlds it is important to decompose IA into its simpler constituents (e.g., the various levels of trusted IA building block / capabilities) because in the net centric world the interplay of these components is inherently less trustworthy than in the isolated systems world. For discussion purposes

### A System Consists of a Workflow\* or Series of Processes



\* Depending on the context a workflow might also be known as a protocol, dialog, orchestration or transaction

Figure 5

let's consider a slightly modified version of the basic process model. This discussion will center on the more specific abstraction of IA away from the morass of complexity that we have made it into.

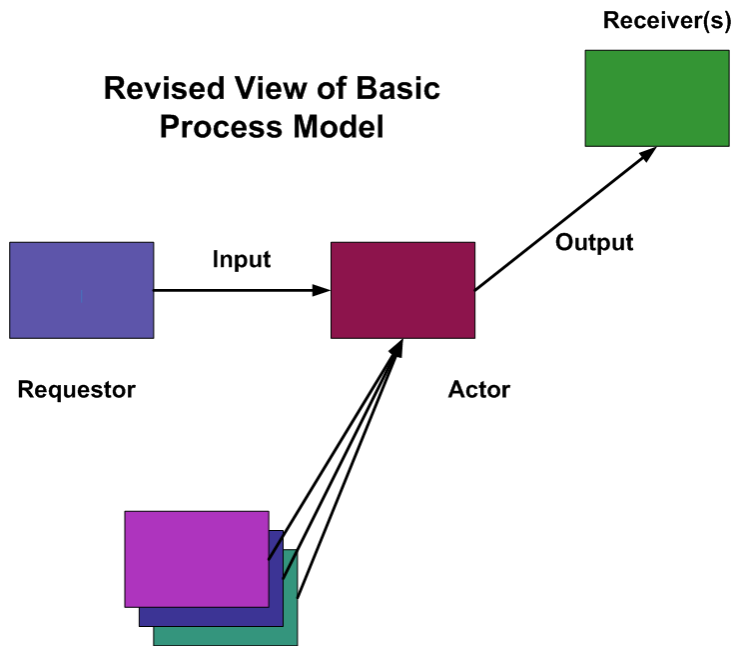


Figure 6

In this basic model there are only so many pieces:

1. Requestor
2. Input data or instructions
3. Actor
4. Process
5. The Output data
6. Transport of Inputs
7. Transport of Outputs
8. Loading of Processor with processes

So in this basic model we have to trust these 8 components. This is pretty simple but as expected there are some additional complexities in the loosely coupled world of the Internet and Net Centricity. We also

have to ask many questions about relationships or 'social' characteristics of the participants, some of them are:

1. Does the Requestor have the authority to deliver the inputs to the Actor?
2. Does the Actor trust the process code that is loaded?
3. Does the Requestor trust the Actor to execute the appropriate process?
4. Does the Actor trust the Receiver to properly handle the outputs?
5. Does the Receiver have the authority to receive these inputs from the Actor?
6. Does the Receiver trust the Actor?
7. Does the Actor have the authority to deliver the outputs to the receiver?
8. Does the Actor trust the transport method of the inputs?
9. Does the Actor trust the transport methods of the outputs?
10. Does the Requestor trust the transport methods of the inputs?
11. Does the Receiver trust the transport methods of the outputs?
12. and a number of others.....

In addition to direct social questions, there are a number of ‘transitive’ questions that get asked, some of them are:

1. Does the Requestor have the authority to access the Receiver through the Actor and the “implemented” process?
2. Does the Receiver trust the process run by the Actor?
3. Does the Requestor trust the process run by the Actor?

The number of permutations of potential questions are quite large with just a simple process model but when the composite models of the real world are introduced the potential ‘social’ and ‘transitive’ questions that could be asked become astronomical as in Figures 5 and 7. In order to handle this complexity the community has attempted to reduce the number of uncontrolled

items and we have then applied complicated security dialogs to the process of determining if something is allowed and authorized or not. These models become even more convoluted when parties do not trust each other directly but instead rely on the endorsement of another party for that trust. This trend toward ever more complex IA processes is unsustainable and doomed to failure. A new model is required, this complexity is mathematically limiting, drains huge amounts of the total resources and instead of fostering flexibility actually is inimical to it. In short, systems must be built more effectively, simpler and predictable, if they are to be secure and scalable.

## What Makes this Such a Difficult Security Problem

1. **Assume nothing and Trust No One:** Base assumptions are no longer valid there are no common assumptions about someone else’s system that can be made. Conformance to standards for operation, platform architecture, information used all have to be explicitly spelled out and some level of attributes assigned to most. Not just for reasons of common governance but to reduce the number of possibilities to something that is tractable. Relevant attributes, which are sometimes called meta-data or meta-character but we will simply call it **Character**, must be defined that are sufficient to describe relevant factors of authority, authenticity, validity, social characteristics, and ownership must be applied to the movement of data, code and instructions across relevant boundaries. Systems do not sit inside locked buildings; they come into constant contact with potentially unsafe systems. IA was simpler when certain limiting assumptions could be applied to people and systems, eliminating the need to explicitly verify trustworthiness. This is not the case any longer.

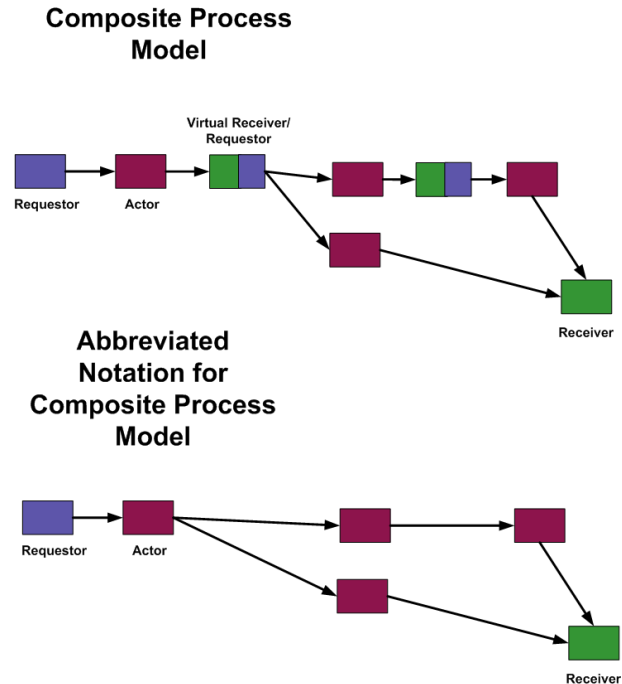


Figure 7

2. **Explosion of Combinations and permutations: Complexity of current systems makes them unobservable – thus uncontrollable and unstable:** Today's open environment defies understanding or even analysis. The number of moving parts and the associated number of potential interactions have exploded beyond the ability of most people to comprehend. The problem is well illustrated by using the example of a somewhat typical Windows PC. In this PC, the one being used to type this paper in fact, there are 2 GB of RAM running 5 high level applications, interacting with 65 processes, using 800 open threads, with 140 loaded services and 40 connected devices, all implementing their own drivers and DLL calls. This machine, having just been optimized and stripped down, only has 150 distinct drivers and a relatively small number, approximately 1,400 registered executable, DLL, .NET and .COM components. It is obvious how complex the interaction could be, as an example, to further develop the mathematics, let's use a simpler device that has three input devices, three associated input buffers and stack handlers, ten middleware message handling processes, three running core applications, four output buffers and stack handlers and six output devices. If someone is trying to comprehensively understand the potential security issues and vulnerabilities in this simple system they would have to fully understand 4320 potential interaction combinations assuming only one system from each layer was affected and that there were no loops, a ridiculously restrictive assumption for almost any system except a signal processor or router. In the "normal" PC being used at this moment the combinations explode, by selecting only 5 of the items listed randomly above to interact from a pool of 1500 objects you get over 62 trillion potential combinations. In other words, in a system of 5 components out of 1500 there are over 62 trillion different system combinations that have to be understood and protected. Introducing networks and services into the model drives the number of combinations toward infinity. In complex systems with ad-hoc and arbitrary interactions it is simply not possible, much less practical to protect everything unless we build our systems to "deterministically" limit the potential combinations and control the relationships. If you know and understand all of the players you can generally trust the actions of the group but in these modern systems there are drivers written in China, active-x components written in India, free anti-virus software written in the Russian Federation and these are just the components that are supposed to be there, we do not even know which components belong and which ones don't, and it is impossible without a real definition of **Character**. As the complexity and scale of systems grows the probability of an untrusted device or compromised component being incorporated grows exponentially. This is even further complicated by the lack of control of the supply chain, both nationally and internationally, in other words, don't trust systems built and provided, either directly or indirectly, by your adversary.
3. **There Are No Boundaries, Everything Needs Cross Domain Support:** Boundary assumptions have traditionally simplified IA tasks but suddenly, in a very real sense, there are no boundaries so assumptions about what is inside and outside the defense perimeter are completely obsolete. The reason for this is simple, the separations between applications and networks has slowly eroded except where real air-gaps remain, even then there are vulnerabilities such as sneaker net born viruses and malware. Recent examples include misbehaving DRM software on CDs and DVDs, and malware installed on USB Flashdrive Drivers.

When simplifying assumptions that do not apply, significant problems arise with our ability to control systems. Systems with shared resources also share vulnerabilities, for example:

- a. Shared DNS is used by all systems to resolve URIs and URLs, if one system successfully attacks DNS other systems depending upon it are at risk
- b. Common priority markings without cryptographic protection of such assertions. On any such system the availability of a local network segment to a high priority application can be denied by a lower priority application spoofing the higher priority markings and flooding the segment with traffic.
- c. BGP and Routing table updates using AI heuristics, a number of systems adjust routing tables automatically based on traffic patterns. Applications implanted in the segment being monitored can manipulate the heuristics and control the behavior of the network.
- d. Network time references are commonly used to determine the expiration of Certificates and tokens, a direct attack on time can keep certificates alive past their expiration or kill them before their time.

There are core functions that are shared and reused between applications creating interdependencies that never existed between applications, organizations, users, and domains before. This sharing means that not only must an organization's systems be trusted, the systems they come into contact with must be trusted or the damage they can cause strictly limited. In an environment of mixed systems it is critical that "virtual" domain boundaries be determined and that nothing be allowed to pass the domain boundaries that doesn't comport to policy. This new concept of encapsulation or "virtual" boundaries is critical to the semantic enforcement of IA in the Net Centric world.

4. **Current approaches to IA are too expensive:** Existing systems use processor intense performance sapping approaches to IA that in complex extensive information sharing systems asymptotically limit capacity and performance. Further, most high and medium assurance systems use expensive low volume equipment. These systems require mass deployment in turn requiring new price performance points, wide scale deployment, and innovative approaches to managing systems.
5. **The range and scope of systems connections is enormous:** Traditional systems had well defined boundaries; security was partially enhanced by this limited scope. When interacting with these vast connected networks the number of participating components, people and systems that potentially interact, particularly when considering the ability to infect systems not just directly but through several layers that cross points of domain separation means that efforts must be taken to limit the propagation or scope of damage across the systems. For example the spread of a virus across a network after certain shared services are compromised by a single computer. These networks will have millions of participants, some poorly behaved. To say the least this has not been the traditional security framework for organizations ranging from the DoD to large securities firms over the years.
6. **No common Framework for Interaction:** Asymmetry is the enemy and creates immovable stovepipes. Most of the power of the Internet comes from its scope and range rising from

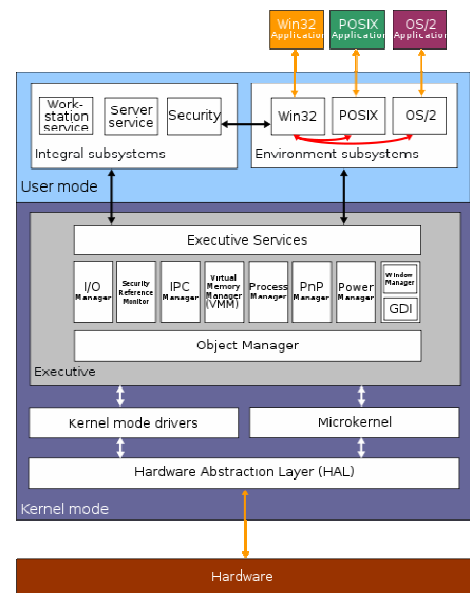
broadly supported common standards for interoperability. It is absolutely necessary to develop the same type of standards for IA, at least for transporting IA context and object character. The use of different algorithms and community standards for marking will still need to be supported with future systems but there is no way to gain economies of scale and interoperability without a means of easily hooking these systems together using “secure enough” common elements.

## Why not lock our systems away and out of reach?

The impetus behind Net Centric applications is to improve our effectiveness by efficiently and rapidly sharing data, in order to, “get the right information to the right people at the right time.” When IA is considered the statement should morph into, “always getting the right information, and only the right information, to the right people, and only the right people, at the right time and only the right time.” However, the systems we have were not designed for this **expanded** task, they were built assuming traditional physical barriers to entry would be in place and would be adequate. A new means for protecting systems is required, one that was designed with the understanding that boundaries are no longer physical. Although the technology is new, there isn’t really anything new to the pattern of attack used by malware. For all the talk about computer viruses and malware, cryptography and digital signatures, PKI and the like, these systems are vulnerable to the same pattern of attack as used in time proven scams used on the streets in New York, it is up to us to develop answers to the attacks that are the appropriate analog to how we deal with similar problems in the physical world. The problem is that as the physical boundary protections are being removed, our systems are no longer isolated from the “wild” and our protection mechanisms have not yet adjusted to the new threats.

## Trust, Boundaries and Complexity

It’s all a matter of **trust, boundaries, and complexity**. How can a system that is put together like the one at the right be trusted? (This is an architectural drawing of Windows/NT). The number of places that could be attacked and the millions of lines of code standing behind the system make even a relatively simple PC a prime attack vector for large networks! Modern systems are impossibly complex, from the modern automobile to the F/A 18F. They are built from thousands of components many of which in turn have dozens or hundreds of parts. Understanding all the possible interactions of these parts is difficult even for the technical insider. The history of engineering suggests that venturing into any new space brings with it unknown and unquantifiable risks. Net Centricity and the vast data sharing networks are replete with emerging risks from attack vectors no one expected and we must plan on that being the case in any future state. As stated before, the permutations of how systems can interact are almost innumerable. We have known how to build reliable complex systems for hundreds of years though and for inexplicable reasons have not used these



proven engineering practices in the design and construction of our information systems. Namely, build components with well defined performance metrics and only use those components within their operating limits or for the purpose intended. Perhaps it is the level of abstraction that obscures the issues but a sound engineering approach to decomposition of the problem set would resolve many of the problems we are having today. In the modern world of the Web and Net Centricity a few things are generally true:

1. Our commercial, government and defense infrastructure are the biggest prizes in the world. Penetrating them for illicit data exfiltration, commandeering them to turn them against us and interference with their promised functions is a huge risk and exposing these systems to the “wild” by connecting them to unprotected networks compounds these risks exponentially.
2. Few people understand even a meaningful portion of the technical infrastructure involved in a modern day networked applications, the damage that can be done is extraordinary due to the range and scope of these systems.
3. The asymmetric attacker has the advantage of choosing when, where and how to attack; meaning that the defenders have to be able to protect themselves from all vectors of attack, thousands of which may not even be previously known vulnerabilities. Traditional methods of defense usually rely on building strong barriers to the enemy: walls, “security moats”, defense perimeters, free fire zones, and heavily fortified demilitarized zones. Almost all of them are predicated on the precept that if we can keep the enemy out we are safe. This does not work in the Net Centric world it is as if the enemy is already among us (where we are our own worst enemy sometimes).

We have to protect the entry and exit from our systems in ways never thought necessary until web browsers and USB drives became ubiquitous. It is impossible to understand and protect everything in its native form factor, putting systems behind protection devices that limit the ways in and that provide control of what any system can do provides a workable answer to this complex problem. The required components can be developed and fielded relatively quickly and, if properly designed, can be built to leverage most of our existing systems investment.

As indicated previously, most protection boundaries in today’s systems, both large and small, assume physical or logical isolation can keep the intruder out but the reality is that we have gone from a ‘fortress’ or glass house design pattern to a social one. The difference in perspective is the same that is encountered between protecting money in a bank vault versus keeping the President safe during the inauguration. All of our systems are coming into contact with other systems that may in fact be contaminated or hostile. The infected systems are rarely notifying us of the danger as they transfer the infection to our systems. The current problem will continue to get worse as we drop our guard against physical and logical connections to the outside world. There are finite limits to what can be accomplished with current technology and infrastructure but more importantly there are fundamental flaws in our assumptions about what is even possible with the systems we have today. The most critical aspect of this situation is the deterioration of our ability to adequately secure our systems from outsider attack and exfiltration of data. This paper arises from principal research conducted by NuParadigm Government Systems, Inc. in a continuation of their IA efforts under a Small Business Innovation

Research grant sponsored by the Office of the Secretary of Defense and the original focus of the research was how to maintain anonymity in shared networks while providing auditability and security. What was discovered by the researchers during this effort is that the complex interaction of the requirements, systems and underlying architecture of the connected systems creates significant challenges to interoperability and security. While the NuParadigm has developed a model for managing auditability and anonymity in these shared systems the more important work arising from the SBIR is that a new framework for how to implement end to end trust in shared environments has been developed; to understand what this framework does, some understanding of the underlying impediments to securing an information sharing environment must be developed.

The more detailed security gaps created by the convergence of systems are:

1. A lack of physical boundaries between systems renders existing barrier protections ineffective.
2. The destruction of context, our current systems assume a lot about the participants and in turn use these assumptions in the design of IA protections. When systems are connected to the outside world the new participants rarely have the same characteristics as the existing participants. This lack of stable context often renders our protection mechanisms ineffective.
3. Physical boundaries are often an illusion. The sense that walls and firewalls are still intact creates a belief that we still have protection in place. But when we open these systems up to the outside world it is like leaving the back door unlocked and removing the guard from that entrance. Unless the transport of data across the threshold of our interconnections can irrefutably and independently establish, VIA, the validity, integrity and authority of the data object before it is allowed we will continue to be vulnerable to insider and outsider attack.
4. It is different in the "wild". As systems are expanded and the communities to which our systems are exposed are less well known the theoretical risks climb geometrically. The assumption behind all connections to the Internet or NIPRNET should be that the system on the other end of a data transmission is the enemy or surrogate masquerading as a safe participant until proven otherwise.
5. Attacks become stronger and easier in more broadly deployed systems. Two friends that know each other's voice and history can trust who they are interacting with but as the number of participants grows and the number of hops from system to system climbs it is easier and easier to implement attacks like "confused deputy" on the underlying systems. As the extent or scopes of systems expand, new social attacks of "triangulation" and "correlation" become a greater and greater risk. The opponent can glean more and more information about actions and behaviors through analyzing associated data flows in connected systems that correlate with real world events. We have seen simple traffic pattern attacks on our systems before but these connected systems give the attacker much higher quality and more extensive data about our behavior.
6. Misalignment of purpose/mission interferes with applicability of tasks. We try to expand our systems but each expansion of scope and mission is slightly misaligned with that to which it connects. As this type of connection is implemented the scope of data needed to accomplish the integration expands requiring the collection of data in one community that is not important to that community's mission. This leads to a loss of data integrity because a community that

does not need the data for its own mission will rarely take the steps necessary to make sure the data is accurate, complete and protected.

7. Loss of resolution limits the reach of these systems through multiple inexact transformations or data mappings and thus limits our ability to protect them. The analogy is that of a Xerox of a Xerox of a Xerox. Each copy becomes a little less clear eventually becoming impossible to read. Any system that connects across multiple hops will have this problem. The IA argument often centers around least privilege enforcement but if our data object goes through 20 or 30 hops of poorly aligned data transformations or mappings the meaning the application of policy to this object becomes meaningless. When exploring such a case with least privilege enforcement the number of participants permitted will be only a fraction of those intended. Since these losses of resolution occur silently at the transformation edges they are often not even recognized. The loss of resolution can cause a silent expansion or attenuation of the data and actions available depending on the bias built into these systems.
8. Compound versus simple policy intersections exist in shared systems making simple policy ineffective for appropriately deciding whether an action should be allowed or not. Since the compound policy is by nature an intersection of a number of independent policies from different communities, PEPs need to be able to adjudicate the intersection of multiple policy statements to determine appropriate actions. To complicate this further, policy has to structured to support commander's intent within the respective AOR, this is rarely the case now leading to huge IA problems when the AOR is changed on the battlefield or poorly understood out of context.
9. Fundamental architectural bottlenecks exist. The current mechanisms for enforcing IA across our networks often are dependent upon central servers for key retrieval, CRL checking and the like. These in-line calls to the central system cause a deterioration of reliability, a significant increase in complexity and unacceptable latency. The solution is further complicated by overhead in the distribution of community keys and certificates and associated refresh cycles. Many systems require non-local adjudication of authority and integrity placing a huge burden on the systems and creating unnecessary dependencies on remote systems.
10. Impossible scope explosion. As these systems expand the scope of their influence will reach unmanageable proportions where data can longer be trusted and the ambiguity caused by resolution loss mounts. This scope explosion leads to a belief that you have an effectively integrated environment but in fact the deresolved data flows are causing your systems to be misled by a flood of useless or, worse, misleading data. It is a simple IA fact that it is impossible to properly protect data that we do not have the ability to understand.

The best way to look at this problem is that complex systems are in fact built from a myriad of simpler components and each of these components was built, in turn, from simpler components each of which was designed with specific objectives in mind whether it was the storage or transmission of data or the analysis of hypersonic radar tracks. There were at the time the systems were built fundamental **assumptions** that were made about what the components had to do and within what context they were going to operate. When security is 'inferred' into our systems and the inferences are stable and appropriate it is possible to build secure systems using simple boundary protections but when we take

these systems and put them in a new context where the assumptions and inferences no longer apply it should not be a surprise that they are difficult to impossible to protect.

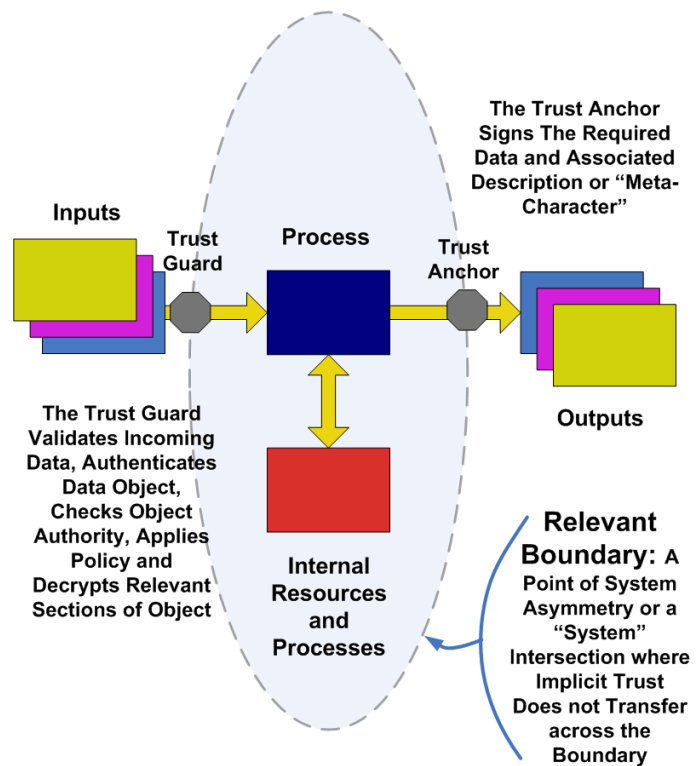
## How can such systems be protected? An End to End Framework for Implementing Trusted Systems and Mitigating Risk

Decomposition and determinism are key factors in successfully protecting these modern systems. In order to make sense of this, systems must be broken down into the relevant components. Relevant components are the level at which legitimate boundary assumptions can be made and below which granularity of trust is not meaningful to the higher order system(s). Once the framework analysis is completed new IA methods and capabilities must be implemented to efficiently protect the systems from attack and to mitigate the risk if a system is compromised. In practice, all security processes are workflows which have been simplified where possible, by using a set of assumptions about the trustworthiness of certain components in a system. These workflows are essentially composites of security and other processing functions that are put together to create more complex processes. It is our lack of discipline concerning the understanding of trust and the proper architectural decomposition of security that makes IA

so difficult, when in fact it is relatively easy at the primitive level. Reaching back to first principals then, what are the primitive concepts we are really interested in? This paper advances the notion that we have to undo a lot of conventional wisdom in the IA space and develop a new approach to solving end to end assurance problems. The proposal is to approach this problem the same way any exceedingly complex problem is approached, through suitable decomposition into interchangeable, reusable and scalable components. Through the text that follows the mechanisms for actually making this work will be discussed and a framework for implementation demonstrated. This new approach has a number of fundamental components:

1. A framework that is rigorously developed to enable and support end to end IA with consistent architectural elements and a flexible reference model.

### Basic Process Model



2. A focus on enforcing policy regarding Validity, Integrity, and Authority, or VIA at all “relevant” intersections.
3. A decomposition of conflated components that have resulted in inflexible difficult to modify IA frameworks.
4. An implementation architecture for composition of IA components so that workflows can be managed and enforced.
5. An object tunneling solution that minimizes severe privilege management issues in current systems, allows complex multi-domain intersections and tolerates system asymmetries.
6. A new comprehensive concept of understanding and trusting Character of objects in order to create and protect Virtual Boundaries between system participants.

## The pieces of a new framework for IA and Trust

In trying to make sense out of the overall problem the authors examined the problems associated with ambiguity and lack of interoperability. The proposed framework may seem familiar to many because the concepts have use in our current discussions; the difference here is that the movement to precise definitions and exacting architectural implementations is clearly proposed. Therefore the paper is offering precise definitions of these concepts and the relationships between them. The critical concepts and components are:

**Community of Interest** – a group of **Participants** that agree to adhere to common rules, object definitions, methods and use prescribed **Policy**, procedures and platforms to create symmetry and simplify assumptions about their interactions. The key aspect of this is that the specification of system interactions has significant inheritance problems. Specifications aim to build some level of **Implied or Implicit Trust** that allows the **Community** to dispense with having to explicitly mark **Objects** with their **Character** describing differences between sub-systems. The inappropriateness or failure to adhere to the **Implied Trust** mechanisms is often one of the primary security weaknesses today’s systems.

**Trust Boundary** – a point of asymmetry across which trustworthiness of the **Character** of a particular **Participant** or **Data** cannot be assumed or in other words across which **Implied Trust** cannot be assumed to transfer properly. **Trust Boundaries** may be nested with separate systems connecting through different **Trust Domains**.

**Policy** – a boundary rule applied to the **Character** of a transecting **Object** crossing a **Trust Boundary**.

**Encryption** – any process that makes the contents of an **Object** unintelligible or difficult to understand for sometime without access to a means for reversing the process, usually a key.

**Endorsement** – any process by which a **Participant** in a system marks an **Object’s Character**, or some portion thereof, as **Trustworthy** or that adds **Character** to an **Object**. The simplest example is a digital signature applied to a data **Object** using a private key where the associated certificate or public key is known.

**Transitive trust** – the delegation of **Authority** by one participant in a system to another.

**Authority** – the scope of actions allowed to a particular **Participant** in a system or the **Participant’s** relevant **Character**. **Authority** is encoded in a vector describing the **Participant**, this can be a simple description, a SAML or XACML assertion, or any other encoding that the community has agreed to.

**Validity** – the adherence of an **Object** to its particular **Type**.

**Authenticity** – the ability to determine that an **Object** is intact without modification and that it was created by a **Participant** with a known **Character**.

**Character** - the total relevant attributes of an **Object** or **Participant** used by a **Policy Enforcement Point, PEP**, to determine if an action is allowed or not. It is important to understand that the security markings themselves are part of an **Object’s Character**, for instance the **Trustworthiness** of the **Object’s** data payload is part of **Character**.

**Explicit Trust** – the use of coding and/or **Encryption** to convey the level of trust of an **Object’s Character**. This can be done by way of checking a CRL for the participant, using a security dialog with external references or by conveying a token with a known level of trust that **Endorses** the **Character** or **Authority** of the **Participant**.

**Implied or Implicit Trust** – the assumption, due to context, that an **Object** or **Participant** has a particular **Character**. An example would be a stripped down computer with limited or no access to open networks that sits behind locked doors with adequate tamper and perimeter violation alarms to prevent outside access or manipulation. Another example would be a computer that uses a content aware firewall with extremely strict connection limitations which also has a custom loader that requires all software to be properly signed before loading and all the components loaded are certified as safe with the keys sitting inside a trusted container like a CAC card. We may build systems that for administrative reasons we believe are safe on the other side of a connection, it is important to realize though that the simplifying assumption may put our systems at risk if an adversary figures out how to attack it. With these types of systems there are external means for defining **Trust Boundaries**. For this reason, particularly, systems should be designed to mitigate risk through behavioral analysis and controls. For instance a system and individual might have authority to access records with personal healthcare information but they would never need to access more than 50 records per day or 100 in a week and they should only access patients that are admitted in their area of responsibility, both levels of control uncommon to older systems.

**Tunneling** – the survival of **Encryption** or **Endorsement** processes across multiple **Trust Boundaries**.

**Trust Source** – the participant in a system that is the root of a **Transitive Trust** chain.

**Type** – an **Object’s** format and/or semantic specification.

**Participant** – and process, entity, or system that is capable of action sitting inside a relevant **Trust Boundary**.

**Trust** – the belief that some particular **Character** applies to **Objects** or **Participants**.

**Compartment, Trust Plane, Trust Surface or Trust Domain**– the total area circumscribed by a particular **Trust Boundary** whether contiguous or not. The concept of plane or, even better, surface is important here. A multi-dimensional surface may contact other surfaces and this is an automatic **Trust Boundary** at the point of contact. An example of this is the use of a **Trusted** network for transport because the **Participants** on the other side are **Trustworthy** for the purpose they are serving. Another example would be the use of common network services such as NTP or DNS, when a system connects to those systems they are **Trusting** certain things to happen or be available.

**Trust Anchor** – an isolated portion of a **Trust Domain** or **Compartment** capable of holding an **Object** and being a **Participant** which acts as a **Proxy** for **Endorsing** an **Object**. A **Trust Anchor** may sit inside its own **Trust Boundary**. An example would be a smart card or CAC card being used to hold private keys, tokens or certificates. A **Trust Anchor** provides for digital signing and or encrypting outbound **Objects**. They are also responsible for enforcing outbound guarding **Policy** features such as testing releasability or routing permissions.

**Trustworthy** – the ability to **Trust** that an **Object** or **Participant** has a particular **Character** with an acceptable level of certainty.

**Object** – data or code bounded as a discrete entity. Data streams are not objects until they are bounded either by beginning and ending or by being explicitly broken into sections with a beginning and end. A channel or switch related to a stream may be an object but the stream itself is not.

**Proxy** – a secondary **Participant** that is **Trusted** by the primary **Participant** to implement a particular process and that has the primary **Participant's** delegated **Authority** for the implemented process.

**Trust Guard or Policy Enforcement Point** – a process node responsible for acting as a **Proxy** for enforcing inbound **Policy** with regard to an **Object**. This node can sit inside its own **Trust Boundary** or not, depending on the risk of attack and the level of **Trust** between systems. When it sits inside a **Participant's Trust Boundary** it is directly at risk for the same attack as the **Participant** and the **Endorsements** of **Character** that the **Participant** is concerned about may be attacked directly.

**Bound Workflow** – The ability to link steps in a workflow so that all **Objects** transverse a **Workflow** as prescribed.

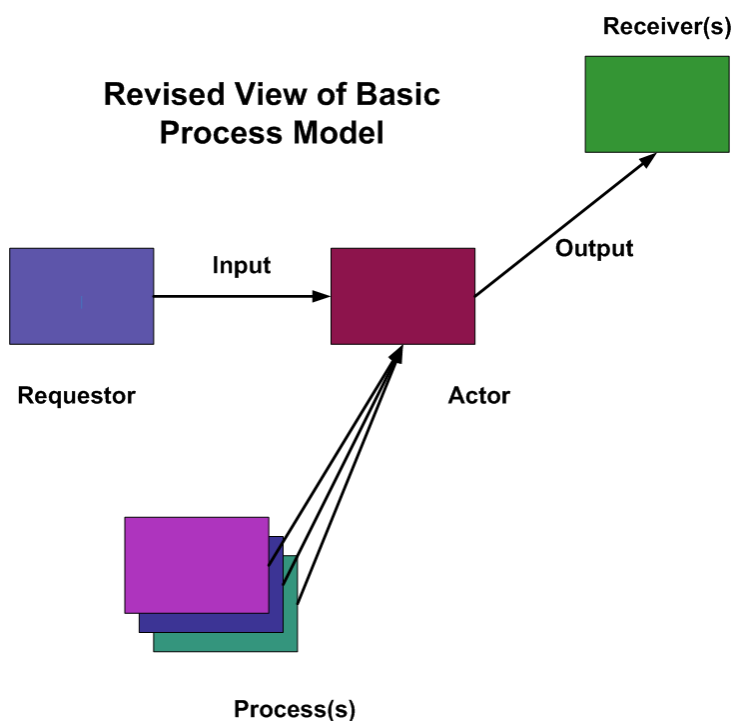
**Workflow** – a composite of processes and **Objects** that flow through multiple **Participants** in a deterministic set of parallel and/or sequential steps.

### **So how do we use this new system?**

Until we have unhackable high assurance devices and processes across all of our systems some of our new trust protections will have to be implemented via strict adherence to physical, operational and procedural protections. However, a number of specific actions should be taken immediately to start building better ways to construct trust frameworks and a high priority should be placed on building a few critical components needed for more robust **Trust** enforcement capabilities.

Is this complicated? Yes, it is but until we understand how we build up **Trust** within a system it is difficult to explain why IA fails or where gaps need to be addressed. IA failures come from a number of places but primarily they relate to:

1. **Trust** of the **Character** of an **Object** or **Participant** when it should not be **Trusted** often caused by failure to recognize the limitations of implied **Trust** to cross the **Trust Boundary** or inappropriately using **Implied Trust**. Methods of externally binding **Trust** at the **Trust Boundaries** must be further developed to **Proxy Trust** between systems that were not originally designed to talk to each other or were designed to partially isolate themselves from the actions available in the external regions.
2. Failure to recognize that a **Trust Boundary** exists. Systems need to be rigorously analyzed for **Trust Boundaries** and places where **Implied Trust** will not transfer safely or effectively.
3. Using poorly mapped or insufficiently granular descriptions of **Character**. **Character** asymmetries at the **Trust Boundary** are problematic, **Trusted** devices or processes must be implemented for transformation at the **Trust Boundary** with the highest fidelity possible. Least privilege must be enforced in these systems when transformation is not perfect.



4. The assumption that **Character** encoding is widely transferrable between systems is **not** accurate. **Character** will be highly eccentric to the applications consuming it, therefore **Encryption** and **Endorsement** methods used in a particular exchange, the **Workflow** process required to pass or block and object at the **Trust Guard** and the **Character Encoding** need to be fully independent instead of bound together as is often the case with today's security processes.

5. Use of **Encryption** or **Endorsement** processes that are insufficiently well protected or are not **Trustworthy**. The fact of the matter is

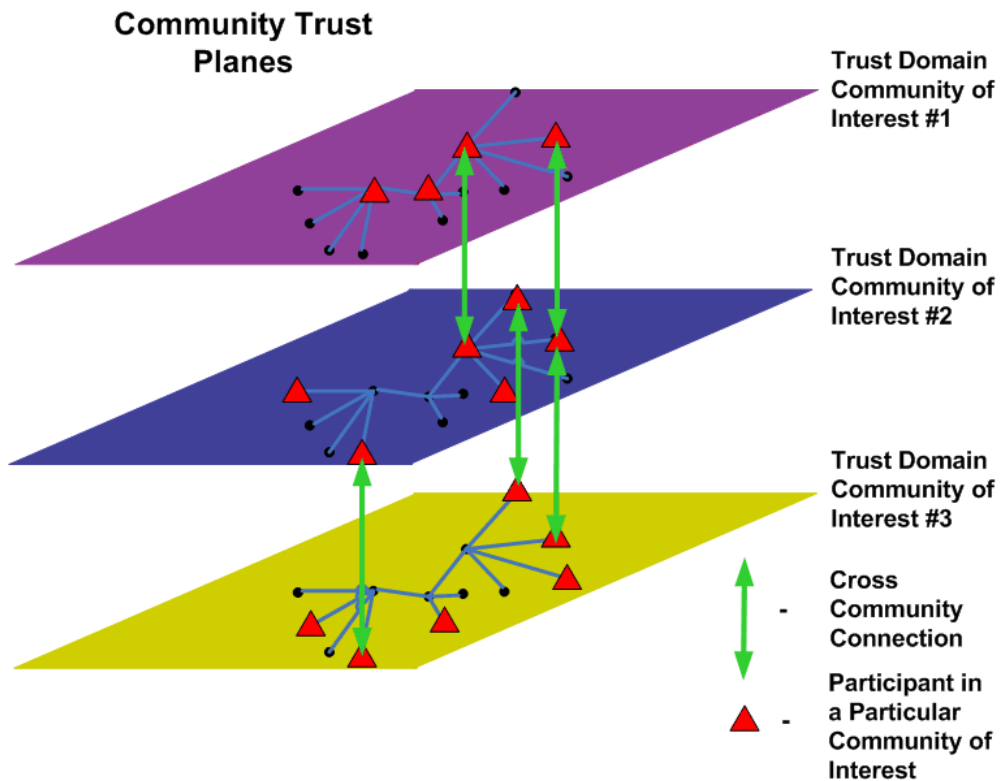
that IA concerns have bias due to risk and there are three ways that **Trust Guards** and **Trust Anchors** can be implemented: Sender controlled, receiver controlled or third party controlled. When assessing the risk of certain interactions they will have to be analyzed as to which pattern is appropriate. There are legitimate reasons for all three configurations.

6. Failure to protect **Character** or use compounded **Character** correctly. **Character** is more than just an encoding it has a different reliability depending on what is required by different systems. For instance, the **Endorsement** by the sender may be sufficient in one system but another may

require a verified “eyes on” review with a second **endorsement**. Or a **Cryptographically Bound Endorsement** that a CRL check was performed before proceeding.

7. Using **Proxies, Objects** and/or **Participants** that are not sufficiently **Trustworthy**. Devices with varying ranges of **Trustworthiness**, particularly under attack or capture are required immediately. These **Proxies** are needed to allow systems to be isolated from each other using a new type of firewall that tests **Character** and **Validity** of transecting Objects.
8. Failure to **Tunnel Trust** using sufficiently **Trustworthy Encryption** and/or **Endorsement** processes.
9. The use of **Untrustworthy Proxies**. A common fallacy is that systems are protected by Firewalls and TransSec pipes. These protection mechanisms are easily penetrated through compromised systems and expose systems to significant man-in-the-middle attack vulnerabilities in multi-hop systems. The **Proxies** must be sufficiently well protected themselves to mitigate risks.
10. Failure to properly **Bind Workflow** where the workflow produces a composite that itself needs protection. As an example, all net centric and web services based systems have significant vulnerabilities to insertion attacks. The systems need to maintain an ability to enforce proper **Workflow** from step to step in any composite process that crosses **Trust Boundaries**. The transactional route history of an **Object** may become part of its **Character** when it is needed. A far simpler pattern is to design systems so that they do not need **Workflow** history in the **Object** and **Character** can be used to control movement through a **Workflow** asynchronously.
11. **Participants** that touch more than one community are automatically **Cross Domain** in a **Trust** sense. They have to be guarded in a similar fashion to a multi-level device. This is possible if the **Participant's** can be **Trusted** to keep **Trust Domains** separated and guards are implemented effectively. Systems that are not adequately protected from subversion should not be allowed to touch more than one community unless such mixing of domains is an acceptable risk.
12. **Authorized Object** is an object endorsed by a Participant having the Authorization to do so. The mechanism of **Trust** is up to the implementer.
13. **Edge Application** is a process implemented by a **Participant** that sits outside a particular **Trust Boundary**.
14. **Choreography** refers to an integration pattern where entities do not have to coordinate their actions explicitly through synchronization or other means. Instructions and/or state might be carried within an **Object**, as an example allowing the **Participant** to act independently of all other **Participants**.

## The Framework:



## Validity, Integrity and Authority

VIA needs to be built into all devices and drivers intersecting these systems. Rigorous testing of object structure and separation of communities will allow the interoperation of many systems without threat of leaking control or data across virtual boundaries. All systems should have this VIA framework tested and certified against overrun and normal traffic attacks. If we implement this simple concept ubiquitously we can seamlessly eliminate malformation, overrun, and infiltration attacks. It is a simple filter in front of all application level interfaces that tests against the format specification for a community data model for proper formation, then releases the object to integrity testing against local policy for that object type and then tests the character of the object against authority and other policy restrictions and rules.

## Character

We confuse a lot of things when we start separating metadata by purpose. Often we talk of authority, context, state and other meta characteristics of our objects, often collectively known as meta-data. Character intersects with policy to determine actions to be taken on the object. Character is a general concept and should be thought of independently of specific meta-data and assertion encoding formats. The policy and assertion have to be aligned with each other and instead of interpreting the context we

should have a convention for marking the character format of a particular object. Character can be derived from data values as well, so it is possible for the data segments and character to overlap. In this case it is recommended that the character segment be separated from the data segment, even if they are the same, the reasons will become clear in the conflation discussion.

## Conflation

There are only two things we do with all our complex IA algorithms: endorsement and obfuscation. We have a terrible tendency to build complex protocols that then become inflexible, unwieldy and unusable anchors in real systems. The reasons are simple as we see more and more complex XML profiles being constructed for various purposes it seems to most that this is a better approach to interoperability but as complexity increases the risk of misalignment increases and by using composite structures it is difficult to modify systems as the need arises. We need to implement a radically new approach to eliminating these inflexible stovepipes using a flexible process control language to describe protocols and to allow the effective yet flexible enforcement of multi-step processes built from their basic primitive functions. So how do we put it together then, by using a simple functional or workflow compositing syntax as follows:

Data:

Type reference for object format and content constraints

Character:

Type reference for object format and content constraints

Applies to any native or compound object

Obfuscation

A function applied to an object of any type

Endorsement

A function applied to an object of any type

The workflow encoding then looks like this:

Character definition  $C_1$  applied to Data format  $D_1$ :

**$C_1(D_1)$**

Character definition  $C_1$  applied to concatenated Data formats  $D_1$  and  $D_2$ :

**$C_1(D_1+D_2)$**

Obfuscation function  $O_1$  applied to the object resulting from application of Character definition  $C_1$  applied to data object that is the simple concatenation of data objects  $D_1$  and  $D_2$ :

$$O_1(C_1(D_1+D_2))$$

Endorsement function  $E_1$  applied to the object resulting from application of Character definition  $C_1$  applied to Data format  $D_1$ :

$$E_1(C_1(D_1+D_2))$$

The process can be arbitrarily complex, for example:  $E_2(O_4(E_1(C_5(D_1+D_2))+O_5(E_1(C_5(D_4))))$  would be a perfectly legal workflow process. Why is this important? By nesting the functional representations and encapsulating them you yield a flexible and provable IA syntax. Further, if process  $E_2$  becomes inadequate to protect a process the protocol can be changed by replacing  $E_2$  with  $E_3$ , let's say. This is a type of stack-wise processing that allows component substitution without affecting any other components in the protocol. The protocols then become a composite representation of sub-functions and data segments that are independent of each other. It is important to note that we are not taking a position on how the encoding of functions should occur but all data required for the eccentric implementation of the function must be contained within the functional section not nested within the object. The function may, of course use object values for reference or arguments in the course of evaluating the function and use standard character data for decisioning within the workflow logic. An example of an eccentric piece of data would be the typing of the key structure to be used or a pointer to a virtual community boundary setting outside the object itself in normal class or subroutine logic these are local variables. Existing protocols and encodings are too complex and insufficiently adaptable in real time to emerging requirements. This can also be viewed as wrapping and unwrapping data in a flexible IA context based on particular requirements of the systems. The only issue is that the output object of a compound workflow may have visibility or reversibility issues. For instance a poorly designed composite protocol may obfuscate character information that is required at a later point in the protocol. It is important that that character data be moved into a segment that is not obscured to the steps that need it. This is an important step away from the current trend of developing hugely complex multipart XML objects that become stovepipes for particular purposes. It also speaks to needing a convention for concatenating and creating hierarchical objects out of sub-objects that is reversible and maintains relationship information through encoding or position. This is not currently supported within the XML standard.

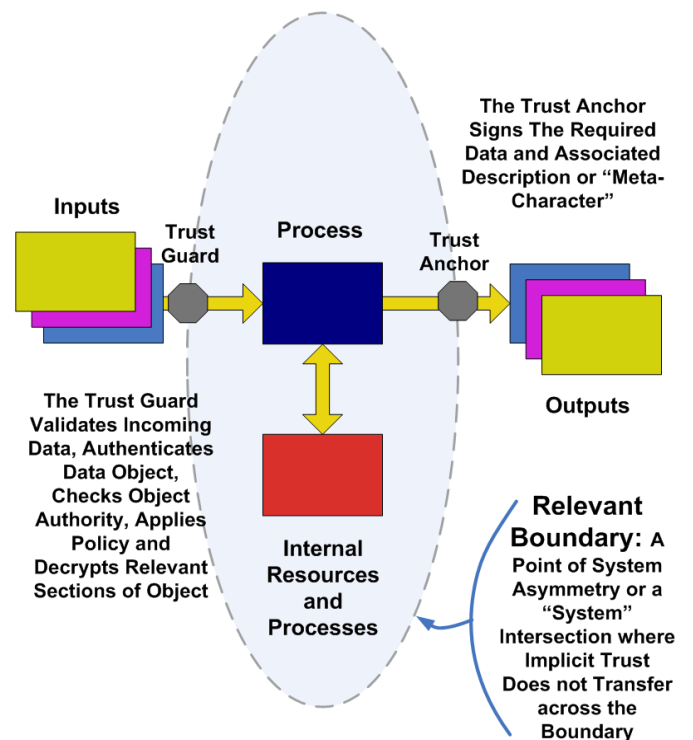
## Communities of Interest

First we must recognize that interconnected systems are organized into **Communities of Interest** that establish the rules of interaction and **Trust** symmetries. Policy and procedure are the only mechanisms for controlling this at this time but auto discovery and development enforceable dynamic rules are key long term development objectives. This is not the same concept as traditional Department of Defense Communities but a collection of Participants that agree to adhere to the same, well understood models for implementing IA, where the recent President's cyber security guidance mandated that we collectively engage in secure relationships with the rest of the world beyond the Federal space (NGOs, local governments, private companies, etc) which *cannot be done* using current IA&A / access control methods. The use of this concept allows the systems to determine the appropriateness of objects traversing the networks and understand the relationship between participants. As envisioned there would be no limit to the number of supported communities. The importance of this differentiation is that data flows can be separated by conformance criteria, whatever they are, allowing effective testing of VIA against the specifications of the community.

## Object Base Trust Enforcement

Second we must implement a rigorous analysis of **Trust Boundaries** and provide the means of protecting systems from improperly formed or **Authorized Objects** as in the referenced Basic Process Model above. This will require the development of new **Trust Guards** and **Trust Anchors**. But just as important, a new approach to systems architecture must be taken where the risks of attack and penetration are analyzed within each relevant **Trust Boundary** and these risks are used by a community to determine what, if anything, **Participants** are allowed to do across the **Trust Boundary**. And a new concept of risk mediation must be incorporated into the **Trust Guards** to mediate behavior not just action or **Authority**. For example, policies should have support for embedded concepts such as area of jurisdiction, single or limited use, chained authorization and other dynamic limitations that are often not part of the systems we use today. These dynamic limitations are a critical aspect of damage or scope containment. An example is the limitation of the number of inquiries by individual user on the system when such limits do not apply within the **Trust Boundary**.

### Basic Process Model



## Common Syntactical Model

The next important step is the development of a simple syntactical model for binding **Trust**, data, methods, state, **Character**. **Authorization** and relevant process steps together so that the outcome of a complex **Workflow** can be **Trusted**. The key recommendation is that **Character** and data be separated and bound cryptographically via digital signatures or encryption functions. Further, security processes should be explicitly separated from the **Character**, data and methods implemented by an **Object**. A basic notional conceptualization would be as shown below in an XML like hierarchical form:

### Object

Identity (Community URI:Participant)

Community Pointer (Community URI:Community Type)

TOB (Date/Time)

TTL (Expired Time)

Data Region

Label

Type

Character

Payload

Data Region

.  
. .  
. . .

Payload Digital Signature

Signature Label

Method of Digital Signature

Data Region Signed

Section of Labeled Data Region Signed

Authority Token (Community URI:Community User, User Public Key, Key Type, TOB, TTL, Token ID, Authority Character)signed by Community

Community Authority Token(Governance Authority URI, Community Public Key, Key Type, TOB, TTL, Token ID, Authority Character)signed by Governance Authority

Endorsement Character

Hash of Data Region with Endorsement Character Appended

Signature of Hash using private key associated with Authority Token Public Key

Payload Encryption

Method of Encryption

Data Region Encrypted

Key Pointer

Cyphertext

Signature Endorsement

Reference Signature label

Method of Digital Signature

**Authority Token (Community URI:Community User, User Public Key, Key Type, TOB, TTL, Token ID, Authority Character)signed by Community**

**Community Authority Token(Governance Authority URI, Community Public Key, Key Type, TOB, TTL, Token ID, Authority Character)signed by Governance Authority**

**Endorsement Character**

**Hash of entire referenced signature region with Endorsement Character Appended**

**Signature of Hash using private key associated with Authority Token Public Key**

This example shows a method of securing a data structure through the use of digital signatures, endorsements and cryptography. It is not intended to be a complete framework for this type of application but an example of the way this framework allows the flexible creation and modification of data objects as they move through the Trusted environment. Instead of using inflexible complex profiles this type of structure allows simple typing and compositing of these simple types into a complex data object. Further, signatures, character encoding and encryption methods are independent allowing the application framework to be unchanged in interacting with the data object regardless of the addition a new endorsements, signatures or encrypted regions. Character could easily be encoded as SAML or XACML assertions or in any format locally acceptable to the system architect. It also allows a stackwise handling of the trusted object rendering the encryption, signature and endorsement planes independently. This approach isolates edge applications from the need to incorporate security dependencies into their processing code and allows the mixing of different cryptographic and character encoding methods suitable for different purposes without affecting the underlying data segments or their format. In this example it is assumed that encryption puts a placeholder in the data structure when a cyphertext region is created, it is done this way to avoid changing the path logic of the core data object when encryption is applied but this is not a requirement, recursive logic could be embedded to handle the interpretation of the encrypted segment differently. Again, it is not intended to offer a solution, per se, but to structure a new approach to handling secure transmission of data in an information sharing or net centric integrated environment. It should also be noted that this is not an XML structure but uses an XML like hierarchy. This approach would allow direct internal referencing within the object and nesting of data structures, both capabilities are beyond current XML specifications. A verbose example of this would be (specified authenticity, disclosure protection, and authority process for bound object( (specified authenticity, disclosure protection, and authority process for inbound object((inbound object, inbound object character)), (specified authenticity, disclosure protection, and authority process for authorization purposes for participant(participant character), (specified authenticity, disclosure protection, and authority process for the methods implemented(participant method), (specified authenticity and authority process for the participant output object type(participant output))). These steps can be greatly simplified depending on the degree of Trust the community has in the methods implemented by the participant and the input received by the Participant. The process description can also be simplified across Trust Boundaries where the Trust asymmetry is limited. Further, the tagging described above can be constructed from chained endorsements of Character. An example of an emerging system using this approach is the ZBAC authorization system for access control. Another example would be the separation of control plane and routing infrastructure from the general networking traffic to implement a trusted computing

environment. It is important to note that is impossible to remove implied trust from current systems, even this syntactical representation has several assumptions built into the degree to which the participant is representing the actions taken internal to its trust boundary

This framework achieves a number of objectives:

1. Flexibility to change portions of the system as requirements and technologies change without having to redo all steps in a workflow,
2. Forces the designer to be rigorous about implied versus explicit Trust,
3. Provides the means to actually build and implement Trust models in complex information sharing environments,
4. Provides for the possible reduction of the scope of influence of a compromised system based on character and behaviorally mediated policy,
5. Sets up a new approach to protecting applications at the object level,
6. Allows safe tunneling of objects through unsafe pathways, and
7. Explicitly shows the designer where cross domain problems can occur.

Without this type of framework secure information sharing will continue to be relegated to stovepipe types of integration and will use highly inflexible domain guards. In contrast, this framework provides the ability to have an extremely powerful and flexible trust framework where participating entities have the ability to make cryptographically protected statements about the character of the data or themselves, policy can then be applied at relevant trust boundaries to see if the object is trusted. The use of a multi-level design in the object wrapper allows the architect to cryptographically separate participating entities based on their membership in communities of interest and the assignment of required authority to the particular entity. The use of the object based or “data centric” approach to security also protects data at rest and from interception to the degree the systems architect implements such protections. And by prepositioning chained authorization tokens it is possible to perform all adjudication and cryptography locally, dramatically lowering network overhead, remote server dependencies and the complexity of cryptographic key management.

Conclusions:

The web services and Net Centric communities in a headlong rush to leverage these new technologies have inadvertently made the Information Assurance problems across the Internet much worse. This arises from:

1. more accessible and standardized interfaces,
2. emerging complexity of composite applications, service reuse, and information sharing across traditional application boundaries,
3. significantly larger and more dynamic user populations with increasingly sophisticated requirements for policy driven control of relationships between systems and people,
4. use of systems with “security enabled and standards based” COTS software, which is necessary, but not sufficient – as the critical aspects are the settings and effective implementation of the minimal IA controls and capabilities – as well as common profiles.

The result has been ineffective IA protections in these new environments, a decrease in flexibility at the application level to address new requirements, and ever spiraling increase in systems complexity from an implementation, management and enhancement perspective. This paper lines out a new approach to architectural encapsulation of identity management, access control, authorization, and policy or rule set adjudication using the Validate-Integrity-Authority stack-wise framework at all relevant system intersections along with a new concept of virtual community boundaries.

It is imperative that the Department of Defense undertake the development of proof of concept and early standardization efforts of this protection framework to work toward actual field hardening and deployment within an 18 to 36 month timeframe. Efforts to utilize Net Centric systems and web services have put our capabilities at risk but this framework offers an answer to the problem, with minimal development effort, costs and lag time. The risk of not revising our IA approach and moving forward as expeditiously as possible is not acceptable, as we collectively continue to put our systems and data and people at much greater risk, which is growing exponentially. Success will also help the commercial sector secure its infrastructure with a common broadly deployed IA solution architecture

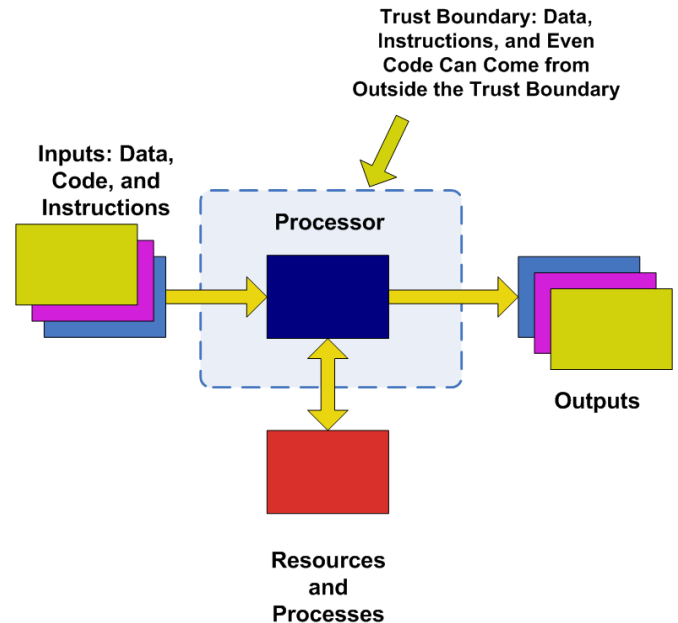
## Appendix A: History of efforts:

This architectural approach to securing the new Net Centric, Web Services and information sharing environments directly arises out of over a decade of work by the NuParadigm Companies in conjunction with DARPA, NSA, Navy, DISA, OSD, OSD-NII, Mitre, Sandia National Laboratories, DHS, Hewlett Packard, Booz Allen Hamilton, General Dynamics, QINETIQ, Northrop Grumman and SpaWar Systems Centers: NOLA, Charleston and San Diego. First efforts started with the program that became the HLS/HLD C2 ACTD and GIG IA SPO to grapple with early information sharing and access control problems as Net Centricity was being refined as a concept culminating in the theoretical design of RAdAC and the implementation of operational IMPP and HOLS alert frameworks. Additional efforts were expended on the design of the original IPAWS framework for DHS and two OSD and Navy sponsored SBIRS's which have resulted in the SecNav sponsored SIDI phase 3 program moving toward operational deployment. In a look towards the comprehensive problem set there have been a number of white paper contributions including content in the Hewlett Packard, SpaWar and NuParadigm paper comparing existing access control methodologies to a new model for access control known as authorization Based Access Control or ZBAC. These efforts build on a broader comprehensive effort to rationalize what is currently being undertaken in the IA universe with contributions to the SpaWar papers to W2COG and a general discussion IA Priorities within the services. These efforts have also built on original efforts by Hewlett Packard demonstrated in their papers on transitive trust and the groundbreaking work covered in their Zebra Copy paper. All of the papers are available upon request from the author of this paper at [hhaury@nuparadigm.com](mailto:hhaury@nuparadigm.com)

## Appendix B: Actual Implementation

So how do we put this together in a trusted framework. It is best to think of the problem in the context of the basic process model again. In this model a process is encapsulated at the trust boundary, it does not matter what happens inside the trust boundary from the perspective of the input source, just that the expected process is performed. And the outputs of the processor are just the inputs of the next process if there are multiple processes. It is important to understand that processes can be nested within other processes and when they are the composite of the nested actions is called an orchestration. When there is a defined series of steps to complete all the actions in a system but the calling entity for one step does not control the downstream steps in any way and the actions in each step are fully encapsulated, the composite of such serial and potentially parallel process steps is called a choreography. This does not mean that the designer is not going to constrain the objects and their behavior across the system but it is important to realize that in this design pattern the responsibility for each piece of the system is fully distributed across many nodes, this has important implications to the applicable trust constructs. But in either model to get a trusted result from a trusted input what is required to be understood?

### Evolving Basic Processing Model



1. That the content of the input object is valid, this infers some sort of typing of the input data structure to infer or determine validity. Again trying to avoid the object profile wars we do not care what the structural format is we just care that a trust guard can validate it using a known and accredited process for validation.
2. The content should be authentic, this is either inferred from the connection or explicitly determined through the use of cryptography. The only requirements are a known encryption/signature algorithm, an accredited process for checking the signature or encryption process used for the cryptography, and a digital signature, hash or cyphertext object and related keys.
3. The content should be authorized, this is either inferred from the connection or explicitly conveyed to the processor. In our model this is coded as a cryptographic token that binds the character of the process implementing the object with a signature determined to be authentic in step 2 above and the authority to implement the process on the type of input delivered to the system.

If this information was embedded in an XML like header it might have a structure like this:

- Object ID (Universal Owner Pointer/OID/unique ID)**
- Owner ([URL:entity](#)) optional**
- Type**
  - Process Type (Universal Type Pointer: Domain Owner URL/Process type/typeID) optional**
  - Content Type (Universal Type Pointer: Domain Owner URL/Content type/typeID) optional**
- Authenticity Type (Universal Type Pointer: Domain Owner URL/Authentic type/typeID) optional**
- Authority Type (Universal Type Pointer: Domain Owner URL/Authority type/typeID) optional**
- Time of Birth (Date/Time)**
- TTL (Time To Live Interval)**
- Object Content**
  - Primary Meta-data region**
    - Meta-type (Universal Type Pointer: Domain Owner URL/Meta-data type/typeID)**
    - Meta-data ((Arbitrary content conforming to structure- Meta-data type)**
  - Primary Content Region**
    - Process-Type (Universal Type Pointer: Domain Owner URL/Process type/typeID) optional**
    - Process-data (Arbitrary content conforming to structure- Process type, as applicable)**
    - Content-Type (Universal Type Pointer: Domain Owner URL/Content type/typeID) optional**
    - Content-data (Arbitrary content conforming to structure- Process type, as applicable)**
  - Primary Authenticity Region**
    - Authenticity-type (Universal Type Pointer: Domain Owner URL/Authentic-data type/typeID)**
    - Authentic-data ((Arbitrary content conforming to structure- Authentic-data type)**
  - Primary Authority Region**
    - Authority-type (Universal Type Pointer: Domain Owner URL/Authority-data type/typeID)**
    - Authority-data ((Arbitrary content conforming to structure- Authority-data type)**
- Object Level Authenticity Region**
  - Authenticity-type (Universal Type Pointer: Domain Owner URL/Authentic-data type/typeID)**
  - Authentic-data ((Arbitrary content conforming to structure- Authentic-data type)**
- Object Level Authority Region**
  - Authority-type (Universal Type Pointer: Domain Owner URL/Authority-data type/typeID)**
  - Authority-data ((Arbitrary content conforming to structure- Authority-data type)**

The content found in each region is the content required to resolve the validity, authenticity and authority of the object or a region of the object. Please note this does not include data centric protections from date disclosure. It is important to note that this is a completely pluggable and extensible framework, the structure of the content regions is up to the designers. An example Secondary Content Region for Authenticity might look like:

- Type (Universal Type Pointer: Domain Owner URL/Content type/typeID)**
- Key Pointer (Pointer to public key - Universal Type Pointer: Domain Owner URL/Key pointer/url:typeID or token name used to source public key internal to object)**
- Hash Type (Universal Type Pointer: Domain Owner URL/HASH type/typeID)**
- Region Signed (Internal Pointer: Starting tag-Ending Tag)**
- Hash Content (optional hash results)**
- Digital Signature (content of digital signature)**

It is important to note that the design and implementation of these elements is completely arbitrary and up to the designer to implement but the reference model allows standard typing and validation or execution modules to be assembled in a composable way across distributed networks. It is important to note that as a hierarchy of relationships is built within this new framework, interoperability will be

assured by implementation of standardized means of implementing functions, up to and including loading relevant code. The model also supports the delivery of trusted content such as programming code to be executed or pseudo-code to drive orchestration processes such as workflows. The process descriptor or typing in the authority or authentication realm allow the systems connected to resolve the format of the object, appropriately interpret its meaning and process the object. It is anticipated that typed objects will be signed by the authority that owns them which allows the consuming entity to verify the integrity of the reference object whether it is data, methods instructions or code. Nesting occurs when an entire object is contained within the content section of the higher level object. Object composites occur when there are multiple object content regions within the object, note the structure of each peer may be an independent type. When there is a hierarchical relationship, for instance authenticity and validity applying to multiple content regions partial nesting should be implemented which would look like this:

```

Object Content
Object Content
  Primary Meta-data region
    Meta-type (Universal Type Pointer: Domain Owner URL/Meta-data type/typeID)
    Meta-data ((Arbitrary content conforming to structure- Meta-data type)
  Primary Meta-data region
    Meta-type (Universal Type Pointer: Domain Owner URL/Meta-data type/typeID)
    Meta-data ((Arbitrary content conforming to structure- Meta-data type)
  Primary Meta-data region
    Meta-type (Universal Type Pointer: Domain Owner URL/Meta-data type/typeID)
    Meta-data ((Arbitrary content conforming to structure- Meta-data type)
  Primary Content Region
    Process-Type (Universal Type Pointer: Domain Owner URL/Process type/typeID) optional
    Process-data (Arbitrary content conforming to structure- Process type, as applicable)
    Content-Type (Universal Type Pointer: Domain Owner URL/Content type/typeID) optional
    Content-data (Arbitrary content conforming to structure- Process type, as applicable)
  Primary Authenticity Region
    Authenticity-type (Universal Type Pointer: Domain Owner URL/Authentic-data type/typeID)
    Authentic-data ((Arbitrary content conforming to structure- Authentic-data type)
  Primary Authority Region
    Authority-type (Universal Type Pointer: Domain Owner URL/Authority-data type/typeID)
    Authority-data ((Arbitrary content conforming to structure- Authority-data type)

```

The next important concept within this framework is endorsement; endorsement is the asymmetric execution of an authentication process by a third party not originally part of the transaction. Endorsements can be simple co-signing of an object and would look like:

```

Type (Universal Type Pointer: Domain Owner URL/Content type/typeID)
Key Pointer (Pointer to public key - Universal Type Pointer: Domain Owner URL/Key pointer/url:typeID or token name
used to source public key internal to object)
Hash Type (Universal Type Pointer: Domain Owner URL/HASH type/typeID)
Sig Type (Universal Type Pointer: Domain Owner URL/Sig type/typeID)
Region Signed (Internal Pointer: Starting tag-Ending Tag)
Hash Content (optional hash results)
Digital Signature (content of digital signature)
Endorsement
  Type (Universal Type Pointer: Domain Owner URL/Content type/typeID)

```

**Key Pointer (Pointer to public key - Universal Type Pointer: Domain Owner URL/Key pointer/url:typeID or token name used to source public key internal to object)**  
**Sig Type (Universal Type Pointer: Domain Owner URL/Sig type/typeID)**  
**Digital Signature (content of digital signature)**

Note the endorsement does not require a hash or hash type if it is signing the same hash, if it is using a different algorithm these elements will be needed.

This structure supports complex distributed processing and the assembly of trusted components where trust must transfer across boundaries and supports arbitrarily complex applications and the intersection of highly eccentric systems. It is important to note that by using proxies at the edges, content objects can be delivered into existing applications without having to modify any application code allowing the use of this trust framework for both new and legacy applications.

## Appendix C: The risk to our country

"The Munk Centre for International Studies at the University of Toronto today released [a research report](#) based on 10 months of investigating what it calls "GhostNet," a cyberespionage operation that has netted stolen documents and gained full control of some of the systems it has breached. GhostNet has infected nearly 1,300 computers in 103 countries -- mostly in Asia, but also in Europe as well as a NATO system. Around 30 percent of the infected machines were "high-value" targets, including foreign affairs ministries in Iran, Bangladesh, and Latvia; embassies for India, South Korea, Indonesia, Romania, Thailand, Taiwan, Germany, and Pakistan; as well as the Asian Development Bank and several news organizations."

-Dark Reading, March 30, 2009

A new report from the Homeland Security Department's U.S. Computer Emergency Readiness Team (US-CERT) adds even more fuel to the fire. The report listed 18,050 cybersecurity incidents in agencies in fiscal 2008, compared to 5,144 in fiscal 2006.

Agencies have reported a steadily increasing number of incidents since 2006, partially because hackers have greater access to malicious software they can use to attack and partially because agencies have improved their incident detection and reporting, said Mischel Kwon, US-CERT director.

FCW, February 23, 2009

Cyberspies have penetrated the U.S. electrical grid and left behind software programs that could be used to disrupt the system, according to current and former national-security officials.

- Wall Street Journal on-line April 8, 2009

"More than 27 million Americans have been victims of identity theft in the last five years.... To deal with the problem, consumers reported nearly \$5 billion in out-of-pocket expenses."

-*The New York Times*

The Pentagon spent more than \$100 million in the last six months responding to and repairing damage from cyber attacks and other computer network problems

-AP April 7, 2009

"This year alone more than 500,000 Americans will be robbed of their identities...with more than \$4 billion stolen in their names."

-*CBSnews.com*

"According to Government Executive, after the intrusion was discovered and the network shut down, it took OSD three weeks, \$4 million, and the introduction of a boatload of new security processes before recovery was complete. The US Department of Defense gets some 70,000 intrusion attempts per day" ...

-*IEEE, Spectrum Online, March 28, 2008*

"A recent report on identity theft warned that there is likely to be 'mass victimization' of consumers within the next two years. The report said consumers should be extra careful to monitor all their financial transactions for unexplained account activity, withdrawals, or fund transfers."

-*The Gartner Group, a technology research group*

"Cyberwarfare is already here.... It's one of our major challenges," said Defense Deputy Secretary Gordon England on Monday at the annual National Community Service and Legislative Conference of the Veterans of Foreign Wars."

“During a Senate Armed Services Committee hearing last week, Sen. John Thune, D-S.D., asked National Intelligence Director Michael McConnell if the United States was prepared to deal with threats against military and civil networks and information systems. ‘We’re not prepared to deal with it,’ said McConnell, identifying both China and Russia as adversaries who are attempting to penetrate U.S. information systems.”

*-Government Executive.com March 2008*

“In some government sectors money is no longer spent on what is known as “Red Teaming” or attack simulation because there is no point in the view of the developers and designers. It is a known fact that an insider can bring any sophisticated Net Centric computer system to its knees in seconds and the patient outside attacker with sufficient resources can position themselves to bring it down in minutes to hours.

*-anonymous government official, October 2008*

As the use of the Internet, web services, and Net Centric systems continues to explode the vulnerability of our systems to attack by our adversaries continues to get worse and worse. The threats to our country posed by the inappropriate trust of systems, networks and the Internet are real and the risks are growing in number, potential impact and attacks are climbing in frequency every day. The weaknesses of the current systems and “known” exploits have grown to the point of near absurdity. There is no viable option but to deal directly with the issues as losses continue to mount and the sophistication of attacks continues to grow. In the course of any discussion it is impossible to cover every vulnerability due to the astronomical number of system permutations. But it is useful to discuss what the significance might be to a knockdown level attack on our improperly protected systems. Risks can be broadly categorized into the four attacker objectives with our current systems:

1. The adversary aims to “exfiltrate” or steal important information, this can range from personal medical information, passwords and credit card numbers to state secrets. If it is on a network or computer it is a potential target. The purpose of the adversary can range from personal pride to state based warfare.
2. The perpetrator would like to forge information to allow them access to systems for the purposes of theft, sabotage, denial, or spying.
3. Capture or misappropriation of systems with the objectives ranging from free use of computer systems capabilities to setting up massive botnets to facilitate future denial of service attacks, and
4. Direct denial through the sabotage of critical network infrastructure or the flooding of networks with excessive amounts of traffic.

## The Escalating Risk

But what do these vulnerabilities mean to the country, the following scenarios are within the realm of capability of a number of our potential adversaries, let’s examine the possibilities:

1. **Christmas attack 20XX.** Our retail operations in a continuing effort to cut costs migrate more and more to the Internet. Some major corporations are one hundred percent dependent upon

Internet based sales but many average companies now rely on the Internet for 10 to 30 percent of their total annual sales. During the height of the Christmas buying season hackers employ a sophisticated distributed transaction attack automating the ordering of hundreds of millions of fraudulent transactions across the Internet. Credit card holders refuse to pay, retailers become embroiled in recriminations about responsibility and liability for tens of billions of dollars worth of product, credit card companies that have guaranteed Internet transactions for consumers go bankrupt, companies facing a complete failure of trust in the transactions across the Internet 'temporarily' stop using the Internet or have to massively retool their online security processes losing important sales and the consumers stop exposing personal data across the Internet because of fear about information being stolen further curtailing sales. Declining Internet sales, massive fraud losses, hesitant consumers and lack of remaining 'brick and mortar' capacity combine to bankrupt hundreds of companies ranging from Internet florists to EBAY and Amazon. Hitting at a time of weak initial recovery from an earlier recession consumer confidence implodes and the country spirals back down into an even deeper recession.

2. **Escalating tensions Winter 20XX.** After years of quietly building silent Botnets another world power reaches the limit of what it declares is the unlawful arrogant hegemony of the United States of America and it unleashes attacks on the Internet itself. Because of the money invested and the stealthy work over years of patient attack all major networks are flooded with traffic. This is timed to coincide with the attacker's strategic attack on several critical transcontinental and transoceanic fiber hubs with simple crews disguised as utility workers cutting a number of very important fiber optic cables. The nation's telephony and network infrastructure has been allowed to converge to the point that many commercial and government networks share critical bandwidth. Suddenly access to networks ends in many places, VOIP telephony is completely disrupted, critical local network loops are knocked out, Point of Sale terminals cannot connect to credit card clearing systems or integrated supply logistics systems. The economy comes to a screeching stop, critical infrastructure from SCADA systems to first responder systems collapse resulting in dangerous shutdowns of large portions of the energy grid during the dead of winter. Hundreds of thousands of persons are displaced to emergency shelters, the economy sputters and fails, thousands of citizens refusing to leave their homes die of exposure, and the economic losses from denial of use and indirect infrastructure damage are in the hundreds of billions of dollars. These new Botnets have been built to heal disruptions in their grid and trans-morph themselves. The Botnet extends itself and activates sleeper zombies faster than the infection can be eliminated. It takes months of effort to clean the systems up and restore normal functioning to the Internet and affected systems. Due to emergency segregation of the networks to restore some basic high priority capabilities, large segments of the Internet no longer have adequate capacity causing significant economic dislocation in the areas affected for an extended period of time.
3. **Military espionage.** Net Centricity continues to build with inadequate systems boundary protections allowing the networks to be instrumented with silent viruses, backdoors, sophisticated man in the middle proxies, and remote control software. The systems are being used to gather data about military operational capability, exfiltrate state secrets and position electronic warfare sentinels throughout our networks. Constant probing of our networks has

heightened our alert status but many successful hacks are undiscovered. Without explanation new capabilities start showing up at an alarming rate in the opposing military services and during a high profile military incursion red force and blue force tracking systems supporting command and control are suddenly compromised. Joint task list assignments target certain elements of coalition forces that are outside our direct command and control space so the compromise goes unnoticed for critical minutes while high altitude and standoff assets are brought to bear on friendly forces and our own 'covert' assets. There are thousands of friendly casualties including the loss of several capital ships of friendly navies and numerous friendly aircraft. In addition, the loss of confidence in the entire information/command chain results in the loss of our information enabled agility. Lengthened decision cycles and the as yet undetected active exfiltration of C4ISR data allows our own networks to become giant sensors for our adversary. Suddenly the advantage we had in our decision turning radius is lost to the opponent with all the resulting real world effects this will have on the battlefield.

The threats posed by these hypothetical scenarios are real but we tend to ignore things that are possible but have not happened yet. Whether we are talking about the Maginot line, Pearl Harbor or the 9/11 terrorist attack, history has shown us that the adversary does not choose to fight on our terms. The only thing standing between us and these types of attacks are the patience, resources, sophistication, dedication of our opponents and the will to use the weapon. It is **not** the state of our systems security. Would a determined well funded adversary have the technical means to pull these attacks off? To date, many attacks have been by criminal groups seeking personal profit or individuals seeking bragging rights. State sponsored activity is out there and increasing, but they are not tipping their hands anymore than necessary as they silently probe our computer systems. Where they successfully penetrate, without detection, we will not even be aware of the compromise of our systems.