Gen-Z Component Authentication:
Foundation for a Secured Infrastructure

Nigel Edwards, Theo Koulouris, Michael Krause
{nigel.edwards, theo.koulouris, mkrause}@hpe.com

January 22, 2019 the US Cybersecurity and Infrastructure Security Agency issued an emergency directive to mitigate DNS infrastructure tampering intended to disrupt and redirect government and business communications. August 21, 2018 Microsoft removed multiple websites allegedly created by the hacking group Fancy Bear to influence the US Midterm elections (Fancy Bear is allegedly responsible for numerous cyberattacks such as deploying an UEFI rootkit attack to subvert systems without Secure Boot protection in order to exfiltrate, i.e., steal, data). August 16, 2017 Maersk reported that the NotPetya cyberattack could cost $300M in lost revenue in addition to costing multiple executives their jobs. In 2016 the revelation that millions of Yahoo user accounts were hacked cut $350M from Verizon's Yahoo acquisition price. The December 19, 2013 Target retail store data breach cost $252M and Target's CEO his job. These are just a sample of an ever growing list of cyberattacks used to perform data and intellectual property theft, conduct cyber warfare, and to extort anyone and everyone through service and infrastructure disruptions.

The first step to building a resilient and secured infrastructure is to assume that every component is an attack vector, i.e., can be used to subvert, disrupt, deny, and destroy physical infrastructure and services, exfiltrate data, extort money, or coerce action. "Every component" means just that—every smart phone, tablet, PC, server, switch / router, USB device, processor, memory/storage/IO module, power and cooling units, firmware, IoT device, vehicle, etc. A component attack can be mounted by counterfeit component substitution through the supply chain, where components are intercepted and replaced or tampered with during transit, system assembly, or post-deployment replacement. A component attack can be mounted by compromising embedded firmware to create persistent backdoors, silently exfiltrate information, and render system firmware and operating system defenses ineffective.

How then does one assess component authenticity? How does one know what components are inside of enclosure and if they are genuine? How does one know that a system that has been in service for months or years and has been subject to periodic maintenance by internal or external staff not been tampered with or replaced? Though a composable infrastructure can provide superior CAPEX / OPEX savings, how can customers know that all dynamically-composed components are authentic? In most cases it is impossible to reliably answer any of these questions short of visual inspection which requires taking portions of infrastructure offline in the hope that a person or scanner can accurately detect discrepancies. Such undertakings are often unreliable and impractical for most and especially for anything at scale such as an enterprise or cloud data center.

The only practical and cost-effective approach is to build strong component authentication and firmware measurement capabilities into every component. The authentication architecture presented in this paper enables systematic component authentication from the point of manufacture through in-field service. The robust authentication mechanisms described use strong cryptographic principles to detect counterfeit or compromised components and to verify firmware and component configuration. In addition, component authentication can be used to establish the component-to-component trust relationships required to exchange secrets and establish encrypted communications between components to ensure confidentiality and data integrity.

The Challenge

Figure 1 shows a typical compute node containing multiple components communicating through multiple interconnect types including: PCIe, I2C / I3C Basic, SPI, DDR, USB, Gen-Z, etc. To avoid cost and complexity, the Gen-Z Consortium along with multiple other industry standards bodies have decided to use forward-looking industry standard security data objects developed within the Distributed Management Task Force (DMTF). These data objects and protocols are compatible with the high-volume foundation established by the USB Type-C Authentication Specification (https://www.usb.org) created to authenticate USB devices and power supplies. Any interconnect, including all those illustrated in *Figure 1*, can be used to transport these security data objects for authentication, verification, and key-exchange.

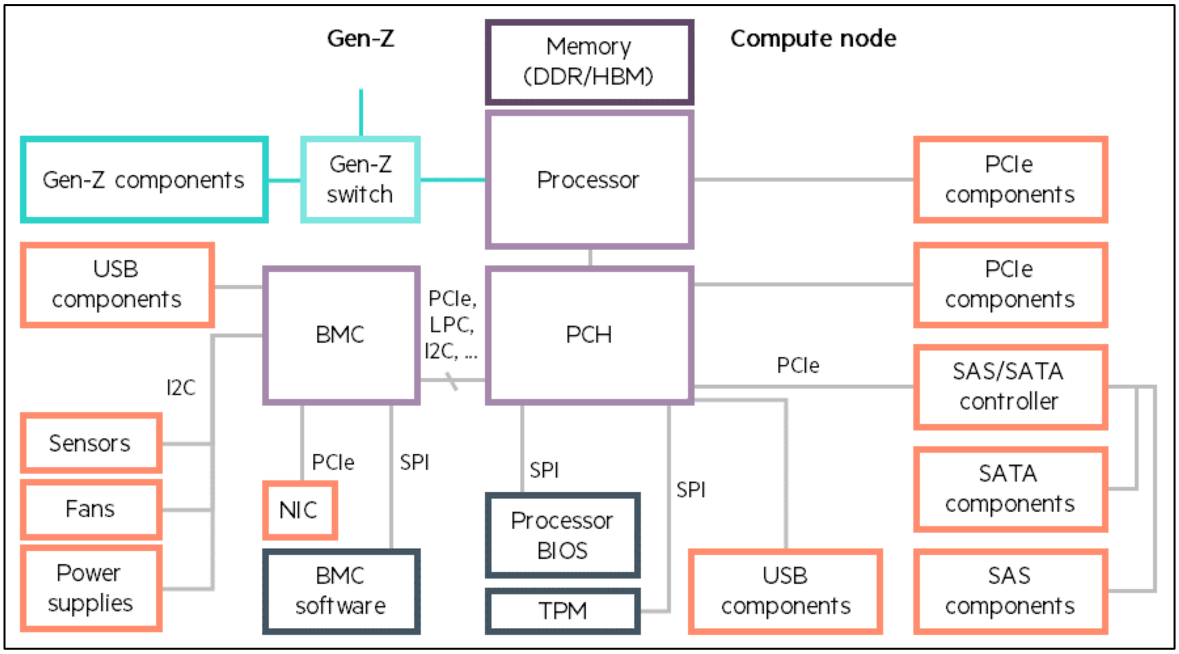

Figure 1: Simple Compute Node

To effectively validate authenticity, component authentication needs to support the five phases in a component's lifecycle:

1. **Component Manufacturing**: Each component needs to be authenticatable at any time during the manufacturing process by an appropriate test device. This allows the authenticity of a component (including all parts included in its assembly) to be verified at any point while in the supply chain.
2. **Component Integration:** Prior to installing a component in an enclosure or an enclosure into a rack or larger enclosure, the integrator or manufacturer needs to verify its authenticity.
3. **Initialization and Power Cycle Events**: Authentication needs to be performed whenever a component or enclosure is initialized or power cycled.
4. **Runtime**: Authentication needs to be performed on-demand to support application or customer-specific validation prior to or during application or service operation. Further, authentication needs to be initiated whenever a component exits a deep low-power state.
5. **Component Addition or Replacement**: Authentication needs to be performed whenever a component is added or replaced.

Component authentication via industry standard security data objects enables vendors and customers to verify that every component is genuine and from a trusted manufacturer. In addition, security data objects enable the secure measurement of embedded firmware and component configuration to detect if a component has been compromised.

Secure data objects can support cryptographic key exchange to enable reliable and private data exchanges between components, preventing a malicious actor from capturing or changing sensitive data.

Component Authentication Steps

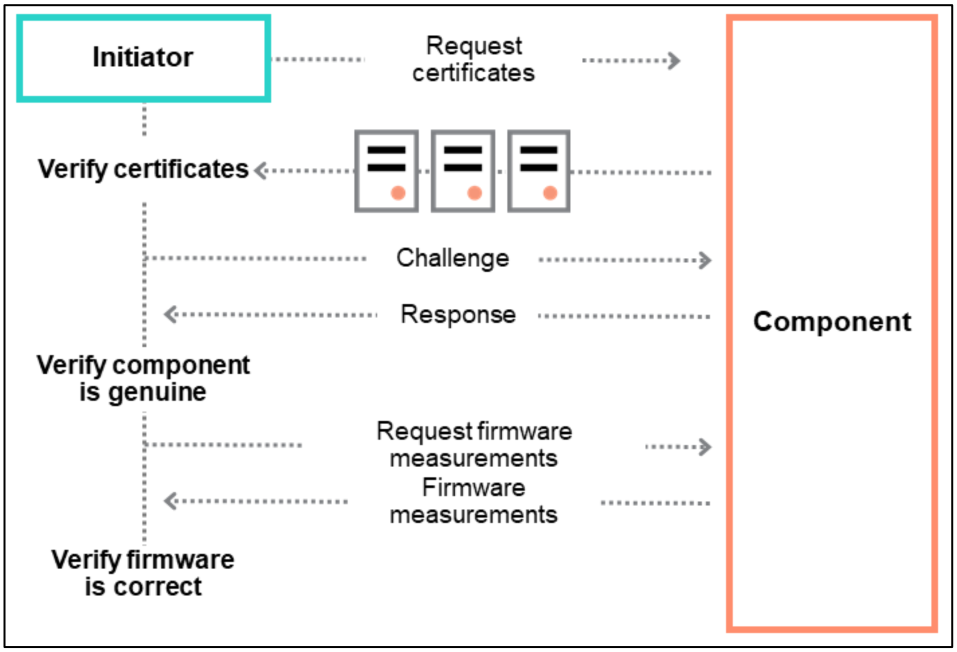

Figure 2: High-level View of the Authentication Protocol

Authentication is built on a well-understood and proven pattern based on the exchange of industry standard security data objects as illustrated in *Figure 2*. Every component is provisioned with a unique identity in the form of a cryptographic key pair and corresponding certificate hierarchy, rooted in a trusted Certification Authority (CA). In the simplest case the component manufacturer is the CA and issues the component a certificate. For brevity we refer to the entity authenticating the component as the authentication initiator. The process begins with the authentication initiator verifying that the component's certificate chain is correct (a component can support up to 8 certificate chains, enabling authentication and certification as a component moves through the manufacturing and assembly process). The authentication initiator then challenges the component to prove its possession of the private key associated with the now trusted public key. The challenge utilizes a single-use random number or *nonce*. The component signs the nonce with its private key, and the initiator validates the signature using the public key, confirming that the responder is in possession of the private key. Correctly responding to this challenge means that the component is genuine and from a trusted manufacturer.

Firmware and Configuration Measurement

Once a component has proven its identity and provenance via the authentication process, the authentication initiator can request from the component a set of measurements of its firmware and configuration objects. These measurements can be compared to a database of validated measurements to assess whether or not the correct firmware is installed and that the component is correctly configured. If a measurement is determined to be incorrect, this indicates that an update is required, that malware has been introduced, or that the component is incorrectly configured.

A measurement is essentially the cryptographic hash of a component's firmware binary image or configuration object. Components report their firmware and configuration measurements in authenticated data structures whose contents are signed with the component's private key.

When an authentication initiator requests the measurements from a component, it provides a nonce that must be included in the response. This validates the integrity and freshness of reported measurements.

Manufacturer Impacts

Component authentication requires components to be provisioned with credentials, i.e., keys and certificates, as part of the manufacturing process. Though this architecture was designed with a high-volume, low-cost mindset, its inclusion in any component type will add modest incremental cost and complexity to component manufacturing and customer deployment. However, given the broad industry acceptance of USB Type-C Authentication, as well as the potential cost, brand damage, job loss, or even physical harm to infrastructure or human beings from a successful cyberattack, the benefits far exceed the incremental cost and complexity.

Certification by Third Parties

The component authentication architecture is designed to be adaptable. Certificates may be issued to components by a third party. For example a third party can provide a service at a secure location, perhaps under government supervision, involving rigorous extensive testing and scanning of the component to verify that it behaves as specified and does not contain malware. Upon successful authentication, the third party can install a new certificate chain into the component with a leaf certificate certifying the component. An authentication initiator can use this certificate chain, rather than or in addition to the ones issued by the original manufacturer.

Gen-Z Security

Component authentication complements and enhances Gen-Z's extensive set of other security features. A key one is built-in hardware-enforced isolation (component-to-component, switch-based packet filtering, peer nonce, page-level access control, etc.). Others include, Hashed Message Authentication Code (HMAC) packet authentication and anti-replay attack protection, and data plane and control plane packet encryption capabilities at the page level for switch topologies and at the component-level for point-to-point topologies.

Conclusion

Gen-Z Consortium members are working within the industry and with multiple industry standards bodies to develop component authentication into an open standard and architecture, covering a wide range of components. Component authentication provides customers with assurance that every hardware component including embedded software making up their information systems is genuine, untampered with and trustworthy. It safeguards the customer's business. It safeguards the supplier's reputation and revenue-stream against counterfeits and attacks compromising their hardware. It raises the bar against both state and criminal malefactors. The time has come for the industry to work together to provide component authentication.