

National Cyber Leap Year Summit 2009: Exploring Paths to New Cyber Security Paradigms Draft Report of Participants' Ideas

August 24, 2009

New Game: Attacks only work once if at all.

This document explores Moving-Target Defense as a path to this new game.

The following ideas were captured in unedited form at the National Cyber Leap Year Summit. The ideas are a summary of the discussion of the participants in the Moving-Target Defense session. They do not necessarily represent the opinions of the co-editors or the organizations they represent. The Summit is managed by QinetiQ North America at the request of the NITRD Program, Office of the Assistant Secretary of Defense Networks and Information Integration, and the White House Office of Science and Technology Policy.

Please **provide your comments**, if any, **by September 3, 2009** for utilization by the Summit's program co-chairs at <http://www.co-ment.net/text/1450/>. To add a comment, select the "Add" tab in the left navigation menu, select (highlight) the portion of the document you are commenting on, and provide your comment. If commenting on an entire section, you may select the section heading to anchor your comment.

If you have any further questions or comments, please visit the National Cyber Leap Year Web site at the following address: <http://www.nitrd.gov/NCLYSummit.aspx>, or send email to leapyear@nitrd.gov.

What is the new game?

In the current game, attackers win by taking advantage of the relatively static nature of our systems. For example, permanent, well known addresses, names, port numbers, etc. represent clearly identifiable parameters that turn vital servers and services into an easy target. Adversaries can plan at their leisure, relatively safe in the assumption that our key IT assets will look the same for a long time. They can map out our likely responses and stockpile a set of exploits that escalates in sophistication as we deploy better defenses. They can afford to invest significant resources in their attacks because they expect to persist in our systems for a long time. In the new game we win by increasing the randomness or decreasing the predictability of our systems. By making the cyber terrain appear chaotic to the adversary, we force him to do reconnaissance and launch exploits anew for every desired penetration; the attacker enjoys no amortization of development costs. The new game, in this context, consists of considering very dynamic rather

than static network architectures. In other words, the new game is about real-time distributed monitoring, control and diagnosis of very dynamic and flexible cyber environments.

1 Mutable Networks: Frequently Randomized Changing of Network Addresses and Responses

- Create Virtual Machines (VMs) that are rotated and exposed to the attacker only for a limited time
- Applicable for short transactions
- Restart with different Operating System
- Concerns
 - Virtualization performance
 - Total cost of ownership
 - Fixed patterns of management
 - Difficult to do root cause analysis because the Intrusion Detection System (IDS)/Intrusion Protection System (IPS) does not work
- Paths to This Change
 - Round robin address movement
 - Frequency-hopping analogies
 - Approaches that are unpredictable or not necessarily random to attackers,
 - Redundancy, recovery, fast switching
 - Deployment on new architectures, e.g., the smart grid
 - Tunnels, for hidden services
 - Building on Content Content Delivery Network (CDN)
 - Deployment on an overlay
- Derailers
 - Lack of demonstrated scalability
 - Lack of Internet Protocol Version 6 (IPv6) adoption because uses large address space
 - Architectural invariants, if any
 - Usability impact on systems

1.1 Description

A prerequisite for building successful cyber defense systems is to investigate effective countermeasures to scanning and reconnaissance attacks that allow for discovering network resources end-addresses and system finger print. Scanning and reconnaissance attacks are precursory steps to launching devastating attacks such as system penetration or denial of service. The objective of this project is to provide the ability to dynamically change the external host interfaces such as names, IP addresses, and port numbers. Also, the external response behavior should be randomly changed to counter scanning worms, and reconnaissance and finger printing attacks. These changes are accomplished by continuously outdating the collected system information within a short time window, and deceiving attackers to fake targets for further analysis.

In this proposed approach, networked systems (i.e., end-hosts) will be assigned different addresses frequently based on random functions such as hash tables. One approach is to select interfaces using the randomized round robin technique. The change has to be done:

- On a high frequency basis to outperform automated scanner and worm propagation
- Quickly to minimize service disruption and delays
- Unpredictable to ensure that future IP addresses and keys are undiscoverable and irreversible (i.e., high entropy distribution)
- Operationally safe to preserve system requirements and service dependencies.

Redundancy can be added to this scheme using Virtual Machines to support recovery and diversity to the attack profile surface.

We have two mechanisms to randomize external system responses:

- First, as a short-term approach, session control responses such as Transmission Control Protocol (TCP) 3-way handshake, in network applications, will be intercepted and modified to give a false finger print identification in order to deceive and analyze the reconnaissance adversaries. However, in the long-term, it will be advantageous to have camouflaging capabilities integrated in the system session control.
- Second, firewalls will also deceive scanners by generating positive responses for all denies packets. The combination of these two techniques will give an effective motion target approach for countermeasure reconnaissance attacks.

1.2 Inertia

- Requires instantaneous update of network routing tables and security policies
- Scalability: How can such activity be done in a timely fashion for large networks?
- Lack of theoretical foundations to model and analyze network configurations.
- Lack of efficient distributed configuration management that can orchestrate such dynamic changes without causing inconsistency and access or availability problems.
- Lack of efficient network proxies and indirection technologies.
- Lack of adoption of IPv6 to maximize IP addresses hopping.
- Lack of efficient and scalable address translator Network Address Translation (NAT).
- Not capable of supporting multiple interfaces in MAC and IP level.
- Lack of techniques to manage session and network perturbation as a result of dynamic changes such as service interruption due to mis-synchronization, and mis-configuration.
- Requires maintenance of service dependency and system invariant.
- Impact and overhead on operational system functionality, reliability and performance.

1.3 Progress

- Availability of efficient and widely accepted virtualization configuration
- Ability of high-speed networks with rapid update capabilities.
- Multi-switching hardware

- Recent improvement in computation including desktop, module checker, Boolean Satisfiability (SAT) solvers
- Better understanding of attacker tactics

1.4 Action Plan

Note: Must add specific actions - currently just listed technical issues]

- Leverage hashing technology, develop a function to generate network interface randomly considering the time, a shared secret key and service dependency
- Modify network protocols to support multiple simultaneous interfaces at the end hosts during the transient changing period.
- Implement a distributed controller to coordinate the dynamic allocation and distribution of network address
- Implement rapid hot-swapping for router and host configuration changes
- Use OS and/or Kernel/Chip level direct reconfigurable address and translation tables
- Use software level retranslation for port connection
- Integrate this technique in Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP) to support dynamic address-hopping technique

1.5 Jump-Start Plan

1.5.1 Technical Plan

- Use a simplified approach to implement the basic components of the system including pseudo random function, and centralized management controller.
- Leverage open source OS such as Linux to make the necessary changing in the protocol stack to make IP tolerant to address-switching transient delay.
- Using diversity of VMs to simulate different system responses (fake finger printing) and create a false identity.
- Building proxies for address translations and redirection.
- Use open source virtual router implementation to demonstration configuration hot-swapping.

1.5.2 Experimentation Plan

- Identify testbed demonstration opportunities and demonstrate relevant capabilities using research networks (eg: Defense Research Engineering Network (DREN), DETER etc). DETER is a testbed for network security projects.
- The following use case studies will be implemented:
 - Use these test beds to implement the basic components of dynamic address motion and evaluate the effectiveness of this approach against random scanning and divide-and-conquer worms. The objective is to demonstrate the effectiveness of this approach to significantly slow down worm propagation by increasing uncertainty in scanning phase. We will also solicit real worm traces from companies like Symantec and Cooperative Association for Internet Data Analysis (CAIDA) repositories.

- Test the finger printing and firewall deceiving techniques against automated scanning tools like Network Mapper (Nmap) and Nessus, a network scanner tool, as well as using real scan traces from Semantic.

1.5.3 Team Collaboration and Bootstrapping

- Approach and engage potential collaborators from configuration management, network device vendors, ISPs and security operations and management industries through a series of talks and panel discussion during an invited 1-day workshop. We identified the following potential collaborators based on their relevance to the projects:
 - RedHat for using Linux in our short-term case study
 - Telcordia for automatic synthesis and verification of network configurations
 - Cisco for the network virtualization and hot-swap configuration capabilities
 - VMWare for integrating finger printing deception mechanism in the virtual machines
 - Symantec for test and evaluation using real scanning traces. We will also deploy this on a real operational network with collaboration with AT&T.
- Engaging government agencies such as the National Security Agency (NSA) and Army Research Office (ARO) / Army Research Laboratory (ARL) to evaluate the potential of this idea on mission critical networks.

1.5.4 Case Study

- Use an identified testbed (e.g., DREN or DETER) to evaluate the effectiveness of this approach against random scanning and divide-and-conquer worms. The objective is to demonstrate the effectiveness of this approach to significantly slow down worm propagation by increasing uncertainty in scanning phase. We will also solicit real worm traces from companies like Symantec and CAIDA repositories.
- Test the finger printing and firewall deceiving techniques against automated scanning tools like Nmap and Nessus tools as well as using real scan traces from Semantic.

2 Diversity in Software

2.1 Description

Currently, we live in a software monoculture - most computers run essentially the same software. This makes it easy for an attacker because the same attack vector is likely to succeed on most computers. If we make every computer run a subtly different version of the same software, a different attack vector is needed for different computers. From the perspective of the end-user, all the different versions behave in exactly the same manner, but they implement their functionality in subtly different ways.

As a result, any specific attack will succeed only on a small fraction of systems and will no longer sweep through the whole internet. An attacker would require a large number of different attacks and would need to target the specific software versions that are susceptible to each specific attack, which radically increases the cost to the attacker. The effective penalty to the attacker is the inability to amortize knowledge over a series of attacks. - each attempt is distinct from any previous attempt or attempts. If multiple versions of the same software are run in parallel on a single computer, attacks could be detected in real-time when the behaviors of the versions diverge as the result of an attack that is successful on only some of the versions, but not on others.

2.2 Inertia

Until now, software was predominantly shipped "in boxes on a CD". Mass production of the CDs made it impractical to give every user a different version. But we are rapidly transitioning to software distribution over the network, where this is no longer a concern.

There is a cost associated with creating diversity. Until now, people have been oblivious to the risks and have not embraced the idea of paying for security. The tradeoff between security vs. performance is only now becoming better understood by a wider audience.

Until now, we have focused on creating the "best" version, e.g., in compiler optimizations. Only one of the versions can be the "best". So if we give a different version to every user, by definition, not everyone can have the "best" version. So there is a performance cost associated with this solution. There is an additional intrinsic cost of diversity - configuration management, centralized administration, etc. might become more onerous.

Security has in the past focused on "predictability" and testing. The idea of running completely different code on each individual computer requires a radical shift in thinking and culture and certification and accreditation, because, by definition, one can no longer test all of the versions, but one is required to trust the compiler.

Understanding the complexities of software and hardware dependencies among linked/embedded applications is not well preserved

2.3 Progress

Distribution of a different program version to each and every customer becomes feasible when software is downloaded via the network rather than installed from a CD. We have just arrived at the point when many programs are now routinely installed only from the internet. For example, more than 400 million people have downloaded the Firefox browser.

Computers now have such high performance that paying a small performance overhead such as 5%-10%, for the extra security brought about by diversity, may be worth the cost in many contexts.

Compilers have advanced very significantly, so that automated generation of variants is now a reliable and predictable process. Even dynamic compilation is now routinely employed with very high reliability. For example, Apple has transitioned millions of users from the PowerPC to the Intel architecture using a fully automated just-in-time compiler without any reported incidents. The reliability of these compilers is stunning, considering that they have been able to automatically translate programs of the size of the Microsoft Office suite fully unattended, without any testing of the resulting output, and on-the fly.

Multi-core processors offering high degrees of parallelism (80 cores already announced by Intel) make it feasible to run several slightly different versions of just one program in parallel.

2.4 Action Plan

- Develop compilers which, instead of choosing the best path, preserve all legal alternative paths.
- Develop a software distribution engine that queues up different variants of a software program so that the first requester gets the first version, the second requester the second, etc. The system would continuously generate new versions to queue up at the same rate as requests are coming in. For small programs, versions could be generated on-the fly at the time of the request, but for larger programs (e.g., Firefox or the Apache server), such versions would be generated ahead of time and queued up for delivery.
- Develop n-version systems that execute multiple versions of the same software in parallel for added resilience against attacks.
- Develop randomization techniques that further increase the variability to an attacker without changing the functionality for the end-user.
- Develop inventory management database to track how versions are distributed and provisioned. In many cases, no inventory management may be necessary at all. For example, we don't really care which version of Firefox any given user has.
- Tackle the hardest problem Commercial Off-the-Shelf (COTS) or layered/embedded multiple COTS)

2.5 Jump-Start Plan

Pick an existing open-source project (Firefox, Apache) with documented past vulnerabilities. Modify the compiler used in its build process to generate many functionally equivalent versions simultaneously. Run old software versions with known vulnerabilities through the diversity mechanism and measure which proportion of attacks no longer succeed on the diversified code base.

3 Robust Random Authentication

3.1 Description

Tests to authenticate someone vary dynamically (at different points)

3.1.1 Concerns

- Usability, user acceptance
- Finite number of mechanisms
- Difficulty in delegating
- Take a small number

3.1.2 Mitigation

- Deploy ubiquitous Public Key Infrastructure (PKI). There are examples where this has been deployed
- Could be designed to make it easy
- Could use a fingerprint stored in Trusted Platform Module (TPM). This eliminates passwords and other weak forms of authentication.
- Provide diversity in end-user authentication for both human users, smart devices (sensors), and application software connections in manner, timing and channel. Apply a combination of multiple biometrics (e.g. face, voice, keystroke), multiple tokens (e.g. PC/phone signature, multi key fobs), and over multiple channels (e.g. web, email, voice, text) to authenticate not only at a defined log-on, but possibly during the session for validation. For the applications layer, use analogous continuous authentication (e.g. a low detectable, frequent challenge/response protocol possibly via keystroke, facial).

3.1.3 Benefits

- Raises the bar for any attacker attempting to steal a user's credentials, authorizations, or impersonate a user's identity by requiring the attacker to steal, counterfeit or spoof stronger credentials (not just user password and out-of-wallet information). Also the attacker must time this, not only at log on time, but over the entire user session at unpredictable times and over multiple channels
- Increases privacy by reducing the spread of Personally Identifiable Information (PII) across multiple websites, as the user can be authenticated by a federated authentication; make possession of PII insufficient to gain control over a user's accounts or to be able to impersonate the user over the Internet because stronger credentials, such as biometrics, are required in addition to knowledge of PII, to be authenticated.

3.2 Inertia

- User acceptance and historical precedence;
- Early immaturity (performance and cost) of biometrics
- Early cost and inconvenience of tokens (necklace problem - by necklace problem we mean that the early implementation of this approach required each website to provide their own token/credential, such as a One Time Password (OTP) token, so the user needed a growing number of tokens/credentials - one per website)

- When the Internet first got commercialized, there was not sufficient commerce to attract organized crime and it was not a sufficiently big problem to require more than ID and password over Secure Sockets Layer (SSL)
- Need for mutual authentication and ability to address man-in-the-middle, man-in-the-browser attacks
- Vulnerability in the initial registration/credentialing process
- Scalability to work with 10s of millions of users over 10s of thousands of sites

3.3 Progress

- Moore's law (decreased cost, increased capability) provides the necessary computational power for authentication devices at more affordable costs;
- Advances in biometrics - improvements in performance at lower cost;
- Advances in tokens and growing ubiquity of the smart phone making multiple channels, biometrics, device fingerprinting, geo-location all practical now;
- Changing attitudes as cyber crime has dramatically risen. User acceptance and demand for stronger authentication is growing, as well as greater acceptance of white-listing, along with coincident improvements in browser design – greater isolation between browser sessions;
- Growing willingness for key identity providers such as Government and Financial Services to cooperate in initial user identification

3.4 Action Plan

- Work with the smart phone companies and carriers to incorporate FI-issued credentials and required access methods
- Utilize the Federal Federated Identity Management Bridge authentication as a foundation to grow upon, as well as other popular Identification schemes (e.g. CardSpace, Open ID)
- Prototype and validate in a test bed using a smart phone, with browser either on PC, or on the phone itself, with strong Financial Service user registration, credential issuing and verification
- Demonstrate that the prototype satisfies user acceptance, privacy, security and liability concerns, and works in the face of defined threat and red team attacks

3.5 Jump-Start Plan

- Build upon current smart phone designs and Wireless Fidelity (WIFI) authentication infrastructure services
- Pick a few compelling high assurance applications (e.g. from Government, Finance, and Healthcare) with friendly users (e.g. customer employees) to pilot

3.5.1 Use Case

As part of this effort we would include a number of examples and test cases that can serve as explicit illustrations of how the pilot can be expanded and used by a larger audience. One test case could be to have three or more financial institutions, at least one non-financial company and at least one government agency cooperate to use interoperate medium Federal Institute of

Processing Standards (FIPS)/National Institute of Standards and Technology (NIST, Level 3) assurance credentials for login to multiple online sites. The scenario might include a member of another critical industry requiring high identity assurance, such as the healthcare industry. The scenario could also illustrate how authentication could be applied to smart devices such as power grid sensors.

4 Resilient Cryptographic Systems

Idea: Most cryptographic techniques, protocols and implementations today are brittle and vulnerable to catastrophic collapse of security due to a single point failure. This is in part because remote penetration, social engineering, insiders, supply chain modifications, and the age-old practice of bribery continue to provide successful means to bypass cryptography. Better cryptography, longer key lengths, algorithm composition, etc., do absolutely nothing to remediate these bypass vulnerabilities. The goal is develop a new generation of cryptographic systems that are resilient to multiple compromises. Although new cryptography can incorporate multiple hard mathematical problems, attention to the broader range of attack surfaces is necessary to staunch current hemorrhaging.

4.1 Description

Cryptographic systems can collapse due to failures in multiple dimensions, or attack surfaces, often beyond the crypto-analytic components. By making these dimensions impervious to single failures, attackers will face increased work factors. Below are listed dimensions of fragility together with approaches to improve resiliency.

4.1.1 Randomizer Failure

- Compensate with multiple random sources.
- Utilize external sources of randomness.
- Devise more resilient protocols to manage low entropy randomness.

4.1.2 Incorrect Implementations (Supply chain)

- Develop independent implementations and compare their outputs.
- Improve third party certification and accreditation.
- Incorporate real time test vectors to check cryptographic operations actively.

4.1.3 Secret Key Compromise

- Use techniques for split keys and distributing them to non-intersecting security domains.
- Develop techniques for key agility.
- Employ third party assistance in crypto computations (example. composite private keys).
- Deploy tamper resistant containers.

4.1.4 Side Channels and Covert Channels

- Develop useful models of information leakage and cryptographic computational methods resistant to such leakage.
- Devise techniques for reducing timing synchrony (consistent timings).
- Deploy techniques for power leveling.
- Implement techniques for obfuscating hardware cache behavior.
- Use encoded computation to maintain secrecy even in the presence of side channels leakage.

- Improve virtual machine separation at hardware and software level, to reduce threat of cross-VM key ex-filtration.
- Identify and construct minimal secure components from which larger secure computations can be built up

4.1.5 Software Bugs

- Write crypto code in safe abstraction-oriented programming languages designed for verifiability.
- Require verified compilers.
- Verify crypto code.

4.1.6 Hardware Failure

- Use active checking to assure correct numeric calculations.
- Design for minimizing catastrophic effects of faults, e.g., prevent "fault attacks", where a single bit flip causes a full key leak, as some current algorithms
- Use late binding logic, e.g., Field Programmable Gate Array (FPGAs), for crypto operations.
- Perform computations redundantly on separate processing units with strategically different supply chains.

4.1.7 Depot and Distribution Vulnerabilities

- Develop crypto systems using certified supply chains.
- Institute certified tracing and handling for crypto systems.
- Devise deployment mechanisms that enable rapid, or even dynamic, update of crypto algorithms or protocols.

4.1.8 Weak Standards

- Engage broader communities in design of standards (pre competition).
- Use NIST competitions to "red team" algorithms.

4.1.9 Loss of Physical Security

- Deploy anti-tampering techniques.
- Use volatile storage for keys.
- Develop techniques to reconstitute trust reactively in response to breach or proactively to assure system loyalty.

4.1.10 Novel Attacks

- Exploit mathematical leverage beyond factoring.
- Develop algorithms that resist quantum attacks.

4.2 Inertia

- System security has been the weakest link.
- The community is entrenched in private key trust model.
- Government resistance to widespread distribution of more robust cryptographic systems.

- Widespread deployment of current PKI models makes upgrading slow.
- Misplaced belief that strategies such as algorithm composition, diversity, and frequent updating will provide more security when, in fact, they primarily introduce unneeded complexity, signatures, expense, updates and licenses (multiple vendors).

4.3 Progress

- Vibrant academic cryptography community.
- New crypto models (e.g., elliptic curve cryptography, identity-based encryption, homomorphic encryption, leak-resistant crypto)
- New authentication schemes (e.g., multi-factor authentication, identity-based authentication, mutual authentication)
- Recent progress in verified compilers and verification of software and hardware.
- Specialized programming languages for crypto (e.g., Cryptol).
- Trusted Platform Module (TPM) and Trusted Computing (TC) effort.
- Greater integrated circuit capacity
- Weak system security renders more conventional crypto ineffective and creates a need.
- New computational platforms (mobile, cloud) and convergence pose new challenges for crypto.
- Considerable experience with deployed cryptographic systems.

4.4 Action Plan

- Fund research to develop more resilient cryptography and an advanced implementation tool chain.
- Fund research to develop wide-area collaboration systems to support design, development, implementation and management of cryptographic systems.
- Establish a program for teaching crypto to advanced high school students, including a summer math camp.
- Develop interoperable standards for resilient cryptographic systems across the vulnerability dimensions.
- Weave resilient crypto into the fabric of system and network architectures (synergistic protection).
- Adopt new standards for government use to prime commercial build out.
- Mandate use of more robust cryptography in areas requiring higher levels of assurance in the context of markets stratified by levels of information assurance necessary for safety and security.

4.5 Jump-Start Plan

- Hold workshops on:
 - Resilient cryptography to mobilize the technical community;
 - Verified adaptive programming languages for crypto;
 - Hardware architectures to support resilient crypto;

- Application needs for early adopting sectors.
- Announce a challenge competition for resilient crypto to engage a broad community in the development of new paradigms for resilient cryptographic systems.
- Jump start research via new funding on advanced programming languages designed for crypto code.
- Fund initial studies and research seedlings to explore the feasibility of resilient cryptographic algorithms, protocols, and software implementation tools in the context of critical sectors.

4.5.1 Use Cases

- Implement stateless clients for financial transactions that leverage personal mobile hardware tokens. Use a thin client and flush all state after every transaction. Persistence occurs at server and the personal token hardware. Move the security onto personal hardware where it can be defended using resiliency techniques.
- Other areas include critical infrastructure, Supervisory Control And Data Acquisition (SCADA) systems, Voice Over Internet Protocol (VoIP) systems and electronic voting.

5 Connectivity Diversity

Introduce duplicative, rotating network connectivity, redundancy in throughput, larger number of network traffic paths

- Concerns
 - Performance, traffic engineering, limited physical diversity
 - Requires communication between multiple parties
 - Routing/complex communication
 - Limited physical diversity
 - Peer-to-peer communication risk
 - Keeping it simple would make it easier to penetrate
- Mitigation
 - Frequency hopping is an example
 - Commercial products that changes port numbers, IP addresses (eg:, Network Address Translation (NAT))
 - Ubiquitous connectivity
 - Enhancements to IP routing protocols
- Useful help from other groups
 - Cyber-economics group can help by developing economic/business models for assured services that satisfy both network providers and mission-critical users.
 - We need an economic model for Service Level Agreements (SLAs) with provider having incentives to meet SLAs; it is a real "pain" when they don't.

5.1 Description

Connectivity diversity (or path diversity) refers to the ability to provide multiple physical and virtual paths between information sources and users. It includes physical path, transmission media, logical path, provider (carrier), and technology diversity. Also included is the capability to create unpredictable and dynamic paths using intelligent Sense-and-Respond mechanisms that minimize the opportunity for single-points of failure. This makes Denial of Service (DoS) attacks and Man-in-the-Middle (MiM) attacks more difficult to achieve because the path that data packets travel through the network changes in unpredictable ways. End systems do not need to know the algorithm for the path changes; they only the network equipment including edge routers needs to know this. Although the technology exists for path diversity and re-routing, the Game Change is to change paths "unpredictably" (from an attacker's perspective) with Sense-and-Respond intelligence.

The business case / benefits for connectivity diversity (in addition to the cyber-security benefits) includes the use of path diversity as a mechanism to support disaster recovery / continuity of operations Disaster Recovery (DR)/ Continuity of Operations Plan (COOP).

5.2 Inertia

Why have we not done this before? What would derail the change?

- Concerns about end-to-end performance from a user perspective. This includes network performance/overhead to dynamically change the paths without disrupting ongoing data flows/connections. [Note: metrics would be useful here.]
- Complexity of creating and managing multiple diverse paths between endpoints.
- Network providers provide reliable service using lowest-possible cost physical media, not diverse or redundant path. [Note: need better wording / more accurate statement here.]
- Network planning and traffic engineering becomes complex.
- Multi-vendor solutions create operational support Operation Expenses (OpEx) issues as well as more cost up front Capital Expenditure (CapEx).

5.3 Progress

Why technically is this feasible now?

- Network providers now provide foundational technologies (Multi-protocol Label Switching (MPLS), anycast/multicast, IPv6).
- Management and monitoring tools are becoming more sophisticated and autonomous, allowing control at a segment-by-segment level.
- Cloud and Service Oriented Architecture (SOA) technologies combine with architecting at the "Services" level of abstraction (vice the technology level), allowing "Services" to be created and accessed independent of the underlying technology.
- Dynamic Domain Name Service (DDNS) is available.
- Connectivity is becoming ubiquitous, with multiple paths available between endpoints (fiber, copper, wireless point-to-point, cellular, 802.11 (WiFi) and 802.16 (WiMax), satellite, Broadband over Powerline).
- Self-healing network technologies are available.

Why environmentally is this feasible now?

- Many enterprises are already providing limited connectivity diversity for DR/COOP.
- Many network providers are competing in the same market, creating redundant paths between endpoints.
- Provider networks are designed with redundant and diverse paths embedded internally.
- Customers are willing to pay for assured services - commercial business models exist e.g. Quality of Service (QoS).

What would mitigate our doubts?

- Availability of bandwidth enables over-provisioning to mitigate performance problems.
- Failover techniques such as SONET Rapid Path Restoration (RPR) have shown that switchovers can happen instantaneously with near-zero performance impact.
- Planning tools that allow prediction of path performance before alternate path selection can be created using current/near-term technology.

- Management tools can select from pre-defined alternate paths can be created to minimize traffic engineering and management complexity.
- Network providers are already using vendor-diversity to avoid sole-source issues and provide different cost/benefit tradeoffs at the different network layers.
- Mission-critical users are less cost-sensitive when buying assured services - different business cases exist.

5.4 Action Plan

What are reasonable paths to this change?

- Pre-planned disaster recovery scenarios taking advantage of resilient connectivity already exist in some places; these can be leveraged as examples of what's already being done.
- Large scale demonstrations can be created on test networks (DREN, Global Environment for Network Innovations (GENI), very high-speed Backbone Network Service (vBNS+), Planet Lab, Emulab/DETER, etc) in support of cyber-exercises. These demonstrations should be done in conjunction with other cyber infrastructure workshops, cyber war-gaming exercises, etc.
- Incremental network planning steps can be made less complex using "brute-force" techniques – over-provisioning, QoS and dedicated Virtual Local Area Network (VLAN)s.
- An "overlay" approach can be used, starting with a small number of diverse paths and overlaying additional path/segment diversity to build in greater and greater levels of robustness.
- Management tools that can orchestrate the required level of dynamicity may need to be developed and rigorously tested - vendors would have a critical role here.

What would accelerate the change?

- Availability of more sophisticated routing protocols that embed significant connectivity diversity and control within the network layer equipment (analogous to Hot Standby Router Protocol (HSRP)).
- Providing significant incentives to network providers for implementing increased levels of diversity (or, conversely, providing significant disincentives when lack of diversity leads to reliability, availability or performance issues (strong SLAs)).
- Evolving network overlays such as Smart Grid control or Healthcare Information interchange could be designed with the necessary sensors for dynamic path diversity "built-in".

5.5 Jump-Start Plan

Pieces of the action plan that can be started now

- The academic and open-source software community should prototype a solution using sense-and-respond intelligence for a quick proof of concept using open-source routing software (Zebra or Quagga).

- A consortium of government (possibly including NATO nations), industry and academia should identify test bed demonstration opportunities and demonstrate relevant capabilities using research networks (e.g., DREN, vBNS+, DETER, etc).

An example use-case is to have a network with multiple physical and logical paths available using current routing and recovery techniques, engage NSA or other skilled red team to perform a Distributed Denial of Service (DDoS) attack targeted at denying service at a target host; then enabling connectivity diversity and performing the same DDoS attack - access to the host should remain available using other network paths and media. (This use case / test case should prove the hypothesis of defeating DDoS attacks.)

Start longer-term research efforts by building collaborative teams such as:

- Engage network providers (e.g., Verizon, AT&T, etc) to determine their current/planned future state and their approaches for responding to security events, to create a synergistic vision and collection of Best Practices related to path diversity.
- Engage Management Systems vendors (e.g., CA, HP, IBM, etc) about extending capability of management platforms to provide connectivity diversity control using Sense-and-Respond methods.
- Engage network equipment vendors (e.g., Cisco, Juniper, etc) for discussions of embedding capability within network equipment.
- Engage Internet Engineering Task Force (IETF) to develop standards for diverse connectivity routing platforms.

6 Decoys

What does this change look like?

Most applications, systems and networks are not perfectly secure. Hence, it is a matter of time until they can be compromised in a targeted attack. The core idea of decoys is to distinguish attackers from authorized users and additionally provide a large number of decoys (fake targets) to attackers while only providing the real targets to authorized users. As a consequence, attackers will be slowed down (probably confused or discouraged) by interacting with fake targets and defense will be able to easier distinguish authorized from unauthorized activities, i.e., detect new attack activity. Ideally, this mechanism will be invisible to the authorized user.

- Value - Defense can detect new attack activity, automatically analyze new attacks, and learn predict and prevent attacks based on early attack stages before the attacker reaches the real target. The result is containment of risk from imperfect networks, systems, and applications by deflecting and mitigating attacks as they develop.
- Concerns
 - Legal barriers
 - Management (ability to detect real system in an emergency)
 - Scalability
 - Cost
- Mitigation
 - Virtualization: ability to create multiple decoys, easily
 - Attempt to change legal framework

6.1 Description

Decoys provide several advantages to defenses in cyberspace. First, they can decisively delay and confuse attackers by presenting them with fake targets. Second, since decoys are not usually accessed, any such access points to ongoing attacker activity, which can range from mapping out networks to launching exploits or denial of service attacks. Detecting new or newly initiated attacks, together with slowing down the attacker, the defense wins valuable time to prepare a response or to study attacker's behavior to discover unknown ways of attackers (unknown vulnerabilities or new ways of evading firewalls, anti-virus, or access controls). Decoys can take different forms to effectively protect various security targets. They can fake systems, virtual machines, applications, data, or networks.

Attackers end up at decoys because the decoys are reachable over shortcuts or they may bypass common access control patterns. The decoys increase the attack surface while decreasing the probability of a successful attack on the real target and hence reduce the attack Return on Investment (ROI).

To significantly slow down and frustrate the attackers, the ratio of real: decoy targets must be very low, for example on the order 1:10000. This, in essence, creates a large additional attack

surface that an attacker needs to cover before eventually zooming in on the real target (c.f., Honey pots and Honey nets). There are several ways to 'slow down' attackers at decoys; they reach from simply shallow multi-system emulations listening on ranges of unused network addresses to full fake run-time environments with fixed IP and real business application configurations (traps, jails) that are more difficult to distinguish from real targets even for attackers taking control of the decoy.

6.2 Inertia

Why have we not done this before? What would derail the change?

- Manageability of creating, destroying, migrating decoys and tracking decoys.
- Cost or lack of scalability of building decoys and maintaining them in the 'image' of evolving targets. This requires extremely fast and low-overhead cloning of systems.
- Legal: If users end up at decoys instead of real services there could be legal consequences, especially for critical services (e.g., controller applications, data base applications, financial transaction servers, emergency services, e.g., based on VoIP).

6.3 Progress

Why technically is this feasible now? Why environmentally is this feasible now? What would mitigate our doubts?

- Virtualization answers several important scalability questions:
 - Cloning of VMs becomes as easy as "forking" a processes (copy on write memory and storage might allow instant cloning even of fully deployed VMs at run-time)
 - Default configurations of NAT-ed, and encrypted communication channels with appropriate access controls prevents attackers from easily sort out decoys by observing network traffic.
 - Optimization based on hardware or OS level virtualization enables to prioritize real targets to limit the overhead of decoys. Such optimization might offer opportunities for attackers to distinguish Decoys from real targets (e.g., response time or other side-channels).
- Advanced analytical capabilities to correlate large traffic streams in real-time enable real-time learning by observing attacks on random decoys to protect the real target.

6.4 Action Plan

What are reasonable paths to this change? What would accelerate this change?

- Develop real-time 'multi'-cloning of VMs or applications at minimal cost and in various depths (OS/Application simulation --> full cloning).
- Develop OS/Apps that automatically create shadow decoys for data and executable files to confuse attackers (data) or increase cost of planting Trojans. Could be seen as a form of file-system randomization.

6.5 Jump-Start Plan

Pieces of the action plan that can be started now

- Create decoys or "shadow" services for systems or VMs on demand for high value targets. Leverage existing honey pot technology, such as Honey nets and Black-hole sensor systems (e.g., see Internet Motion Sensor). Configure the decoys according to the perceived threat if required (e.g., make sure the attacked service or OS is emulated or simulated sufficiently to not raise suspicion).
- Analyze distributed attacks detected at sensors to layout the best positions for in-line network Intrusion Prevention Systems (IPS). Then, enable decoys to create detectors or simply signatures on-the-fly. Finally, configure IPS at those strategic network positions and provision them with those newly created signatures or detectors. Virtualized environments offer sufficient capabilities to instantiate network IPS, e.g., on open source industry standard such as Xen, using Domain0 network interception, or VMware using the VMSafe introspection APIs. Real-time stream analytics can analyze decoy sensor data even in case of broad attacks on-the-fly and correlate it with network layout information to determine strategic intersection points for the IPS.
- Test signature and detector creation in a small setting, then run large scale tests to validate and optimize the positioning of IPS for different network topologies, e.g., use private virtualized testbeds.
- Later steps would include moving from the black-hole/honey pot approach that traps random attacks to a close-target approach that can protect individual systems (identified by IP address) or applications (IP address + protocol + port number). This requires (a) sophisticated real-time analytics that safely differentiate between attackers and authorized 'clients', and (b) a balancer that forwards requests from authorized clients to the real target and requests from potential attackers to decoy copies of the target.

6.5.1 Use Cases

- First layer of defense, slowing down attackers and offering a pre-warning system.
- Contain risk (raise cost of attackers) of unnoticed compromise of high-value targets through zero-day exploits or other vulnerabilities by external attackers.
- Safely study and analyze new attacks in cyber space to create models for attack prediction, prevention, mitigation, and response.

7 Configuration-Space Randomization for Infrastructure

7.1 Description

Configuration is the glue that logically integrates components to support end-to-end services. It defines the logical structures and relationships at and across multiple protocol layers. Acquiring this information is critical for attack planning, e.g., for identifying high-value targets, the paths to reach them, the intermediate components to compromise, and customizing attacks to each target. We propose to make this information much harder for an adversary to acquire by randomizing it, but, doing so in such a way, that end-to-end services continue to be available. This is analogous to address-space randomization for software that makes it much harder to plan buffer overflow attacks and frequency hopping that makes it difficult to plan jamming attacks on communication links.

Notes

- A medium-scale infrastructure can contain 100,000 configuration variables defined in the configuration files of its components. Thus, there is a very large space of possible configurations. Rapidly “moving” between different points in this space can make it very hard for an adversary to guess the correct configuration, and rapidly invalidate his “map” of the configuration.
- The idea can be used to protect infrastructure at any layer: physical, MAC, network, virtual private networking, messaging, peer-to-peer and application. Examples of configurations that one can change are addressing, security policies (firewall rules), virtual networking architecture, routing protocol architecture, and virtual server architecture.
- The idea is orthogonal to diversity because one can change configuration without diversity and still confuse an adversary.
- The idea is intended not only to resist but also survive intrusions and contain their damage.
- NOTE: A capability to find a new configuration satisfying end-to-end requirements is a useful one for other approaches to moving-target defense. For example, if a new virtual machine replaces an existing one, its needs to be configured to support all services that depend on it. In general, its configuration is not identical to that of the virtual machine it just replaced.

7.2 Inertia

- Infrastructure design, computing configurations consistent with end-to-end requirements, and debugging configurations to enforce these have been very hard problems. Today, these are manually solved. Dynamic reconfiguration has, therefore, been inconceivable.
- Governance, especially in a collaboration environment is hard. If there is no centralized configuration authority, then reconfiguration that is consistent with intended policies of all collaborators requires agreement of all of these.
- Scalability, cost and operational impact and corporate acceptability have to be proved.

7.3 Progress

- Modern model-checkers and SAT-based constraint solvers allow one to efficiently compute configurations satisfying end-to-end requirements. These can solve millions of constraint in millions of variables in seconds.
- Modern fault-tolerance protocols (including routing protocols for networks) allow millisecond-scale reconfiguration. Of course, these must be correctly configured or recovery is precluded in spite of availability of redundant resources.
- Virtualization has become widely available, accepted and efficient.
- Resources have become much cheaper allowing us to create diversity and redundancy.
- There are well defined interfaces to infrastructure components for their control and configuration.

7.4 Action Plan

- Understand business case for idea in consultation with administrators that operate real infrastructure. An example of this would be Defense Information Systems Agency's (DISA) or the National Security Agency's (NSA) collaboration infrastructure that use host and network virtualization.
- Develop faster methods of translating end-to-end requirement/specification into configurations
- Develop faster safe reconfiguration methods, i.e., for changing configuration without disrupting mission-critical services or introducing security breaches
- Develop distributed reconfiguration methods
- Develop cooperative reconfiguration methods to allow implementation of idea across administrative boundaries
- Quantitatively evaluate effectiveness of idea with mid-term and final "exams". "Exams" will be administered by red teams

7.5 Jump-Start Plan

- Realize the IETF spirit of rough consensus and running code
- Team with administrators of real collaboration infrastructure e.g., from DISA and NSA. These use both host and network virtualization.
- Team with red-team experts at these organizations
- Identify the security and functionality requirements that administrators most care about
- Create a test bed with e.g., routers and virtual machines, and set up these requirements. This test bed can be set up in e.g., DETER, or in partnership with a company with large laboratory capabilities.
- Define and implement configuration randomization plan
- Quantitatively evaluate increase in red-team's difficulty in successfully violating security or functionality requirements. Also, assess performance impact.

7.5.1 Use-cases/Scenarios

- A worm may try to locate the address of a server offering a particular service. But it may need to compromise other machines before it can attack the server. Before, the adversary has had a chance to compromise other machines, our system would have randomly moved the service to another machine, so the attack would be rendered ineffective.
- Host-to-host traffic is randomly made to flow through tunnels and firewall policy is changed to permit only tunnel traffic. Then, an adversary's packets are blocked.
- The layering of IPSec tunnel architecture over the IP network is randomly changed. If an adversary had planned on sniffing at a component where IPSec traffic is decrypted, that plan would be invalidated.

8 Distributed Data Shell Game

Idea: Break data into pieces and move it around. The results will ensure all aspects of CIA: Confidentiality, Integrity and Availability. The process obscures data thereby assuring confidentiality. Any violations of a piece of data's integrity will result in failure to recombine. Availability is enhanced by distributing the risk across locations and allowing recovery when a location is lost. The addition of cryptography to the system will further increase confidentiality and privacy.

- Break data of interest to the attacked in multiple pieces, spread them to different – redundancy scattering – fragments have to be operated on. Use different keys
- Bit torrent
- IP issues – originally driven by the need to compress data
- Low hanging fruit

Concerns

- Larger bandwidth costs
- Law enforcement issues: how do you recover data
- How to write applications (legacy)
- Cultural problems
- Cost

Mitigation

- Improving data de-duplication and redundancy
- Low cost storage
- Already proven (bit torrent, cloud computing)
- Data vanishing
- Easy APIs

8.1 Description

- Break up data into pieces and distribute those pieces to different locations, which could be logical or physical. Individually the pieces reveal little to an adversary. They can only be combined at the time of proper authentication.
- To add another hurdle to the attacker, the locations of the pieces change periodically. The rate of this change will be based on the level of risk. For example, the rate of location switching increases as the number of incidents increases or as the value of the data increases.
- Cryptographic techniques can be added at the time of the data separation or at later stages in the process.
- Design into the system an audit trail that shows what has accessed and combined the data.

8.2 Inertia

- Cost of storage

- Infrastructure-centric data model
- Cost of bandwidth
- Performance hits on the database
- Increase in network latency
- Culture of people seeking local control over data

8.3 Progress

- Lower cost of storage
- People are getting used to storing their data remotely both at an individual and corporate level
- More suppliers of bandwidth for data movement
- Distributed data bases are becoming more accepted
- Network management is driving up network efficiency

8.4 Action Plan

- Demonstrate the new capability to national leaders in a major test range. Use NSA's red team to attempt to identify the moving data. Identify the additional work effort needed by the attackers to reach the data.
- Market the idea as a business continuity capability that allows a business to recover operations when one location is lost. Other locations will have other pieces of the data and can recalculate and re-assemble the data. This distributes the risk of a failure at any one location, and highlights its benefit for information availability.
- Promote the value of the system for being able to detect the integrity of the data. You can't reassemble the data, if any of the pieces has been compromised.
- Emphasize to early adopters its value for reducing concerns with data destruction and archiving because the data at any one location is of no value -- one can leave it behind.

8.5 Jump-Start Plan

- Develop a limited demonstration of a few elements of the solution leveraging currently available technology such as the Tahoe File System
- Go to industry standards group and show them what was accomplished
- Make the information available to the consumer and vendor community with the goal of creating a consumer demand.

8.5.1 Use Case

- Human resources (HR) and financial data are two of the most critical assets of any company. Both types of data, which are both competition-sensitive and personally private, need to be accessed frequently by authorized users. The confidentiality and continuous availability of this data must be assured for business operations. Currently this data is centrally stored.
- Users at the corporation or its partners gain access to the data base, and often copy the data into their local space. This exposes more data than necessary to users, and fosters uncontrolled distribution of copies.

- By distributing this data into dispersed locations, its confidentiality is assured. Yet, by allowing authorized users at either the corporate site or partner sites to access a recombination of individual data records assures its access to those who need to use it. There are certain times when this data's sensitivity is more critical and its loss presents even greater risk than normal; for example, just prior to running an earnings report. At this time, the locations of the data are changed, i.e. the data becomes a moving target.

9 Security on Demand

Change the current mindset from the idea that security needs to keep bad guys out to assuming that we are essentially in a fundamentally insecure environment. Therefore, if you need security (trustworthiness), you need to do things differently. The "things you would do differently" will present a computing void to the adversary (i.e., if he breaks in he will not find the address book, which will reside on the detached stick; if a zombie is installed, most of the time, he will not have a fully functional network to propagate-in general, he will have access to useless information, resources etc, or things that will become useless within a short period of time). You will dynamically constitute a "trustworthy cocoon" -- on demand, to run the application that needs higher security. The cocoon will include the application as well as the infrastructure you need to use that application, and the trustworthiness will be verifiable. At the same time, the cocoon will take a different shape (variant) each time, and each cocoon will be short lived, and exposed to public networks for a short duration.

Note this is not a silver bullet to all problems-- this technique will work better for applications that do not need long duration sessions.

- Break data of interest to the attacker into multiple pieces, spread them to different – redundancy scattering – fragments have to be operated on. Use different keys
- Bit torrent
- IP issues – originally driven by the need to compress data
- Low hanging fruit

Concerns

- Larger bandwidth costs
- Law enforcement issues: how do you recover data
- How to write applications (legacy)
- Cultural problems
- Cost

Mitigation

- Improving data de-duplication and redundancy
- Low cost storage
- Already proven (bit torrent, cloud computing)
- Data vanishing
- Easy APIs

9.1 Description

- Break up data into pieces and distribute those pieces to different logical or physical locations. Individually, the pieces reveal little to an adversary. They can only be combined at the time of proper authentication.

- To add another hurdle for the attacker to overcome, periodically change the locations of the pieces. The rate of this change will be based on the level of risk. For example, the rate of location switching increases as the number of incidents increases or as the value of the data increases.
- Cryptographic techniques can be added at the time of the data separation or at later stages in the process.
- Design into the system an audit trail that shows who has accessed and combined the data.

9.2 Inertia

- Cost of storage
- Infrastructure-centric data model
- Cost of bandwidth
- Performance hits on the database
- Increase in network latency
- Culture of people seeking local control over data

9.3 Progress

- Lower cost of storage
- People are getting used to storing their data remotely both at an individual and corporate level
- More suppliers of bandwidth for data movement
- Distributed data bases are becoming more accepted
- Network management is driving up network efficiency

9.4 Action Plan

- Demonstrate the new capability to national leaders in a major test range. Use NSA's red team to attempt to identify the moving data. Identify the additional work effort needed by the attackers to reach the data.
- Market the idea as a business continuity capability that allows a business to recover operations when one location is lost. Other locations will have other pieces of the data and can recalculate and re-assemble the data. This distributes the risk of a failure at any one location, and highlights its benefit for information availability.
- Promote the value of the system for being able to detect the integrity of the data. You can't reassemble the data, if any of the pieces has been compromised.
- Emphasize to early adopters its value for reducing concerns with data destruction and archiving because the data at any one location is of no value -- one can leave it behind.

9.5 Jump-Start Plan

- Develop a limited demonstration of a few elements of the solution leveraging currently available technology such as the Tahoe File System
- Go to industry standards group and show them what was accomplished
- Make the information available to the consumer and vendor community with the goal of creating a consumer demand.

9.5.1 Use Case

- Human resources (HR) and financial data are two of the most critical assets of any company. Both types of data, which are both competition-sensitive and personally private, need to be accessed frequently by authorized users. The confidentiality and continuous availability of this data must be assured for business operations. Currently this data is centrally stored.
- Users at the corporation or its partners gain access to the data base, and often copy the data into their local space. This exposes more data than necessary to users, and fosters uncontrolled distribution of copies.
- By distributing this data into dispersed locations, its confidentiality is assured. Yet, by allowing authorized users at either the corporate site or partner sites to access a recombination of individual data records assures its access to those who need to use it. There are certain times when this data's sensitivity is more critical and its loss presents even greater risk than normal; for example, just prior to running an earnings report. At this time, the locations of the data are changed, i.e., the data becomes a moving target.

10 Security on Demand

Change the current mindset from security needs to keep bad guys out to assuming that we are essentially in a fundamentally insecure environment. Therefore, if you need security (trustworthiness), you need to do things differently. The "things you would do differently" will present a computing void to the adversary (i.e., if he breaks in he will not find the address book, which will reside on the detached stick; if a zombie is installed, most of the time, he will not have a fully functional network to propagate-in general, he will have access to useless information, resources etc, or things that will become useless within a short period of time). You will dynamically constitute a "trustworthy cocoon" -- on demand, to run the application that needs higher security. The cocoon will include the application as well as the infrastructure you need to use that application, and the trustworthiness will be verifiable. At the same time, the cocoon will take a different shape (variant) each time, and each cocoon will be short lived, and exposed to public networks for a short duration.

Note this is not a silver bullet for all problems-- this technique will work better for applications that do not need long duration sessions.

- Separate VM for each application that can be run on a USB device (a stick with enough CPU/memory to run Linux) – e.g., Spyros Rosetta
- Leverage emerging processor architecture like Intel Virtualization Technology (VT)/Active Management Technology (AMT) or Advanced Micro Devices (AMD) Pacifica to establish a trusted path from the USB device to the laptop/desktop
- Use the laptops capability to do IO Kernel-based Virtual Machine (KVM) + Network))
- Low hanging fruit

Concern

- These things can be attacked also
- Smart sticks could be poisoned – could be shipped with malware
- Inability to pass data between different domains
- Acceptability

Mitigation:

- Proven, devices exist
- CM is manageable

10.1 Description

(Concept of Operations (CONOPS)) What will it look like?

Imagine the future where traditional desktop/laptop computers have become the chassis on which key chain computing Secure Digital Input/Output (SDIO) devices with enough CPU and memory to run Linux and at least one VM can be plugged in-- the laptop/desktops will only be used to provide the IO/peripheral functions to the key chain devices. You will have one dedicated device for each of your critical applications (e.g., email, banking, Google app client etc.) running a VM

specialized to run that app—(e.g., all other services and ports disabled). A verified version can be preloaded to the device, but the VM can also have software to load variants of the app (leverage SW diversity) from a "trusted source" (see below for how to get to that source). It is also conceivable that the device will only have a very basic loader-- and when you connect to the network you will be pushed a secure variant of the entire VM.

If you need to use banking, you plug in the "banking app" device. The device-chassis pair engages in attestation checking (TPM and other HW support in modern processor architectures). If the check succeeds, the device boots up. Then, from the device's memory, a functionally equivalent variant of that application could be loaded to run on the device. (Alternatively, as noted before, a variant can be downloaded after you have network access).

When the device boots up, you (the user) request a protected path (imagine establishing a VPN tunnel) to destination from your network provider. For this to work, like a telephone network, the chassis must have a dial tone--i.e., instead of always on broadband, the chassis is connected to the ISP with a very basic highly controlled channel. If your request for secure path is granted, you have a fatter pipe, but also with VPN-type protection. You can have better QoS if you pay more:

- Then you use your application to do your transaction, save data on the device (or copy if you need to save VM (actually data for VM if any) on that, hung up on the protected path and unplug.
- Analogous things could be done at the server side too. Imagine the enterprise procuring CPU/servers from the cloud, and establishing links between them on demand.

Benefits

- The application is online for a short duration (short exposure for adversary)
- You are not connecting to the chassis unless you verify its attestation.
- You run a different variant each time.
- You procure a secure link each time.
- Enterprise management and IP rights management become easier (when the application is pushed to the stick device).

10.2 Inertia

- Concerns/inertia
- Device technology was not mature (CPU/memory on stick)
- Virtualization technology was not there
- Bandwidth on Demand (BoD) was not there
- The concept has not been demonstrated/evaluated for scale/complexity

Derailers

- There is a bootstrap issue-- easy to see the client side CONOPS. If we make the server/services moving, how do we connect the client and server in a trusted way? Man in the middle?
 - Mitigation approaches: Secure Directory/discovery services that becomes available with ISP dial tone, leverage Uniform Resource Name (URN), Digital Object Identifier (DOI) handles etc.
- Education/Acceptance-- how to get vendors/users/service providers accept this?
 - Mitigation approaches: for end users, make it easy/transparent; for providers/vendors: show them that there is cost savings or additional revenue stream (new services, control spam, better protection against bot nets etc)
- What if the smart stick is shipped with bad code?
 - Mitigation: What if MS (or choose your favorite vendor) ships your favorite product on a media that you paid for? This is no different, and no worse.
- What if the chassis computer being attacked (corrupt, rootkitted, recruited by botnet)?
 - Mitigation: The proposed solution is no worse than what we currently have. BoD limits exposure/usefulness of these attacks. Processor architecture (and other mechanisms can be engineered -- prior work exists)will facilitate isolation of all communication from keyboard to the stick

10.3 Progress

Feasible Technology

- VM, BoD, attestation techniques are here now.
- Mechanisms to create SW diversity automatically and at a low cost and with different vulnerability mix has been demonstrated (Just-in-time (JIT)), link/load level transforms, compilers).
- Cloud computing, Spread spectrum/"hopping" techniques are commercially available.

Environmentally Feasible

On the environmental front: realization that we are under attack, and perfect security that will prevent that is a pipe dream.

10.4 Action Plan

- Need to serve a wide range of users (Grandma to mission critical).
- Need to engage different stakeholders: Government services (enterprise applications), big defense contractors (mission critical applications), academic/industrial research, network providers, hardware (processor and SDIO manufacturers).
- Assemble a dream team: one intellectual lead (who is in there not sell products, but get paid for the R&D); one service/sw vendor (to offer their software on Security on Demand (SoD) sticks or a defense contractor for transition to mission critical application; one network provider to offer BoD; one hardware vendor to offer new hardware platforms; one academic research institution to liaison with academic research/open source community.

- For longer term, the dream team will develop SoD applications for the proposed Healthcare Information Network or the emerging Smart Grid.

10.5 Jump-Start Plan

Do an advanced technology demonstration (ATD) pilot on a moderate scale: choose one application (a good attack target such as outlook and exchange), give 500 random volunteers the stick device loaded with SoD client and host a dynamically managed SoD exchange servers in multiple clouds. Use BoD among the Exchange servers, allow volunteers to request for protected service from the ISP. Engage a red team to attack the clients. This project is shovel ready (BBN Technologies and CSC inputs to the NITRD Conference Leap Year processes provide more detail, prior work from a SANS can be rolled in as well) and can be started in the next 60-90 days. The project will have a 9 month development phase (to work out the right scope and remaining engineering) followed by a 9 month field trial.

10.5.1 Use Case

- Need a sponsor to convene the team of various stakeholders including the application owner, hardware vendor, network provider and architect/integrators.
- The outlook-exchange target application may not be a good example-- perhaps a specialized browser for doing financial transactions is a better one where the client state can be at various places (adds one dimension for varying the application).

11 Terrorist Organization Model

Idea: use the decentralized nature of terrorist groups and cells as a reference model for a new information system. Terrorist groups are hard to penetrate, not susceptible to large losses if a subpart is compromised, and can work autonomously with a very small rule set. This model is a "game changing" idea in that it approaches computer and network science in a radically different manner.

- Study terrorist model and why it is hard to penetrate, how it is resilient, if one gets captured, all get captured
- Concerns
 - Revolutionary change compared to the current hierarchical model
 - Cultural resistance
 - Lots of unknowns

Mitigating

- Coalition: sharing networks (concept worked on by NATO) – low hanging fruit
- Gaming industry – massive multiplayer online games
- Lessons learned from mobile ad-hoc networks (MANETs)
- Cultural acceptance from the new generation

11.1 Description

This is a fundamentally different approach to information systems as compared to today's hierarchical models.

- Rather than linear command/control relationships, tight lines of communication, and high dependence on the successful operations of other groups (processes) the terrorist model has very loose ties, autonomy of parts, and self organized leadership.
- It also has other attributes that make it very resilient to penetration and disruption such as "tribal leadership" or "headless organizations".

11.2 Inertia

There would be significant cultural resistance to this approach, due to the many decades of development invested in the current architectures and reference models. Also, the idea of "terrorist groups" is offensive to many and might hamper good innovation and creativity. There are many unknowns and not much literature on the specifics of how these groups communicate and protect themselves.

11.3 Progress

Some applications use an early and crude application of this methodology such as Massive Multiplayer Online Games (MMOGs), disposable hardware devices, social networking sites, web 2.0, and the portion of our society known as "Generation Y". Service Oriented Architectures (SOAs) might also provide some insight into how this model might work due to the "loose coupling" of services offered by SOA.

11.4 Action Plan

Need to better understand how terrorist groups organize, how their information networking evolves, why they are hard to penetrate, where the resilience comes from, and how the capturing of one person or cell has little impact on the entire operation. These groups might follow the principles of complex and chaotic systems, which could in turn provide insights for a new reference model for information systems.

11.5 Jump-Start Plan

- Use existing sharing networks and systems such as that being developed by NATO, lessons from MMOGs, or even concepts from MANETs as a basis for developing an experimental framework or model.
- Leverage the different cultural values of the Y Generation, and create a Facegroup page, Wiki, or other virtual meeting place where this idea can be discussed and fleshed out.
- Obtain funding from Department of Homeland Security (DHS)/Science and Technology (S&T) for a pilot in this area, and establish a public/private consortium to develop proof-of-concept technical solutions.

12 Smart Motion Adaptation Management

Redundancy and diversity in SW, infrastructure and resources create the space where defended systems can shape shift. Develop a sound model to manage the movement in that space such that it is unpredictable to the attacker. Use variety of modeling techniques including but not limited to game-theory, machine learning, statistical, control theory, cognitive reasoning and planning to develop the algorithms that manage the dynamic system behavior.

- Model based motion management, M^4
- Black hat:
 - Hasn't been enabled in terms of mechanisms
 - Scalability
 - False positives – problem common in these approaches
 - Practitioners have good ideas looking for a fit
 - Does the model fit reality?
- Yellow hat
 - Provability feature
 - Way to adapt
 - High speed processing
 - Bayesian decision trees
 - Advanced reasoning engines

12.1 Description

The "smart management" will use the various options and/or possibilities unleashed by other techniques. For example, how to place replicas, which address/port to use, which variant to use, how to configure the network (overlay/interconnection) etc. all dynamic adaptation decisions will be governed by this smart management mechanism.

Benefits

- System dynamically configures itself for optimal security-performance trade off
- Proactive (as opposed to reactive - limit exposure)
- Adaptation is based on sound theory - easier to establish the operating regions (bounds, control theoretic proofs that certain bad conditions will never arise)
- Performance improvement
- Financial impact and brand protection

12.2 Inertia

- The degrees of freedom to navigate and the space to manage was smaller or not there-- it is now (or we can see how it can be) with the other techniques before.
- Mathematical formalisms were not mature.

- Processor speed/capacity to run the compute intensive adaptation management decision making algorithms
- The communication bandwidth needed to compute the decisions was not there (wide area, reliable, ubiquitous, high bandwidth)

Derailers

- Decision cycle time: need to move faster than the attacker
- Complexity of the algorithms: self explanatory
- Model fidelity: how do we know that the model fits reality
- Uncertainty/incompleteness of observations/measurements driving the decision model
- Attacks on the management may lead to delayed or plain incorrect decisions
- Acceptance (validation) of automated adaptation management can be tricky (how do I know it will do the right thing?)

12.3 Progress

Feasible Technology

- Proof of concept of various types of adaptation management capability (algorithms, models) and architecture (hierarchical, centralized, peer to peer) demonstrated
- Diversity/redundancy space to manage now available

Environmentally - The stakeholders are more receptive now-- with the adaptation space growing large, smart management is inevitable.

12.4 Action Plan

- Identify a transition target (smart grid/Healthcare Information Network) -- build the new entity such that it has smart dynamism built in.
- Grid or HIN with smart management cannot be built in one step--attempt to reach interim milestones: First build a smart management mechanism that works in a passive mode (it gets all the data, does all the computation, produces results-- but does not control the system --- the results are for humans to validate the mechanism). As the second milestone, use the smart management mechanism as an expert assistant -- it will offer suggestions to real operators/controllers, and perform some tasks automatically, but under operator's supervision --- operator needs to check off first. The final milestone is to make the smart management system fully operational -- the operators will still have a override switch.
- Assemble a team to work on this. A number of past Defense Advanced Research Projects Agency (DARPA) and National Science Foundation (NSF) funded project developed and demonstrated building block capabilities that can be used

12.5 Jump-Start Plan

- Developing a moderate scale smart management architecture can start within the next 60-90 days. Existing (e.g., DETER, Planet Lab) and planned (National Cyber Range) testbeds can be used to provide venue for testing. After the initial proof of concept, make this framework open such that "expansion technology" vendors can contribute their

technology and create their own experiment to see how the smart management mechanism can effectively manage it, what the issues (performance, new vulnerability) are so that new research can start to address them.

- Different increments with increased scale, increased scope (more dimensions to manage). Initial candidates of "expansion technologies" that can be integrated with the initial smart management architecture framework are "software diversity" and "infrastructure diversity".
- Validate each increment (test, red team).
- Dream team for the pilot: one team experienced in building adaptive and survivable system architecture, technology providers in the software and infrastructure diversity, a government and private sector stakeholder who could use the smart management capability and provide the use case/threat requirements etc., and a red team like IV&V.
- The first step is to identify a sponsor and put together the dream team.

APPENDIX A: Related References

APPENDIX B: Acronyms

Acronym	Description
ATD	Advanced Technology Demonstration
AMD	Advanced Micro Devices
AMT	Active Management Technology
ARL	Army Research Laboratory
ARO	Army Research Office
BoD	Bandwidth on Demand
CAIDA	Cooperative Association for Internet Data Analysis
CapEx	Capital Expenditure
CDN	Content Delivery Network
CONOPS	Concept of Operations
COOP	DisaContinuity of Operations Plan
COTS	Commercial Off-the-Shelf
DARPA	Defense Advanced Research Projects
DDNS	Dynamic Domain Name Service
DDoS	Distributed Denial of Service
DETER	Testbed for Network Security projects
DHCP	Dynamic Host Configuration Protocol
DHS	Department of Homeland Security
DNS	Domain Name System
DOI	Digital Object Identifier
DoS	Denial of Service
DR	Disaster Recovery
DREN	Defense Research Engineering Network
FPGA	Field Programmable Gate Array
FIPS	Federal Information Processing Standards

GENI	Global Environment for Network Innovations
HR	Human resources
HSRP	Hot Standby Router Protocol
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IPS	Intrusion Prevention System
IPv6	Internet Protocol Version 6
JIT	Just-in-Time
KVM	Kernel-based Virtual Machine
MANET	Mobile Adhoc Networks
MMOG	Massive Multiplayer Online Games Massive Multiplayer Online Games
MPLS	(Multi-protocol Label Switching
NAT	Network Address Translation
NSA	National Security Agency
Nessus	A network scanner tool
NIST	National Institute of Standards and Technology
Nmap	Network Mapper
NSA	National Security Agency
NSF	National Science Foundation
OpEx	Operation Expenditure
OTP	One Time Password
PII	Personally Identifiable Information
PKI	Public Key Infrastructure
QoS	Quality of Service
ROI	Return on Investment
RPR	SONET Rapid Path Restoration (RPR)
SAT	Boolean Satisfiability

S&T	Science and Technology (S&T)
SCADA	Supervisory Control And Data Acquisition
SDIO	Secure Digital Input/Output
SOA	Service Oriented Architectures
SoD	Security on Demand
SLA	Service Level Agreement
SONET	SONET Rapid Path Restoration (RPR)
SSL	Secure Sockets Layer
VT	Virtualization Technology
TCP	Transmission Control Protocol
TPM	Trusted Platform Module (TPM),
TC	Trusted Computing
vBNS	Very Highspeed Backbone Network Service
URN	Uniform Resource Name (URN)
VLAN	Virtual Local Area Network
VoIP	Voice Over Internet Protocol
WIFI	Wireless Fidelity
Xen	Open source industry standard