Lessons Learned from Designing a Security Architecture for Real-World Government Agencies

Amy Hughes, Hamed Okhravi, Kevin Perry, Nabil Schear, Richard Shay, Mary Ellen Zurko, and Paula Donovan
MIT Lincoln Laboratory
Lexington, MA
{ahughes, hamed.okhravi, kevin.perry, nabil, richard.shay, maryellen.zurko, pjdonovan}@ll.mit.edu

***Abstract*—Deploying cybersecurity technologies in large enterprises is a challenging task. In this article, we highlight our lessons learned from developing a proposed security architecture for U.S. federal departments and agencies. Our objective is to help security researchers and practitioners better understand the nuances involved.**

I. Introduction

New cutting-edge cybersecurity technologies can be exciting. There are many new technologies popping up every day – so many that it can be a bit overwhelming at times. However, these new technologies do not exist or operate in a vacuum. Instead, the individual technologies operate in – and become a part of – the technological ecosystem that we call the *architecture*. An architecture can be large or small – from a home user to a massive government agency. Whenever a new technology is developed and deployed, it is often in the context of an implicit architecture. So while the concept of a security architecture may be less immediately exciting than whatever the latest individual piece of technology is, having a working, cogent, and dynamic architecture is necessary to utilize that new technology.

Because of the growing threat to government networks and systems both large and small, we at MIT Lincoln Laboratory (MIT LL) were recently tasked with proposing a unified security architecture for U.S. federal departments and agencies (D/As). In the process of designing this architecture, we uncovered several surprising findings about security architectures and how to incorporate disparate security technologies into an architecture. We learned a number of salient lessons that we would like to share.

Why can't we just take the existing best individual practices and technologies and mash them together? Our experience is that security is not just a collection of products. In fact, we found that there are many possible unintended consequences to rolling out a new technology. For example, failure to consider the user impact of a defense can prevent it from being accepted or used as an effective security measure. Additionally, a security technology may be prohibitively expensive, or cause major integration and compatibility challenges.

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited. This material is based upon work supported by the Department of Defense under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Defense.

In this article, we share our lessons learned from our efforts designing a security architecture for the federal government. We highlight a number of topics that deserve consideration when evaluating a prospective security technology for an existing enterprise. Our lessons are related to introducing new security technologies and best practices to a system, with as little pain as possible. Practitioners can consider our lessons when deciding which technologies to adopt. Security researchers can consider our lessons to help their upcoming technologies become more easily adopted.

We present our lessons learned about introducing a prospective security technology to an organization's enterprise network in five categories. First, we observed that there are *constraints* on what technologies the organization can introduce – especially related to cost and compliance (Section V-A). Second, our experience leads us to believe that an organization should consider the nuances of the *threats* that it faces and whether and how these threats are addressed by the technology (Section V-B). Third, we found that the organization should consider the *user impact* of the prospective technology, and how it may impact user sentiment and behavior (Section V-C). Fourth, each organization has its own character and may benefit from considering whether the prospective technology can be *tailored* to its unique context and circumstance (Section V-D); a great technology for one enterprise may be a poor fit for another. Fifth, we believe that an organization considering a security technology should consider a realistic and acceptable *timeline and budget* (Section V-E); for example, some technology will require a lengthy, phased roll-out process.

II. Background and Scope

Over the past decade, government organizations of all sizes have seen a substantial uptick in the number and sophistication of cyber attacks they face [1]. Several large government organizations, such as the Department of State [2] and the Office of Personnel Management [3], have been targeted by advanced attackers and their data was stolen en masse. Even smaller government agencies, whose missions and data are not of great interest to nation-state attackers, are being caught up in criminal hacking activity like ransomware and DDoS-for-hire botnets [4]. An active and burgeoning security product marketplace has emerged to provide products that claim to quell the concerns of Chief Information Security Officers (CISOs) about these attacks. Products range from firewalls to

zero-trust networks and from anti-viruses to incident response platforms. Setting up all of these products in an enterprise represents a substantial challenge because there is little guidance on how to create a unified architecture that sensibly combines the products. This lack of architectural thinking has made creating effective cyber defenses illusive to the federal government to date, even in their largest-scale defense systems like Einstein [5].

To address this architectural gap and help improve cybersecurity for the federal government, MIT LL was tasked by our government sponsor to propose a next generation security architecture to be used in all U.S. federal D/As. This project involved considering numerous classes of threats, matching them to a set of already-existing or nearly market-ready technologies and capabilities, and building a holistic architecture for the D/As. In addition to dealing with threats and providing a sound collection of the current and future security products in the marketplace, the architecture also needed to address the challenges of adoption in the government including compliance, piecemeal acquisition, and user acceptance.

The word *architecture* is an overloaded term in our community, referring to high-level concepts such as enterprise architecture and others. Our usage of the term architecture in this article is a domain-specific one that refers to the elements of a system, their integration, and inter-working with a focus on their security implications. This usage, in essence, is a shorthand for the above description, and may be different from other senses of this word with which the reader might be familiar.

We also emphasize that this article does not actually describe the architecture itself. Rather, we focus on presenting the practical lessons we have learned about the caveats of deploying such an architecture in D/As that might benefit future efforts. Moreover, this article does not have the space to present an exhaustive, systematic list of all of the lessons we learned by researching in this space over the course of multiple years. Instead, we are emphasizing and highlighting some of the most salient and practical take-aways.

This paper aims to be of immediate utility to both theoreticians and practitioners. This paper does not attempt to create another detailed, theoretical framework for the adoption of new security technologies. As tantalizing as it would be to create such a framework, there are a few practical issues with that approach. First, the space already has at least one mandated framework (the Risk Management Framework). Second, we feel that staying true to our actual experiences during this effort means presenting our findings as discrete actionable lessons, rather than attempting to abstract them into a broader framework. The central intuition we offer is that a new problem-solving technology does not stand alone. Instead, it must be integrated into a larger set of existing technologies. In other words, promising new technology must be viewed as a prospective part of a larger technology ecosystem, or architecture. The rest of the lessons build on this.

III. How It Is Done Today

The requirements for cybersecurity today are defined by various standards and compliance requirements put forth by agencies such as the National Institute of Standards and Technology (NIST) and the Defense Information Systems Agency (DISA). These include the Federal Information Processing Standard (FIPS) series developed by NIST [6] and the Security Technical Implementation Guide (STIG) series [7] by DISA. While these standards tackle individual security controls and practices, they do not provide a holistic approach for security architecture.

Perhaps the most commonly used holistic approach for federal D/As' security architecture today is adherence to the Continuous Diagnostic and Mitigation (CDM) program. Developed by the Department of Homeland Security (DHS), CDM provides federal D/As with capabilities and tools that identify cybersecurity risks on an ongoing basis, prioritize these risks based upon potential impacts, and enable cybersecurity personnel to mitigate the most significant problems first. The CDM "desired state" is encoded in policies, including those required for various forms of compliance, such as the Federal Information Security Management Act (FISMA). Prioritization is based on a security posture derived from NIST SP 800-37 (Risk Management Framework [RMF]).

CDM's technical capability requirements are categorized by Tool Functional Areas. Any particular technology may map to some subset of requirements in one or more of these areas. Thus, architectural coverage may require careful thought and planning to target a suite of technologies with maximal requirement coverage and minimal overlap, as well as appropriate cross technology integration. D/As are offered a menu of products to select from, which are installed by an integrator. The structure of the selections offered to a D/A may necessitate the replacement of existing deployed functional security-equivalent technology with a new product that integrates with the other products on the menu choice. This may have costs in terms of a D/A's understanding of the protections and use of the replacement security technology, and its integration with legacy technologies that remain in place at the D/A.

In particular, CDM heavily focuses on acquisition and compliance. It, however, does not consider how a technology might be *tailored* to the unique environment of D/As (see lesson L4) and it might thus incur significant *cost* for integration with existing products that a D/A might already have (see lesson L1). This may result in overlapping coverage of threats (see lesson L2) and can complicate deployment timeline (see lesson L5). CDM does not consider impact of the technologies on users or their behavior either (see lesson L3).

Another approach today is what is called 'Zero Trust' architectures. Zero Trust architectures are under active consideration by a variety of agencies of the federal government [8]. Lessons learned in that context largely focus on the application of Zero Trust principles in an abstract sense. The most notable concrete discussion of the application of Zero Trust principles is Google's BeyondCorp [9]. This work does not abstract to general lessons learned suitable to a class of organizations, but instead, presents operational lessons specific to Google's needs and goals. For example, considerations of cost are entirely lacking.

IV. What We Did

To accomplish our goal of improving the overall cybersecurity of the federal .gov space, we set out to implement a phased system evolution approach of analysis, design, development, validation, and integration of the existing security architecture. Our analysis utilized a Systems Analysis approach for addressing the current security gaps and needs of D/As. According to Malcolm Hoag [10], Systems Analysis is a "systematic examination of a problem of choice in which each step of the analysis is made explicit wherever possible." There are several Systems Analysis approaches that have been utilized over the years, but the approach we used included four main components: identifying mission needs, defining requirements, performing assessments, and delivering recommendations.

We identified mission needs by conducting both open source and internal research on several D/As. Starting with a review of the vast number of D/As of the executive branch, we conducted interviews with Chief Information Officers (CIOs), Chief Information Security Officers (CISOs), Subject Matter Experts (SMEs), and operators at a handful of different agencies ranging across national security, public safety, and public service missions. Our data gathering included key critical system and data assets which were used both in support of and defense of their missions. In our requirements definition, we analyzed the data to determine which threats would cause potential impacts to the D/A's critical mission assets. We then performed a risk assessment [11] of the threats that would cause the highest impact to the D/A's mission, to determine the likelihood of occurrence for the threat. Based on the risk assessment, we offered recommendations in terms of best practices and tools to mitigate the threats, with a prioritization of actions based on the likelihood of the threat and the difficulty of mitigation implementation for the D/A.

Since the goal was to improve the overall security of the federal .gov space, each of the individual D/A analyses were combined based on threats, likelihood assessments, and mitigations to determine their common problems and needs. Our team then developed an architecture to address these challenges, utilizing best practices and tools to address current and potential future threats. The analysis findings often pointed out challenges that typical private and public organizations face. For instance, many D/A users experienced "cybersecurity fatigue" with mission users bristling at restrictive policies and trying to circumvent them, and several cybersecurity analysts overburdened with firewall, Intrusion Prevention System (IPS), and local device logs trying to find the proverbial needle in the haystack. Moreover, as is usual in the field of cybersecurity, the D/As also lack a standard and widely accepted set of measuring tools and metrics to evaluate their current cybersecurity posture and help them improve. New technologies were hard to deploy due to cost or lack of support for legacy devices, and current capabilities were not well integrated to detect advanced threats. In addition, many D/As relied too much on their boundary firewalls, IPSs, and proxies to protect their networks and devices. These observations and findings drove the underlying principles of the architecture:

- Defense in depth
- Loose coupling
- Actionable analysis
- Context-aware defenses
- Automated recommendations

Each capability in the architecture was designed with these principles in mind to guide current and future deployments of cybersecurity tools and address the common set of challenges mentioned previously.

The architecture was designed around these five pillars directly responding to the five problem and need areas identified by the analysis. The design includes defense in depth to ensure that attacks that bypass one defense are met with at least another. The defenses are loosely coupled to enable easier integration, interchange, and upgrade of tools; and, to mitigate the need for wholesale replacements or changes of existing tools and assets. The architecture infuses actionable analysis to help the cyber defenders better understand the current state of their security and how to improve. Context-aware defenses are utilized to enable better integration and information sharing among the cyber defenses. The architecture implements automated recommendations to help users mitigate problems more quickly, and with less effort.

Our lessons learned meld long-standing best practices with insights from modern field observations. Our design principles of defense in depth, loose coupling, and context-aware defenses draw inspiration from the principles of least common mechanism, separation of duty, and continuous improvement adapted from Saltzer & Schroeder [12], [13]. Whereas, our discussion of actionable analysis and automated recommendations are more necessitated by the cybersecurity fatigue and overwhelming technology options that have recently emerged.

V. Lessons Learned

Our analysis and architecture development uncovered a number of lessons about deploying security technologies into large enterprises which we share in the following sections. We hope that these lessons help security researchers and practitioners in considering major impediments when designing or deploying new security mechanisms.

A. L1: Constraints

Technology deployment can be hindered by constraints. In this section, we describe our experience with a well-known constraint: *cost* and a lesser known constraint: *compliance*.

1) Cost: Cost is often one of the most visible and tangible constraints when deploying a technology. However, the discussion of cost is more nuanced than imagined, and one of our lessons from this effort was that some types of costs create more hindrance in acceptability than others.

The cost of deploying a technology can be divided into these categories: acquisition cost, operational cost, and maintenance/support cost. Acquisition cost can refer to dollars spent on developing, polishing, deploying, and training required to get a technology up and running within an enterprise. This is the "up-front" cost. Federal D/As usually know how to accept and account for this cost because of well-established acquisition possesses. Operational and maintenance/support costs,

however, are much less tangible; therefore, they can create a larger concern for CISOs. Such costs are often not quantified at all by technology providers. If they are estimated by vendors, the estimates are highly dependent on the assumptions about the particular deployment environment, which often does not translate to a different environment. These "on-going," hidden costs scare CISOs. What exacerbates the situation is the lack of meaningful bases to quantify security benefits. Security technologies are often described in terms of attacks that they can prevent. However, how much those attacks would cost if successful, in terms of loss of mission functions, intellectual property, reputation, sensitive data, etc. is not well understood. As such, CISOs are often reduced to making technology acquisition decisions mainly on an up-front cost basis, which misses important dimensions of cost/benefit analyses provided in domains other than security (e.g., productivity technologies like document sharing).

2) Compliance: Another set of constraints, which are often unfamiliar to technology developers, is related to compliance. In order to deploy technologies in almost any government D/A, they have to be certified and compliant with various forms of standards and requirements. Federal Information Security Management Act (FISMA), National Industrial Security Program Operating Manual (NISPOM), and more recently Risk Management Framework (RMF) are some such compliance requirements. Moreover, there are specific requirements that might apply to certain sectors, e.g., Health Insurance Portability and Accountability Act (HIPAA) for health care related D/As, and specific requirements that might apply to certain types of technologies, e.g., FIPS 140-2 for crypto algorithms. Managing such a complex web of compliance requirements is a difficult job on its own; as such, any new technology that might introduce new complexity in that dimension is highly undesirable from a CISO's point of view. Moreover, this precludes the usage of already-available products if their compliance is not yet certified. This complexity and its imposed constraints is often not well understood among researchers. We often face the question: "Why don't D/As just use x to protect against y?" — missing the nuances involved in resolving and satisfying the compliance requirements.

On the other hand, technologies that are already certified, or simplify the compliance process, often have a leg up in competition for deployment in federal D/As. A lesson for the broader community of researchers and developers is to try to make their new technology more easily certifiable by thinking about compliance ahead of time.

B. L2: Threat Modeling Nuances

When considering the addition of a new security technology to an enterprise, we learned that there are many nuances involved in understanding the threats it tries to mitigate. While threat modeling and risk assessment has been mandated by existing standards such as FIPS 199/200, there are practical challenges to its successful realization.

A typical process for threat modeling starts by first creating a list of high value assets. These are the "crown jewels" that need to be protected. Second, a list of expected threats that the organization faces and their likelihood and impact is created. This assesses the *risk* posed by such threats. Third, a list of already-present security technologies (such as those implemented by RMF) can be assembled. With these lists in hand, administrators can begin to evaluate the prospective security technology.

The nuances arise from the fact that different security technologies are not independent of one another. They can be mutually exclusive, overlapping, superseding, or even conflicting with one another. The objective is not to determine whether the technology is useful unto itself, but rather whether the technology is useful to the specific organization, considering the other technologies already deployed in that organization.

When considering the security impact of a new technology, the questions below may be worth consideration.

- Does the prospective technology offer protection from threats that are not already covered by existing technologies?
- If a technology already exists, is it properly configured such that it will be effective against the threat?
- Does the prospective technology enable replacement of any existing technologies?
- Does the prospective technology offer protection from threats that are especially likely or concerning?
- Does the prospective technology interfere with another, already deployed technology?

For example, consider the evaluation of a patch-management technology. An organization might make a prioritized list of its threats and a list of its relevant existing security technologies. The organization may find that there are classes of threats that are almost exclusively addressed through patching, such as software exploits. It may then find an existing solution that covers most of its patching needs. It may also find that there are classes of threats that are not heavily impacted by a patch-management technology, such as insider threats.

The effect that a prospective technology has on the scope of threats faced by an organization is one of the primary considerations when evaluating a new technology, but it cannot be understood in isolation. The prospective technology must be evaluated in the context of other technologies already deployed in an organization, which often complicates the process of threat modeling.

C. L3: User Impact

While the usability of security technology and processes is an established research field [14], and large organizations can consider the labor-cost impacts of security technologies in their Total Cost of Ownership (TCO) or Operational Expenses (OPEX), we found that evaluating the full range and cost of the user impact of deploying a security technology is not an established practice. Unfortunately, lack of consideration of user impacts can result in unexpected costs, delays, and even an undermining of the protections against threats promised by the security technology. Conversely, the lack of applicable information about user impacts and requirements of a deployed security technology can create a "once bitten, twice

shy" effect, where the unknown impacts are a reason not to proceed with deployment. User impacts can be to any human population within, or interacting with, the D/A deploying a security technology, including employee users, administrators, security operations experts, and partners and customers of the D/A.

Change aversion is prevalent with cybersecurity practitioners. Mandates and forced tool changes often cause cybersecurity practitioners to reject potential benefits or improvements of adoption. We observed some of this around discussion of the email-security technologies that are part of what is mandated in the Department of Homeland Security's (DHS's) Binding Operational Directive (BOD) 18-01 [15]. The deployment of email domain authentication technologies such as SPF, DKIM, and DMARC can provide protections against campaigns that use spoofed email-origin information as part of their delivery stage. On the other hand, there are known false positives with each of the technologies that can cause valid email from the organization deploying the protections not to be delivered. Cybersecurity SMEs in some organizations are understandably unwilling to be responsible for mission-aligned email not being delivered due to known false positives or due to errors in code, deployment, or configuration.

This change aversion can be mitigated by good adoption plans that prove the benefits before adoption, but this can take thought, time, and cost. In this example, DMARC policy can be rolled out before the SPF technology it refers to. This policy allows for reporting, but not blocking, email purported to come from the organization's email domain, but that did not properly validate. These reports can be analyzed. There may be false positives that can be avoided with changes in domain validation configuration, or the solution may require changes in the use of email intermediaries (a more costly and potentially prohibitive consideration). Some reports may list the malicious emails that try to spoof their origin. If the technology is reporting and not blocking such emails, a next level of defense that analyzes information for intrusion prevention purposes may be able to make appropriate use of the reports.

An understanding of what technologies will and will not do can be a necessary part of correct deployment, configuration, and use of a technology. In our discussions with mission users, we found that they are not given enough feedback by their technologies to support an understanding of the security posture they provide. This leaves all users in the dark about what security hygiene is needed or no longer needed with the deployment of a cybersecurity technology. For example, when discussing real or potential attacks with a user responsible for some part of the cybersecurity posture, we often hear "Shouldn't 'Technology X' have blocked that attack?". Users build up their model of technologies in part by how they are named and discussed. It seems reasonable to say: "I put in a Next Generation Firewall. Why are Next Generation attacks coming in?" It is possible that a false sense of security from cybersecurity technologies can decrease the protections offered and produce an unknown risk posture for the organization.

An additional false sense of security can come from the compliance-related security practices discussed above. In the types of organizations with extensive compliance requirements, it is common for CISOs to believe that the compliance requirements established by appropriate experts should guarantee them security. This is not a mindset shared by many users, who can see more closely certain compliance requirements' impacts. How best to create, support, and nudge security behaviors in users of all types is an ongoing research question in the usable security research community.

A lesson for the broader community is to expect this misunderstanding from a user when designing a new technology.

D. L4: Tailoring

When designing an architecture that encompasses so many different types of organizations and missions, one must strike a balance between providing sufficient guidance, while not overprescribing the solution. In order to achieve this balance, our approach focused on a decoupled architecture of capabilities to allow for tailoring of the architecture or the capabilities to suit the unique needs of a given D/A. Taking the analogy of a suit tailor, it is easier to start with a slightly larger suit and "bring it in" to fit. In order to know how "large" to make the architecture, it is necessary to discover the variety and scope of the different mission types and assets that need to be protected. An architecture must consider the scale of the protection solutions, which can include number of items to be protected and performance factors of those protections. The architecture must be able to account for legacy solutions that, typically, can't be updated or modified to improve security. Finally, the architecture should avoid instances of prescribing sole-source vendor solutions that limit the ability to integrate with other vendor tools.

The first item that comes to mind when considering how to tailor an architecture to an organization is size and scale. Several D/As complained about the scale of solutions not meeting their needs in both number of instances and in performance parameters, such as bandwidth. Our analysis showed that an architecture that encompassed the federal space needed to account for D/As that ranged from fewer than 100 to several 100s of thousands of systems. Consider a common endpoint protection software, such as Host Intrusion Prevention Systems (HIPS). These solutions must consider factors such as complexity, management overhead, and cost when considering the need to deploy the solutions to every endpoint. Is the configuration easy enough for the small D/A with only two information technology (IT) employees, yet complex enough to account for thousands of different system protection needs? Does its management system scale from 10 instances to 100s of thousands of instances? The cost of a $100 piece of software for one instance can quickly become well over $10 million dollars for a large organization. These scaling challenges are more obvious, but the performance impact considerations can be more difficult to predict and avoid. For instance, most D/As employ boundary defenses to inspect all inbound and outbound traffic, which has led to bottlenecks in throughput for larger D/As and unnecessary complexity, costs, and latency impacts for smaller D/As.

Legacy systems are another pain point for organizations that need to tailor architectural recommendations to their

existing operations. Several D/As admitted that security recommendations were waved due to the inability to modify legacy systems or concern for the performance impacts of doing so. This leaves the legacy system with a deficient defense, often without understanding the risks to the overall security posture. Therefore, architects should beware of capabilities that are dependent on particular host resources, such as operating system, network interfaces and protocols, and memory types. For instance, D/As are often faced with blanket requirements to install endpoint security software on every device, including smart light bulbs, cameras, or other Internet of Things (IoT)-type devices. Most endpoint security software cannot be installed on these types of devices, leaving them unable to satisfy the requirement. Some D/As faced with these requirements chose to remove these systems from their network, causing a significant impact to their mission performance. In these instances, the D/A would benefit from a capability that determines the actual risk of that device's security posture to the organization and its mission when considering a potential waver of the requirement.

While the world of IT and cybersecurity often creates open standards and protocols, companies still continue to create solutions that cause users to be "locked in" to a particular vendor's tools. D/As tout single-vendor implementations as being easier to integrate, providing one point of contact for support and maintenance, and having a familiar interface for usability. However, today's security architectures require integration amongst many varied capabilities to defend against advanced threats. Security architectures designed with loose coupling will ensure that the tools required to enable the necessary defenses are able to be replaced by multiple vendors and integrate with well-defined interfaces. Some Security Information and Event Manager (SIEM) tools provide a good example of both loosely coupled and tightly coupled integration. They can gather security event information from a variety of security and standard networking components utilizing common formats that standardize the data being sent by the other devices. In addition, they also offer useful capabilities to integrate with both firewalls and packet capture units, which allows for retrieval of the data related to the security event. While such capabilities offer time savings for a laborious task, these features are unique to certain brands of SIEMs, a particular subset of security tools. When the time comes to change the tools due to any number of factors (e.g., cost, end of life), users that are dependent on this useful functionality will have their mission disrupted by the change.

E. L5: Timeline and Budget

We learned a number of lessons from our analysis and engagement with D/As in the area of deploying security technologies over time. To start with, every established organization today already has legacy systems for both security and functionality; any new technology needs to be integrated with these systems. Large enterprises are known to have 70+ security products alone. We saw that new technologies would evolve and appear, based on both new threats, as well as evolving business and mission requirements. Every organization we talked to has specific timelines that dictate when they can buy, refresh, and upgrade their technologies.

We observed that distributed interoperability across services and endpoints that interact with each other needs to be initially established, and then continued through any changes in the technologies provided in the form of upgrades, patches, and other changes. A simple phased roll-out example, dictated by technology, is a tightly coupled client/server technology, such as a mail client and server. While it is recognized good practice to design changes across versions to work with (or at least not fail badly with) older versions, we note that new functionality that requires both client and server code is often best deployed on the server first, then across the clients. This minimizes end user confusion attempting to use a function not yet supported in the server. In addition, as we called out above, organizations have a specific time period for an update cycle. The scale of their deployment may mean that there can be too many clients to update in a compressed time period, due to resource constraints. We see that it is common that clients will be updated in phases, requiring the server to work with a variety of client versions.

In addition to periods dictated by technology and scale, we saw that organizations have specific processes, timelines, and cycles that can constrain roll-out timing on the calendar. An organization's pre-defined refresh/upgrade cycle may map to specific times on the calendar, and that can conflict with or delay roll-out plans. In that case, roll-out needs to be in sync with the existing refresh cycles. If an update plan was developed without a mapping to the refresh cycles, it will come at the cost of not adhering to the plan schedule. This can be a particular danger when a plan is made abstractly, based on the technology and number of systems involved alone, without customization for organizational schedule constraints. The lesson is that a phased technology roll-out should match an organization's timeline, both in terms of amount of time for each stage and overall calendar time for the staging.

Another complicating factor here is budget. The cost of for upgrading security technologies is often expected to be drawn from a fixed pool of funds set aside for upkeep and as such, it is subject to budget cuts. Moreover, budgets are allocated in cycles based on Program Objective Memorandum (POM) that covers the 5-year Future Year Defense Program (FYDP). This means that even if an ideal technology exits today to counter a specific threat, the budget for its roll-out cannot be worked into allocation plans for a few years. Continuing resolutions can further complicate this challenge and create additional roll-out delays.

We observed that in addition to scheduling, there may be other organizational constraints impacting roll-out timeline. Some organizations, such as large organizations or those with compliance requirements, have fairly extensive processes, where the "null task" for a roll-out can take more than 6 months. Anecdotally, we see that it takes 9-12 months for the most basic roll-out task to be executed within the compliance requirements of a government contract. Up to 6 months of that may be lead time, before any actual technology deployment starts. Smaller or more agile organizations may find it easier to deploy a technology in multiple rounds of small

simple updates. However, the processes for larger, compliance-constrained organizations penalize that approach. In some cases, breaking the roll-out into larger chunks makes more sense. In other cases, an organization will have a very specific organized maintenance window that needs to be met. A "whole shot" large deployment cannot fit in a single window, so spacing of the windows dictates the full-plan timeline.

One approach we see that works well with the many timeline and budgetary restrictions, is to design technology to provide value through out incremental deployments, in phases. The notion of a phased roll-out enables organizations to benefit from increased security immediately, rather than needing to wait until everything is ready. A useful initial phase of roll-out is a monitoring-only phase. This phase is available in some access control and intrusion prevention technologies. It allows an initial low-impact roll-out that reports what was detected or what would have been blocked. This allows the organization to see the potential value of the protections, as well as the potential costs, in terms of resource overhead or workflow impacts. One potential complication with this approach is that CISOs need to understand the benefit they will get from updates beyond a monitoring stage, particularly if they cannot make immediate operational use of the initial monitoring information. The value to the organization needs to be understood by the CISO up front. Subsequent additions can build on the monitoring-only phase, with approaches such as automated actions based on the monitoring.

Another approach that we see works well with staged roll-out is *loose coupling* of components. This is an architectural approach of some modern technologies, such as web services. A loosely-coupled architectural approach minimizes the amount of knowledge that each component needs about each other. Loose coupling is facilitated by a number of technical approaches. For example, minimizing the connections across components by having a many-to-one communication pattern instead of many-to-many, one can minimize the number of integration points that need to work across every roll-out. Relying on standards for interoperability and integration can provide flexibility across changes. Using data (which can be transformed or translated) instead of coded functions, for integrated communications, provides additional flexibility.

In summary, we learned that given the critical reliance organizations have on security technologies, their utility and functionality must be sustained over their full lifecycle. We see that security technologies need to accommodate and support gradual initial introduction and integration of components in phases, as well as changes to those technologies over time. CISOs need phased deployments that provide phased benefits to the organization and its mission.

VI. Take Aways

The major takeaway from our effort is that security is not just about what guarantees a security technology provides. Nuances involved in building an architecture and deploying technologies in large enterprises go far beyond the mere protection provided and are often factors that are not considered by technology researchers and developers. These nuances range from understanding hidden costs, ensuring compliance with standards, and understanding user impact of the technology, to tailoring a particular architecture/capability to its target environment and matching deployment phases with organizational acquisition/upgrade cycles. We encourage researchers and developers to consider these nuances in their work, with the hope that cutting-edge technology can make its way to practice much faster and more efficiently.

References

[1] S. Widup, M. Spitler, D. Hylender, and G. Bassett, "2018 Verizon data breach investigations report," Apr 2018.

[2] E. Nakashima, "New details emerge about 2014 Russian hack of the state department: It was 'hand to hand combat'," http://wapo.st/2otPiX8?tid=ss_tw&utm_term=.70b83c99827e, Apr 2017.

[3] C. Reform, *The OPM Data Breach: How the Government Jeopardized Our National Security for More Than a Generation*. CreateSpace Independent Publishing Platform, 2016. [Online]. Available: https://books.google.com/books?id=cOYRvgAACAAJ

[4] A. Sternstein, "DHS: Agencies reported 321 cases of potential ransomware," https://www.nextgov.com/cybersecurity/2016/03/dhs-agencies-reported-321-cases-potential-ransomware/127107/, March 2016.

[5] GAO, "DHS needs to enhance capabilities, improve planning, and support greater adoption of its national cybersecurity protection system," https://www.gao.gov/assets/680/674829.pdf, Jan 2016.

[6] National Institute of Standards and Technology, "FIPS General Information," https://www.nist.gov/itl/fips-general-information, Feb 2010.

[7] DISA, "Security Technical Implementation Guides (STIGs)," https://public.cyber.mil/stigs/.

[8] K. D. Uttecht, "Zero Trust (ZT) Concepts for Federal Government Architectures," MIT Lincoln Laboratory, Lexington, MA 2020, 2020.

[9] B. Osborn, J. McWilliams, B. Beyer, and M. Saltonstall, "Beyondcorp: Design to deployment at google," *;login:*, vol. 41, pp. 28–34, 2016. [Online]. Available: https://www.usenix.org/publications/login/spring2016/osborn

[10] M. W. Hoag, "An introduction to systems analysis," RAND Corporation, Santa Monica, CA, Tech. Rep., 1956.

[11] R. Ross, "Guide for conducting risk assessments," NIST 800-30 Rev 1, NIST, 2012.

[12] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975.

[13] R. E. Smith, "A contemporary look at saltzer and schroeder's 1975 design principles," *IEEE Security & Privacy*, vol. 10, no. 6, pp. 20–25, 2012.

[14] S. Garfinkel and H. R. Lipford, "Usable security: History, themes, and challenges," *Synthesis Lectures on Information Security, Privacy, and Trust*, vol. 5, no. 2, pp. 1–124, 2014.

[15] DHS, "DHS Binding Operational Directive 18-01," https://cyber.dhs.gov/bod/18-01/, Oct 2017.