Report of Discussions from Breakout Sessions

Annual Industry Workshop
March 27-29, 2017

Tempe Mission Palms
Tempe, Arizona

CREDC IS FUNDED BY THE U.S. DEPARTMENT OF ENERGY AND
THE U.S. DEPARTMENT OF HOMELAND SECURITY

Contents

About CREDC

The Cyber Resilient Energy Delivery Consortium (CREDC) is composed of ten universities and two national laboratories, led by the University of Illinois at Urbana-Champaign, conducting a variety of research activities in support of the cyber security and resiliency of energy delivery systems. Sponsored by the Department of Energy (DOE) and the Department of Homeland Security (DHS), CREDC follows from the earlier Trustworthy Cyber Infrastructure for the Power Grid (TCIPG) project. CREDC's research scope has expanded to encompass energy delivery systems outside the electric power sector, including oil and gas, as well as the complexities introduced by coupled energy infrastructures. CREDC conducts activities with anticipated deliverable prototype technology in an 18- to 24-month timeframe, as well as longer-term research that anticipates the impact of emerging disruptive technologies, such as big data and

cloud environments as well as the Industrial Internet of Things (I^2OT). The research is guided by an Industry Advisory Board and is often done in coordination with industry partners to maximize the beneficial impact of CREDC research on the sector.

The consortium includes researchers from Argonne National Laboratory, Arizona State University, Dartmouth College, the Massachusetts Institute of Technology, Old Dominion University, Oregon State University, the Pacific Northwest National Laboratory, Rutgers University, Tennessee State University, the University of Illinois at Urbana-Champaign, the University of Houston, and Washington State University.

Introduction

The 2017 CREDC Annual Industry Workshop was held March 27-29 at the Tempe Mission Palms in Tempe, Arizona. Consortium partner ASU served as the local host. The workshop was followed by an annual review meeting on the afternoon of March 29 which included members of the CREDC team, the Department of Energy, and some CREDC Industrial Advisory Board members. Erin Walsh of the Department of Homeland Security joined the annual review meeting via WebEx. The workshop featured a case study presentation by Ben Miller that analyzed the evolving Ukraine cyber-attack(s). The workshop program offered moderated breakout discussion sessions on topics that impact cybersecurity and resiliency of EDS. Three topics were offered over two sessions, allowing participants to engage in two topic discussions. The final day offered a session summarizing highlights from the breakout sessions. The program also featured an industry/research partnership showcase, CREDC research showcase presentations, lightning talk presentations, and a CREDC research poster session.

The industry workshop hosted 111 participants – 62 from the CREDC team and 49 from non-CREDC organizations. Six of our Industrial Advisory Board members were able to attend both the workshop and review meeting. The poster session featured 26 research activities.

Archives from the industry workshop are located online at:
http://go.illinois.edu/CREDCIW17CONTENT.

Breakout Discussion Section Summaries

Participants were asked to join two of three moderated discussion groups[1] on relevant topics impacting EDS cyber resiliency. These topics had been identified by the CREDC leadership while organizing the workshop. Each breakout discussion section was led by a moderator, and a scribe took notes during the discussion.

The session topics were:

- **Cyber Supply Chain Provenance and Protection**
  Session Chair: Dennis Gammel, Schweitzer Engineering Laboratories
- **Engineering Secure EDS**
  Session Chair: Zachary Tudor, Idaho National Laboratory

[1] Workshop participants chose breakout sessions of interest to themselves and were promised that no comments or opinions they expressed would be attributed to them or their organizations.

- **PKI in Current and Emerging EDS**
  Session Chair: Sean W. Smith, Dartmouth College

The following is a summary of the discussions in the breakout groups, edited from notes taken by session scribes.

Cyber Supply Chain Provenance and Protection

Problem Statement

Product development proceeds through life cycle stages as illustrated in the figure below. It may be vertically integrated (a single entity controls all stages) or may involve different teams, subassemblies, firmware, and other sub-components imported to the process at each stage. In practice, most development processes fall somewhere in between. Other aspects to consider include personnel, complexity and cost, and crossover technology (the risk of bringing in a component not designed for the domain of application of the product development in question).

Assuring the cyber supply chain is particularly challenging because of the use of software and firmware libraries as well as compilers and operating systems not typically under the control of the product developer. We may consider the TLS Heartbleed bug (heartbleed.com) as an example of a supply-chain vulnerability introduced at the research stage. Stuxnet included various supply chain components, notably the fact that the code was signed with a stolen digital certificate (https://www.wired.com/2011/07/how-digital-detectives-deciphered-stuxnet/).

Supply chain risks can arise from diverse issues including environmental, economic, poor communication, unreliable delivery, inconsistency, political instability, and obsolescence. In addition, one must consider interdiction, counterfeit components, and covert functionality. The latter is of particular concern with respect to cyber, because a component may pass all functionality tests, but establishing that it contains no additional logic is in practice very difficult.

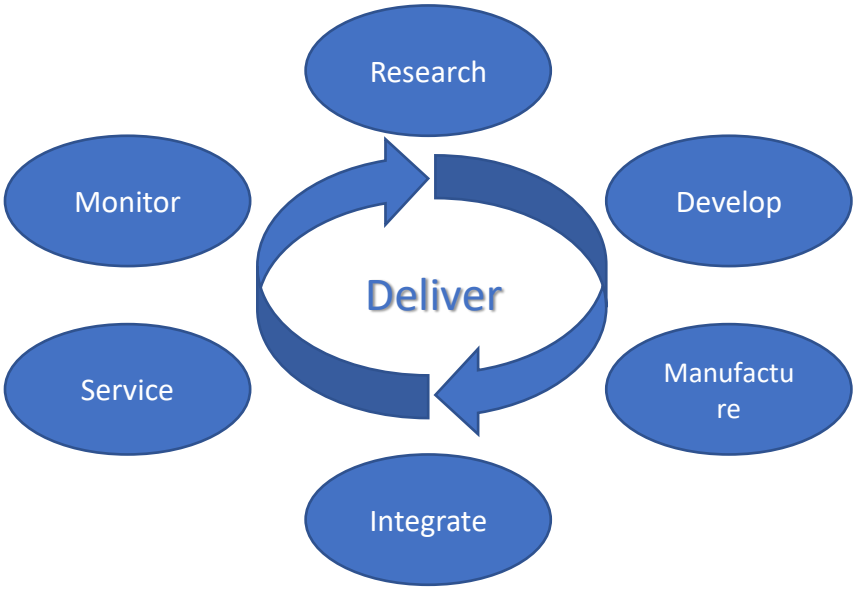

Figure 1. Product Development Life Cycle (Courtesy SEL Inc.)

Approaches

In order to assure the correctness of components from the supply chain, at least to an acceptable level of confidence, the developer pursues various strategies. The developer may assess the supply chain by

- Evaluating suppliers, considering reputation, documentation of features, and development process,
- Assessing products, by tracking products or through certification,
- Assessing chain of custody, considering supply chain length, trust in personnel, delivery time, and packaging. This may include a software registry to track software provenance, although such a registry may be useful to hackers as well.

The buyer may also test delivered components, seeking to assure that there is no unexpected functionality. As noted above, this is difficult. The area of cost-effective testing for component assurance still contains a number of research questions.

A possible approach is verification against a "golden image" of some fraction of the delivered components, possibly at the level of the unified extensible firmware interface (UEFI). Static analysis and positive-negative testing is also employed. There is work funded by DOE and other organizations examining chip security (including debug channels and JTAG) and binaries. However, this does not scale to large systems, such as an EMS that might include on the order of 25M lines of code.

Reverse engineering is also used. A customer may lift the cover on a device and search for information about identifiable components contained within. This is also what an attacker might do. The usefulness of this approach is lessened in the case that a manufacturer includes commodity sub-components such as memory chips that may come from different suppliers, so that different copies of a particular device may differ under the hood. In fact, a manufacturer may choose to do this in order to achieve equivalent functionality through diversity of implementation, mitigating the risk from any individual supply chain. But truly equivalent functionality using sub-components from different suppliers may be difficult to achieve at integration.

We note that vendor IP considerations work against reverse engineering approaches.

The developer may rely on certification by an external organization. There are existing standards, such as IEC 62443. It is difficult and expensive to test at high evaluation assurance levels (EAL), and there is no universally agreed best practice to do so. Furthermore, there is the issue of modification to the component. Does the developer repeat the process, or allow the supplier to "self-attest" to "minor" changes?

Risk Transfer

In a sense, supply chain management is an exercise in risk transfer. The developer seeks to reduce complexity by removing un-needed functionality (not always possible in practice).

As we go further in the supply chain, we are adding more and more complexity. The resulting life cycle is a mix of hundreds of individual product life cycles and thousands of humans involved.

- How do we eliminate of some of this complexity?
- Should we look at more standard to have better consistency?

There are different perspectives between vendors and buyers. The buyer is concerned with secure, correct functionality, while the vendor is incentivized to add features to maximize profit. This tension may be addressed through carefully drafted procurement language that places liability on the supplier rather than on the owner-operator.

Much of the above is not unique to EDS or ICS, but the consequences are different from those of IT systems. A vulnerability in an EDS component, inserted from supply chain or by other means, cannot be patched as readily as is the case in IT. As such, the issue of disclosure is more problematic. If the vulnerability is disclosed, or if a patch is released that a hacker can reverse-engineer, it is likely that many systems in the field will remain at risk for a significant time interval.

Research questions

As noted above, assuring the absence of covert functionality in components with cyber capability is a research problem for which there are no general solutions at present.

The discussion group suggested research to define and develop the equivalent of “air bags and seat belts” in substation equipment. To a degree, this is present in safety systems that are part of ICS, commonly in oil and gas.

It was suggested that blockchain mechanisms might form the basis of solutions for tracking provenance.

Additional areas for applied supply chain research (what exists? what are the gaps?) may include:

- Physical integrated circuit or digital component signature analysis tools
- Simple, continuous firmware and settings verification without the firmware’s involvement
- Low cost chain of custody and delivery tracking
- Product and component signature and verification tools (build on NMAP?)
- “Dialing home” monitoring tools or security controls

Engineering Secure EDS

Problem Statement

The world is full of threats that we don’t understand and vulnerabilities that are not apparent, where maintaining compliance with cyber-security regulations and industry standards demands ever-increasing resources without any promise of successfully preventing or mitigating attacks, and where there are far more security and resilience tools and technologies available than can possibly by implemented by any one energy delivery entity.

The observation that “Just because a patch’s CVSS score is critical does not mean it is critical in *our* environment due to where it is deployed or what other mitigating controls might be in place” suggests that a different world view of how to achieve cyber-security and cyber-resilience is needed. Researchers are beginning to look at the problem from more systems-

engineering viewpoints, such as INL's consequence-based, cyber-informed engineering (CCE), and NIST's recent publication "Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems" (https://doi.org/10.6028/NIST.SP.800-160).

Early in the discussion we concluded that a better title for the session would be *Engineering Resilient Energy Delivery Systems* not only to reflect CREDC's title but also because the broader term "resilient" better reflects the approach taken in Systems Engineering. Systems Engineering is concerned with how to design, operate and maintain large systems so as to deliver *emergent properties.* Emergent properties are behaviors and characteristics of the system as a whole that are not properties of components or subsystems. As a very simple example, a chain is a system whose behavior is quite different from the properties of its individual links. Similarly a bridge has emergent properties that are not apparent in the concrete and steel from which it is built.

This breakout session was premised on the idea that security, and more broadly resilience, are emergent properties of energy delivery systems and that we could be taking a more systems-engineering like approach to achieving them. Such an approach would concentrate on the required resilience properties and how they are to be achieved and maintained during the design and operation of energy delivery systems. The properties and how they are achieved would be developed in the context of the mission of each system.

This topic supports Roadmap strategy "Sustain Security Improvements" and "Develop and Implement New Protective Measures to Reduce Risk."

Observations

Session participants made a number of observations about the overall idea (in addition to the important point about focusing on resilience and not merely security):

- The perspective is potentially transformational
- Better understanding of the fragility of EDS (how they do or can fail to deliver the required service) is essential to designing for resiliency
- The methodologies must incorporate all-hazards approaches
- An advantage of doing so is that controls and mitigations may serve more than one purpose
- Methodologies are needed not only for initial system design but also to ensure that systems maintain resilience as they evolve over their lifetimes
- At this stage in the development of systems engineering practice around resilience we should concentrate on identifying enabling tenets rather than developing restricting requirements
    - Develop "Ten Commandments" of resilient engineering

Barriers

These might be considered barriers, or they might be considered existing challenges that a systems engineering methodology for resiliency would help the industry address.

- Industry needs a motivating event
- Features or convenience go against security

- Efficiency goes counter to reliability and security, so how do you find a happy middle ground
- Cyber security is not an end point, it is something that we operate in
- It's impossible to take every risk off the table
- There are many technologies that increase security and resilience. Only a small fraction of the available and appropriate technologies can be used in any given system. How can we give designers tools to make good technology choices?
- Need good recovery mechanisms
- Moving from physical to cyber is difficult to grasp. The physical world is a bit easier to understand as the inject vector is physical proximity, not varied like cyber is
- Third party connections are essential, and they often cannot be decoupled/cut off for various reasons (support, warranty, etc.)
- Managing vendors is increasingly difficult and secure the connectivity to the system
- Consider the protection of the system from the operators of the system itself
- A methodology is needed for evaluating a system's resilience in relation to its deployment in a particular domain
- Missions can conflict
- Designing a system is a separate discipline from deploying it; maybe there needs to be two approaches (and they would need to be complementary)
- Power people use power tools for planning/operations; but there aren't any design tools that assist you in designing resilient systems based on particular constraints
- Designers' lack of appreciation/understanding of attack techniques
    - People tend to focus on known malware or known vulnerabilities rather than on the full range of techniques available to attackers to accomplish their end goals
    - Tactical vs strategic thinking causes more problems down the road

Research questions

At this early stage, the attendees thought that we should be asking ourselves "what tools and technologies need to be developed in order to support utilities in adopting a systems engineering approach to the design and operation of their cyber-security/cyber-resilience operations?"

The attendees also suggested an initial list of tenets or principles that could underlie a systems engineering methodology for resilience. Some of these, though desirable, are perhaps not achievable: what should be done instead? And some suggests component technologies that don't yet exist. Should resilient systems engineering take them on as research tasks?

- Control actions should be verified based on system state before acting
- Safety engineering constraints must be adhered to in order to have a secure EDS
- Isolate/segment trusted and untrusted components from each other
- The system should not be allowed to take an action that harms itself

- You must be able to trust the sensors
- Design systems so that unacceptable consequences are physically impossible

Summary Session Notes

A few additional points came up in the plenary summary session:

- As a practical matter, it is unclear where the responsibility for engineering resiliency into systems could lie. Especially for the power grid, where "the system" spans thousands of entities, who is in a position to demand resiliency and who has the technical wherewithal to create it?
- Having a design paradigm would be great; but real deployment of it would require trust and cooperation and coordination from the government to reconcile existing compliance requirements with what industry could achieve by proceeding down a resilience engineering and deployment path.
- It would be helpful to concentrate on creating composable systems where secured components could be used and built upon to have a secure systems of systems. For instance, having a secured operating system would allow application developers to have strong security assumptions about the underlying facilities. It is worth pointing out, however, that if resilience is indeed an emergent property as we have contended, then having secure/resilient components is insufficient to having secure/resilient systems.

PKI in Current and Emerging EDS

Motivation

This panel explored technology issues and requirements regarding securing remote communications in energy delivery systems.

Consider scenarios where a command or data measurement, critical to the functioning of the grid, is being sent over a channel that an adversary may be able to manipulate. For example:

- the adversary may eavesdrop on valid communications (although this was not considered a serious issue in EDS)
- the adversary may maliciously modify otherwise valid communications
- the adversary may forge communications outright
- the adversary may replay a valid communication in a different context (e.g. by replaying an old message, or by replaying a message intended for a different receiver)

The field of applied cryptography gives many tools to solve such problems; the sender and receiver may do various kinds of mathematical transformations on the messages in order to make such malicious actions essentially intractable. Typically, these tools take one of two approaches:

- in *symmetric key cryptography*, the sender and receiver must each share a secret key first.
- in *public key cryptography*, each entity can have a pair of keys---a public one and a private one---such that calculating the private key from the public key is believed to be essentially intractable.

Although symmetric cryptography usually involves much lighter-weight computation, the requirement of shared secrets can become burdensome in many scenarios.

- If a sender needs to talk to *N* different receivers, then the sender requires to keep track of *N* different secret keys. A population of *N* devices will need on the order of N^2 secrets.
- If either the sender or receiver have a security breach, then the secret key has been compromised and needs to be replaced.
- If the sender and receiver have not met before or are from different populations (e.g., a machine issued by organization *X* talking to one issued by organization *Y*) then the ceremonies involved to establish shared secrets may prove complicated.

As a consequence, public key schemes become attractive:

- Each entity need only know one secret---its own private key.
- Standard techniques then let an entity add security to communications even with an entity it has never met before.

Public key techniques also allow *non-repudiation*: the ability of B to later prove to C that A sent B a particular message. (Some participants felt non-repudiation may also prove important in EDS.)

PKI

However, for a population to use public key cryptography, we need to figure out how each entity can establish a public/private key pair, and how an entity can learn the public key of each of its communication partners. The glue that enables this all to happen has come to be known as *public key infrastructure (PKI).* In practice (e.g. SSL on the Web, or S/MIME email), PKI has come to be dominated by the X.509 family of standards and operations, involving myriad issues, including:

- *certificates*: statements, signed by a *certificate authority,* binding a public key to given entity
- *trust roots:* parties whom a party implicitly trusts as saying true things about certificates
- *trust paths*: a valid chain of certificates from a party's trust root to a certificate of interest. (If A wants to work with B, then A needs to discover a valid trust path from one of A's trust roots to B's certificate)
- *revocation:* the mechanisms involved in distributing the information that the assertion in an otherwise valid certificate no longer holds.
- *replacement:* certificates typically have an expiration date (in part to make revocation easier); cryptographic practice dictates that keys should also be replaced regularly.

The conventional wisdom held by many is that the X.509 approach quickly becomes intimidatingly complicated once it starts trying to handle these issues; the response held by some who have worked more closely in the field is that these issues must be handled somehow if the system is to work.

Initial Discussion

The panel started out with discussion on the how to fit the complexities of the X.509 approach to the needs of EDS.

One set of issues came from the logistics of operation and administration of a standard X.509 approach. Should an EDS organization bring in an external 3rd party to run the PKI? Should the organization instead try to run the PKI itself? Standard PKI would seem to require IT connectivity for time and revocation information---how does this reconcile with the EDS principle of isolation for OT?

Another set of issues came from exploring how the entity interaction requirements in EDS fit the flexibility in X.509 PKI. One one hand, EDS may avoid much complexity if the flexibility is not required, and all foreseeable EDS deployments can guarantee simplifying assumptions such as:

- *Trivial trust paths*: each entity only needs one trust root, and a communicating population need only have one CA
- *Trivial communication patterns*: such as hub and spoke (entities only talk to one central one) or along fixed and small neighbor sets.
- *Trivial revocation*: it's never needed, due to a combination of assertions seldom going bad, and limited damage from the use of revoked certificate.

However, will these simplifying assumptions always hold?

- Will an entity need to interact with an entity managed by a different organization?
- Will communication ever be many-to-many? Will things need to talk things they haven't seen before? What about mobile entities, such as electric cars?
- Will EDS want to exploit overlapping communication topologies? E.g., an electric car might have crypto identity to enable billing and tolls, but local EDS might want to also use that to help coordinate charging of cars in a neighborhood? A piece of substation equipment may have crypto identity to talk within the EDS, but also might have identity to enable communication with its vendor (e.g. for software updates and performance telemetry)?

Furthermore, X.509 typically only allows entities to have one identity, fairly static. Will EDS require more complicated notions, such as:

- "I am a device of type X from vendor Y, but installed at substation Z"
- "I have software component S patched to level N"

A final set of initial issues concerned whether standard X.509 would need to be re-engineered to accommodate the constraints of EDS.

- Will constrained EDS devices have sufficient entropy to generate strong keys? Will they have sufficient computational power for the advanced mathematical operations PKI can require?
- What about the tendency of EDS devices to live much longer than the shelf life of cryptography?

- Will constrained EDS communication channels have enough bandwidth for standard revocation and path discovery techniques? Will PKI introduce too much latency?

Lively Discussion

Lively discussion then ensued. Some dealt with the issues of legacy:

- How can PKI accommodate legacy EDS? In addition to the lifetime issues raised above, participants were concerned about the long life of legacy networks and also the short lives of security vendors. There was some debate about whether bump-in-the-wire approaches were a good idea; some participants felt a better approach would be to design future legacy EDS devices with sufficient headspace for future cryptographic needs.
- Long-lived EDS devices can change ownership. How will the PKI tools accommodate that?
- How can EDS accommodate legacy PKI? Will an EDS PKI always be independent, or will there be "requirement creep" such as making it interact with Web SSL? Another concern was raised about re-thinking PKI "best practices" for EDS---the example was given that if certificate renewal does not happen on time, legacy PKI practice would require rejecting the certificates, and shutting down the pipeline.

Other discussion dealt with looking at the interaction of EDS requirements with PKI flexibility.

- Who talks to whom? What about rare but predictable scenarios?
- With standard cryptographic tools, it can be easy to conflate identity with authorization. In EDS, will "identity" of a sender always be sufficient for a receiver to judge whether to act on a message, or will we need something more? The TCIP-era PhD work of Chris Masone analyzing how authorization worked in the MISO telephone transcripts from the 2003 blackout showed that identity PKI would not have sufficed.
- Is adversarial modification of remote communications a significant threat? Within the same room, some participants insisted that important actions were always carried out by rolling trucks rather than sending electronic communication; other participants said they were already using electronic communication to control critical operations.
- If important devices will always be behind a protected physical perimeter, then cryptographic protection of the communications can potentially end at the perimeter. However, will this always be the case? What about electric vehicles, or distributed energy resources, or smart industrial buildings?
- Will communications with smart home appliances ever be critical enough to grid stability to warrant concern?
- Can we reduce the risk of bad messages by making relying parties smarter? One approach might be to use the "secret weapon" of the physics underlying processes an EDS controls. For example, changing a relay's interface from "tell me what parameters to use" to "tell me which of these three pre-defined settings to use" might still accommodate all operational requirements but lessen the threat from a forged command.

Towards an Industrial Key Infrastructure

From the discussion, it was clear that there is a lot of churn in this space right now. It reminded the moderator of discussions in TCIP industry meetings circa 2005, when the question was asked "will you ever use the Internet?"

We concluded that we need to collect more usage scenarios---looking not just at the present but what will come along in the next decade---as we develop requirements for what some participants called *industrial key infrastructure*: the PKI glue necessary for securing critical communications in the specialized applications domains of EDS.

Funding Acknowledgement and Disclaimer

Acknowledgement

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000780.