Protecting Critical Infrastructure

Recently, we have seen attacks against critical infrastructure around the globe. Triton, for example, infiltrated the critical safety systems of an energy plant. While the financial toll of cyber-attacks has become an unfortunate norm. Attacks that can damage connected systems, and put human lives and property at risk are emerging with new regularity. Microsoft and industry partners can together mitigate such attacks by using trusted execution environments such as Azure Confidential Computing, Intel Software Guard Extensions, ARM TrustZone, and SecureElements.

In addition to threats covered in typical security analyses, new ones must also be mitigated for critical infrastructure:

Malware that has already compromised a normally-trusted device.

Rogue third-party administrators with access to the hosting system or software, but without operating rights.

Foreign state actors with legal influence over cloud, software, and hardware vendors in their jurisdiction.

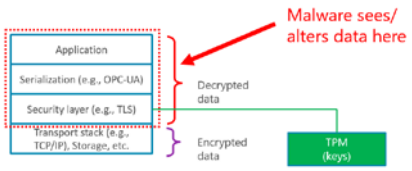
The key security principle is:

The device owner/operator must be in complete control of critical systems.

Specifically, this means that critical systems cannot be controlled by Microsoft or any other cloud, OS or application provider or foreign government in whose jurisdiction they may reside.

Protecting Data During Execution

One common misconception is that it is sufficient to protect data in flight and data at rest. This leaves a vulnerability in *data in execution*.



Data in execution can be protected by Trusted Execution Environments (TEEs) such as Intel SGX, ARM TrustZone, and SecureElements.

Key Points:

- Critical Infrastructure necessitates strict prevention measures, rather than just relying on detection and remediation.
- Trusted Execution Environments provide hardware assurance against malware.
- All hardware parts needed are readily available today.
- Open source code and industry standard protocols provide transparency and ability to vet critical operations and software used.
- Solutions can be deployed without requiring changes to existing equipment.
- Microsoft's solution not require placing implicit trust in Microsoft or any other software or cloud vendor.

A TEE is any hardware that enforces three guarantees:

1) The device has a unique security identity
2) Any code inside the TEE is operator-authorized code
3) Any data inside the TEE cannot be read by code outside the TEE

© Microsoft 2018

Commented [JF1]: What if we pivor to: "Never Lose Control of your Infrastructure"?

Commented [HG2]: Sentence structure: "damage [..] are emerging". Can damage emerge? Otherwise, restructure.

Commented [HG3]: "Intel Software Guard Extensions"

Commented [HG4]: "ARM TrustZone"?

Commented [AZ6]: "new"

Commented [DT8]: Don't imply we can defend against DOS. Probably need to mention DOS explicitly.

Commented [JF9R8]: This can go to the whitepaper

Commented [AZ10]: Who is "they" in this context?

Should be "critical systems"

Commented [HG12]: Isn't this CyReP-specific?

Commented [DT13R12]: No

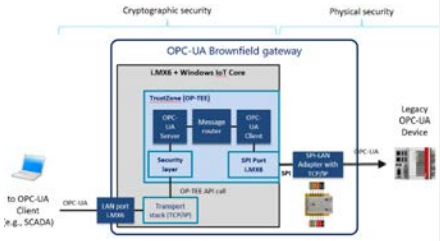
End-to-end scenarios often involve many components, including human interaction devices (e.g., SCADA systems), public or private cloud services, edge compute devices such as gateways, controllers connected to field busses, etc. *All* devices that could potentially control critical systems must be protected with trusted execution.

Secure I/O

Ultimately however, there is actuator/sensor functionality that is controlled electrically and physical protection is used to prevent tampering of such connections. If malware can access such connections, such as via a kernel compromise, then vulnerabilities exist. To defend against them, it is necessary to treat such connections as *trusted peripherals* that can only be accessed from within a trusted execution environment. TEEs such as TrustZone on an iMX.6 can do this today.

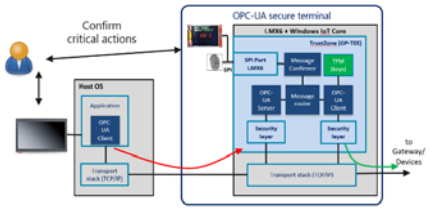
It is impractical, however, to require replacing expensive equipment to put in hardware security chips, since such equipment often has a lifespan of 10 or more years. Fortunately, a "brownfield" deployment is possible by putting a gateway with a TEE in front of such equipment and protecting the connections between the security gateway and the existing equipment in the same way as the connections to the physical actuators are secured: physical security. The key requirement for such a gateway is that communication to the existing equipment must *only* be possible from within the trusted execution environment, not from any networking stack in the OS.

Many variations are possible, but one example is shown below.



OPC-UA Security Gateway with Trusted I/O

The gateway above can defend ~~against~~ attacks against the equipment behind the gateway, but by itself is not sufficient to defend against a compromised user device or SCADA system that can send commands that appear as legitimate. For example, a compromised device could trick an operator into performing the wrong action, or simply initiate an action without informing the operator. Pairing it with a Secure Terminal that requires confirmation of unusual critical actions can defend against such attacks. Again, many variations are possible, but one example is shown below.



Secure Confirmation Terminal

In this diagram, commands go through a gateway that checks each one against policy. Any commands that policy deems as requiring confirmation result in a message on a secure display and a way for the operator to confirm the intended operation, where the screen and input device are *only* accessible from the TEE, out of reach of any malware. Once confirmed, the commands are signed by a key in the TEE, which key is authenticated by the gateway or server before commands are accepted.

Confidential Cloud Computing

© Microsoft 2018

Finally, it is important to protect any public or private cloud services that could be used to directly or indirectly control critical operations. Examples include: provisioning services, key management services, certificate authorities, patch management servers, and logging services. Such services must not only use secure protocols, and protect keys and data at rest, they must do all critical operations in a trusted execution environment that is protected from public cloud hosters and OS vendors. **Azure Confidential Computing** enables cloud hosting of these services.

Why work with Microsoft on Securing your Critical Infrastructure?

- 25 years of embedded experience
- High-Quality Software Development assets to be inspectable through the Code Center Premium program
- Windows Device Update Center enables OEMs to use the global CDN of Windows Update to cloud control updates and deliver OEM-specific files
- Windows 10 IoT Core with i.MX6 includes CyReP ready firmware with Trusted I/O support
- Azure Confidential Computing brings Trusted Execution to the Cloud
- Cross-Platform and Industry Standards based solutions
- Placeholder: Satya says you control of your data
- Placeholder: Microsoft is smart in security

If you're interested in working with Microsoft on this project to enable critical infrastructure protection through your company, standards organization, or as a research project, we'd love to hear from you.
Contact: <Need to put some contact info here. aka.ms, etc…>

© Microsoft 2018