Security in Your IoT Networks
Panel Discussion

Presented by **Enrique Herrera**

Market Principal – Connected Services

Feel free to add comments or questions to the panel.

When poll is active, respond at **PollEv.com/osisoftpolling** Text **OSISOFTPOLLING** to **22333** once to join

Andrew Ginter

Waterfall Security Solutions

VP Industrial Solutions

Eric Cosman

ARC Advisory

Contributing Consultant

Rod Schultz

Rubicon Labs

VP Product

Security in your IoT Networks
Questions & Answers

Q1. Benign instruments can be co-opted by overwriting the IOT device OS. How do we safely put instruments of any type on the big internet?

[Owen] IoT devices can reduce exposure to firmware overwrite first by making only outbound connections to the internet and second by implementing secure boot based on digital signature

[Cosman] The placing of instruments on the “big Internet” is inevitable.
The key is assigning the correct trust level to the information that they provide. It will less than that assigned to devices over which you have full control, or that are better protected.

[Ginter] The only safe assumption is that all Internet-exposed CPUs will eventually be compromised. Our system designs need to be acceptably safe in the face of worst-case compromise. Systems that cannot tolerate routine compromise cannot be directly connected to the Internet. Strong physical and network perimeter protection, will always be part of the most sensitive control systems, IIoT or not.

[Schultz] The obvious answer here is secure boot for the OS, but that’s much easier said than done. IoT devices just don’t have the resources available for robust security, focusing the system to be actively monitored to validate that a device has not gone rouge based upon its behavior.

Q2. How does IOT security fall into the traditional Isa 95 and 98 security definitions? How would it fit into a traditional level divisions of level 4 to level 0?

[Owen] Asset maintenance by service providers in level 4 is one of the early adopter roles for IoT although other ISA95/99 automation functional roles will eventually be supported. IoT implementation for safety critical (level 0) requires careful attention to implementation details to ensure safety is enhanced rather than degraded.

[Cosman] There are a variety opinions on this question. On one end, there is the philosophy that IOT (or IIoT) simply represents another conceptual level above ISA95 level 4. This seems to be the OSISoft approach. Other have expressed the opinion that IIoT will “blow up” the hierarchical approach to defining the reference model and system architecture. As with most debates of this nature, the final outcome will likely fall somewhere between these two positions.

[Ginter] The SP99 security level definitions reflect IT priorities - they focus on "preventing disclosure of information." Industrial sites generally care much more about preventing unauthorized operation of physical equipment than information disclosure. The latest IIoT security advice in the Industrial Internet Consortium (IIC) Security Framework recognizes this and talks about layers of IIoT gateways, including unidirectional gateways, as essential protection for the most sensitive IIoT equipment.

Q3. What criteria should be considered in deciding whether a device is a candidate for IoT?

[Owen] Primarily consider the use case and lifecycle. Are any inputs or outputs affecting critical systems? How long does the device need to be in service and how hard is it to update?

[Cosman] Agree that intended use is the determining factor.

[Ginter] The overriding questions are always costs and benefits. How great are the business benefits of greater connectivity for some kind of device? And are the costs acceptable, especially the costs and risks of increased attack surface? The third law of SCADA security says that all attacks are information, and every bit of information can be an attack.
This means attacks can pivot through compromised equipment, putting connected equipment on the same network at risk. This is what we mean when we talk about "increased attack surface." A single Internet-connected device on a critical network can put the entire network at risk.

[Schultz] A few things to ask:
1. What is the value of putting this device on a network, what do I gain?
2. What is the cost of putting this device on a network if it is compromised, what do I lose?
3. How do I respond to a breach of the system that this device is a part of if it is (or is not) connected to a network?

Q4. Who is responsible for ensuring security of edge devices - we know they can be used for large scale DDOS attacks. The device manufacturer? The device owner? Governments?

[Owen] Owner of public IP address associated with the edge device is the 'defacto' responsible entity, however enforcement of policy is currently very lax – the internet is still the 'wild west'.

[Ginter] DDOS attacks are inevitable from at least consumer-grade devices. Some jurisdictions may implement penalties for owners of compromised devices or IP addresses, but others will not. My own guess is that eventually, vigilantes who tire of their favorite sites being targeted will exploit the same vulnerabilities that botnet owners are starting to use. We will see consumer devices routinely bricked to reduce the size of DDOS bot-nets.

[Schultz] This is going to be dependent upon the device, its functionality, its role, and where it sits in the network. Incentives and disincentives will need to be established to motivate anyone who is trying to move low margin products onto a network, but the responsibility will eventually fall to the device manufactures and the providers of the specific SW and HW.

Q5. What advice would you give to customer who want to TEST the security of their environments both wired and wireless?

[Owen] Integrate testing with security operations such as red-blue exercises that target defensive capability like discovery of authorized and unauthorized: devices, communications, and applications. Consider leveraging external scanning tools like Bitsight and Shodan.

[Cosman] I agree with this advice, assuming that there is already an accurate inventory or description of the assets in place. My experience is that is often not the case. You have to confident in what you have before you can secure it, or test your response.

[Ginter] Test to what degree? Nothing is secure. Before testing anything, we need to set the bar. For most control systems, I suggest the bar should be this high: no attacker sipping coffee on another continent should have any chance of compromising important control systems. This means deploying the strongest network and physical perimeter protections. No network attack should have any hope of succeeding. No compromised file should have any hope of arriving inside the physical perimeter on a USB drive or laptop. I see programs and systems this strong deployed routinely, but not nearly universally. The motto of the University of Calgary, my alma mater, translates to "I will lift up my eyes". On average, we need to set our sights higher.

[Schultz] I'd recommend that they hire at least two different teams to do penetration testing of their environment. By no means is that enough, but it will at least expose the low hanging security challenges the need to be addressed.

Q6. Do you see firewall and secure tunneling protocols built into smart instruments in the future?

[Owen] Yes, including an industry transition to purpose built protocols like Secure SCADA Protocol (SSP21).

[Cosman] Yes.

[Ginter] Many kinds of security technologies can improve IIoT security for many classes of devices. A good survey is available in the IIC Security Framework document. However, I encourage buyers to regard all claims about "secure communications," "secure tunneling protocols," and everything else "secure" with deep suspicion. The first law of SCADA security says "nothing is secure." Everyone using "secure" as an adjective is either selling something, or has just bought something. I'm a vendor, I would know. I use "secure" as an adjective from time to time. See if you can catch me at it.

[Schultz] That really depends on the resources available on the device. If we’re talking about low powered MCUs, then I expect a lot of that to be handled by the gateways, with the hope of a security umbrella effect happening for what’s below that gateway. To securely tunnel data to a device you’ve got to do a secure key exchange with bidirectional authentication. That’s going to be very challenging for low powered device if you are options for PKI.

Q7. Can you comment on best practices for authentication and authorization related to accessing applications/data running on edge devices. Certificates vs. Tokens?

[Owen] Applications on edge devices should initiate connections to a service providers and take advantage of advanced security offerings. Device identity (certificates or tokens) should be anchored in hardware such as a TPM.

[Ginter] For equipment directly connected to the Internet, vendors seem to be using PKI pretty consistently. The same is true for most connectivity within high-level networks in plants. Shared-secret keys are easier to set up and manage for star-connectivity device networks an plant LANs. For everything else, the jury is still out.

Q8. Standardization of security assurance to customer?

[Owen] Existing software assurance regimens have yet to be proven effective for markets at scale; we should focus on baseline cyber security metrics that are simple for customers.

[Cosman] Agree, but keep an eye on those assurance and compliance programs.
They will be proven in use.

[Ginter] Lots of people talk about certifying devices, and not enough about securing systems. Modern, targeted attacks target systems, not devices. Systems permissions are easier to exploit than vulnerabilities.
It is not primarily product vendors who need to certify security to owners and operators, but systems integrators designing systems.

Q9. How to balance between providing security related information about a system for assurance and the risk of disclosing too much information to make hacker's life easier?

[Owen] Provide extra safeguards (ISAC, NDA, etc) on information that is *not* actionable by defenders. Actionable information for defenders should be provided with minimal friction to help ‘level the playing field’.

[Cosman] Provisions must be made to protect the information shared [and with whom].

[Ginter] People criticize "security through obscurity." The one place such security is indicated though, is when protecting the detailed designs of our most sensitive systems - IP addresses, software versions, security software, and so on. There is no need to make our attackers' lives easier by disclosing this information to them. We also need to be confident that our vendors and auditors protect these sensitive designs as thoroughly as we do ourselves.

[Schultz] This is a very difficult balancing act. Rather than provide the technical details that could potentially teach an attacker how to attack your system, you may be able to provide penetration test results from a vendor for specific threat models (the customer is concerned about data in transit, data at rest, when a debugger is attached…). Clearly there needs to be trust between the penetration test vendor and the customer, but that will allow the vulnerable technical details of your product to remain hidden. No system is 100% secure, so agreeing to that up front is also a good first step.

Q10. How does a vendor provide security assurance when there is no standard certification for security for any particular industry?

[Owen] Warranty programs often fill gaps in certification approaches, for example automobiles have many certified and tested components but product warranty is the primary assurance.

[Cosman] (Independent) Certification and compliance programs are beginning to emerge, and we can expect them to gain acceptance and improve in reach and capability. Independence is the key.

[Ginter] Again, assembling a bunch of certified-secure devices into an insecure configuration defeats all of the vendor certifications. On the other hand, strong systems design can yield very secure systems, even when many components are not up to the latest standards. Too many people are talking about vendor & product certification without any sort of context in terms of secure systems design. And the person submitting the question has it right - designing systems for security is very much industry-specific.

[Schultz] Penetration test results coupled with an ability to update, revoke, and renew a breached system.

Q11. Internet facing security technology often require frequent security updates including software patches and password changes or key regeneration. How will we handle this with embedded devices that can't be easily updated?

[Owen] It's fair to expect increasing security capability from embedded devices due to lower cost of technology (Moore's Law). Security servicing is extra effort so there will always be a threshold of low end devices without support.

[Cosman] Even embedded devices can be modified, upgraded or patched as long as they are designed to support this. In cases where this is not possible the only real alternative is the use of separate mitigating controls.

[Ginter] The second law of SCADA security - all software can be hacked.
Internet-exposed devices that cannot be updated are no better than consumer-grade devices. They will be compromised. I'm guessing they will eventually be bricked. For devices on more-protected networks, protect the network/system and we can tolerate less-secure devices on those networks.

[Schultz] If you can't easily update a device, then you should probably think twice about putting on the network. Adaptable security will need to be designed into these device to abstract this challenge, but updatability is critical to restore the security of a system.

Q12. When choosing tools/languages for development or Operating systems for deployment, are some more secure than others? Which ones are inherently insecure?

[Owen] Native code languages (C, C++) are prone to memory corruption issues and should only be selected for development projects when there is a compelling reason.

[Ginter] Again, all software can be hacked. Even Java applications have vulnerabilities, and so does the Java runtime. Control systems must be designed to avoid the most unacceptable safety and reliability consequences, even when Internet-exposed equipment is inevitably hacked. That said, operating systems that make heavy use of Java apps do seem to have fewer vulnerabilities than average. Choosing such an OS could result in less effort applying security updates.

[Schultz] A few suggestions:
C11 to help with proper memory management. The less code you have the better, so be very careful what features you decide to build into your software. Any OS that provides a sandbox is great, but that will generally prevent you from targeting smaller devices. Work directly with your MCU provider to understand the secondary security features their HW has, how to use it, and what the tradeoffs are.

Q13. Does software containerization help? There are no edges.

[Owen] Virtualization and container technology can help achieve security policy enforcement and rapid recovery objectives.

[Cosman] Agree.

[Ginter] Containerization helps, but all software can be hacked. We need hardware-based containers for our most important control systems.

Contact Information

Brian Bostwick

Brian@OSIsoft.com

Cyber Security Market Principal

OSIsoft, LLC

Bryan Owen PE

bryan@OSIsoft.com

Principal Cyber Security Manager

OSIsoft, LLC

감사합니다

谢谢

Danke

Merci

Gracias

Thank You

ありがとう

Спасибо

Obrigado