The ISP Column
A column on things Internet

October 2023
Geoff Huston

Notes from NANOG 89: Trust and Network Infrastructure

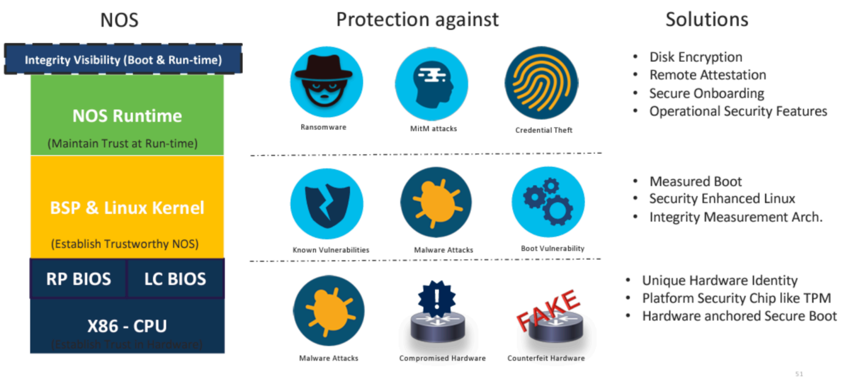
Trust is such a difficult concept in any context, and certainly computer networks are no exception. How can you be assured that your network infrastructure us running on authentic platforms, both hardware and software, and its operation has not been compromised in any way? The combination of complex supply chains and the potential of a skilled adversary to inject compromised elements into these supply chains requires a comprehensive view of how to protect the integrity of network platforms, and this was the theme of a presentation by Cisco's Rakesh Kandula at NANOG 89.

The challenges of securing the network infrastructure of many platforms increase as the networks increase in size and scope, as the number of remote and/or unattended locations increase, and the tolerance for interruptions reduces, particularly when the network is used used to support the operation of critical functions. If you take a comprehensive view of how platforms can be compromised, then the story starts with hardware and the issue of corruption of supply chains for processor chips and ASICs. Counterfeit and tampered hardware has been seen in resale markets, and if they get used in sensitive contexts the entire service platform can potentially be compromised. These days it's prudent to ensure that all hardware is given a unique cryptographically protected identity, and all onboarding processes include checking the authenticity of devices. Obviously, the identity needs to be tamper-proof, and the check needs to be undertaken in a secure manner.

Authentic hardware does not necessarily prevent all forms of potential interference in the boot process, resulting in the insertion of corrupted firmware or operating system software into the device. This can be achieved by altering the boot interface, or booting from an alternate device, or forcing the device to boot from an older version of the code that contains known exploits.

The operating system needs to apply protections to the runtime environment for the various applications that are running ion the device. There are operating system environments that assist in this, such as the Secure Environment for Linux (SELinux) project, which is a security enhancement to Linux which allows users and administrators more control over access control and filesystem operations. Access can be constrained on a level of granularity of users and applications over who can access which resources and under what circumstances.

The overall generic picture of securing network infrastructure platform is shown in Figure 1.



NOS

Integrity Visibility (Boot & Run-time)

NOS Runtime

(Maintain Trust at Run-time)

BSP & Linux Kernel

(Establish Trustworthy NOS)

RP BIOS

LC BIOS

X86 - CPU

(Establish Trust in Hardware)

Protection against

Ransomware

MitM attacks

Credential Theft

Known Vulnerabilities

Malware Attacks

Boot Vulnerability

Malware Attacks

Compromised Hardware

Counterfeit Hardware

Solutions

- Disk Encryption
- Remote Attestation
- Secure Onboarding
- Operational Security Features

- Measured Boot
- Security Enhanced Linux
- Integrity Measurement Arch.

- Unique Hardware Identity
- Platform Security Chip like TPM
- Hardware anchored Secure Boot

Figure 1 – Securing Network Infrastructure Platforms, from Kandula, NANOG 89

This is a lot of work. But it's by no means all the required work. Defensive security measures tend to work to a principle of diminishing returns and increasing cost. The first few measures, such as securing communications channels, constraining data flows and maintaining platforms that are updated according to known vulnerabilities, are generally simple and effective measures. Additional measures to check the authenticity of the hardware being used, ensure the integrity, authenticity and currency of the image being booted have a potentially higher cost and offer protection against more esoteric forms of attack. At some point the computer used to generate the executing code image needs to itself be certified, and the runtime profile needs to be assembled and compared against a standard benchmark. All stored data needs to be encrypted and these data keys need to be carefully maintained in a secure manner. The list of such actions can get quite extensive and very detailed. Standards are a great response here, and a commonly used framework is that defined in the US with the Federal Information Processing Standards (FIPS). Created by the National Institute of Standards and Technology's (NIST) Computer Security Division, FIPS defines a data security and computer system standard that organizations in the Federal sphere must adhere to in order to reduce their information technology risk to an acceptable level at a reasonable cost. Unsurprisingly, many others outside the US Federal procurement and operational areas have adopted the same standard. There are a number of modules within the defined security framework, but it appears that FIPS-140, which details cryptographic modules and testing requirements in both hardware and software is the most relevant for Internet infrastructure.

FIPS-140 outlines four levels of security. Modules are validated based on how well they fulfil the needs of the scenarios they will be used in. Level 1 is the lowest level of security, covering the basic security features. This is probably inadequate for trusted network infrastructure platforms. Level 2 improves the physical security aspects of cryptographic modules. Examples of required physical security measures are tamper-evident coatings, seals, or pick-resistant locks. Role-based authentication is included in this security level and ensures the operator accessing the module is authorized and is limited to their assigned actions. Level 2 also allows for software cryptography in a multi-user system environment. Level 3 requires enhanced physical security. A multi-chip embedded module has to be contained in a strong enclosure that erases critical security parameters when it is removed. This security level also uses identity-based authentication. Identities, roles, and assigned actions are authenticated before access is granted. A module complying with Level 3 security has data ports for critical security parameters physically separated from other data ports. For multi-user systems, the Operating System must be more trusted than in Level 2. Level 4 is the most secure part of the FIPS-140 standard. It requires tamper detection circuits to be able to detect any physical penetration. This level is best for when the devices are in a unprotected physical environment that intruders can access. Module protection in Level 4 extends to keeping voltage and temperature conditions within normal operating ranges. Modules must be able to detect fluctuations and erase themselves. Modules can also be designed to operate outside its normal operating range and remain secure.

These standards (FIPS) are commonly used to generate and store cryptographic keys on hardware security modules (HSMs), although the same standards and procedures can be applied more generally to critical network infrastructure platforms, although it is not so commonly used as a specification outside of HSMs and secure key storage.

How are we going with trustable Internet infrastructure?

At the heart of the Internet is the naming and addressing scheme. Are the infrastructure elements that operate this infrastructure use these security practices? Do they publish what they are doing? It would seem that today our track record in this area of trustable critical infrastructure operation is a mixed story.

Names

In the name space the procedures used to operate the single key that lies at the apex of the DNSSEC key hierarchy are published by the IANA. but there is some variance between intent and operational practice. For example, it is common practice to regularly change the trusted key value of the root key of DNSSEC, the Key-Signing Key of the Root Zone. The documented intent is that "Each RZ KSK will be scheduled to be rolled over through a key ceremony as required, or after 5 years of operation." The first KSK key roll occurred on the 11th October 2018, eight years after the initial KSK use commenced in July 2010. The second KSK has been in use for five years now and while there are plans being developed to perform a second KSK roll, the five-year deadline has already passed.

The underlying principle here is to "*say what you do, and do what you say*". The documentation about the management of the root key is well maintained. The operational procedures associated with this role have had their surprising moments (such as the failure of the locks on one of the safes that contain the KSK HSM but the operation practice appears to be sufficiently disciplined and robust so as not to undermine trust in the integrity of the root key.

What about the root servers at the apex of DNS name hierarchy? Given that some two thirds of the Internet's user base simply trust everything that they are told by the DNS, because they do not use DNSSEC validation, and given that all DNS queries logically start with a query to the root zone, then trust in the integrity of the operation of the root zone servers is paramount. The root server requirements are documented in RFC 7720 and the security considerations on the operational side of the root name servers are discussed in a Root Server document RSSAC-001. This is an eight-year old document that is perhaps more aspirational than informative. For example, the document states that "Individual Root Server Operators will adopt or continue to follow best practices with regard to operational security in the operation of their infrastructure." It does not state what measures are being used in these "best practices" and how the various root server operators conform to such practices in operating their service platforms. From a perspective of taking public actions to engender trust in this function, there appears to be some considerable room for improvement.

As one moves down the DNS hierarchy, looking both at the servers of the top-level domains and the servers for lower-level domains, the level of public information about the measures these service operators are taking to ensure trust in their operation becomes quite sparse. The DNS name resolution protocol is commonly layered above unencrypted UDP packet exchanges, so the possibility for interception, eavesdropping and tampering of DNS queries and responses has been realised in many scenarios. Even with recent moves to use the DNS over encrypted transport, such as DNS over TLS or DNS over HTTPS, the DNS model of trusted intermediaries (in the form of recursive resolvers) rapidly erodes the intrinsic value of authenticated encrypted transport. An authenticated intermediary can still pass an incorrect DNS response over a secured transport session.

And then there is the question "Is this DNS name the name of the entity I intended to communicate with?" How are entities and the service they provide over the Internet associated with DNS service names in an open and trustworthy framework? Again, the answer is hardly reassuring. If you also factor in the behaviour of IDNs (the use of Unicode in the DNS to allow domain names that use scripts other than Latin) then what you see is deliberately not what you are getting, and the entire issue of trust is based on entirely opaque foundations.

Without the use of DNSSEC and secure transport the DNS is hardly "trustworthy", and abuse of the DNS has become so institutionalised that we appear to have largely given up on making the DNS intrinsically more trustworthy. Even the use of DNSSEC and secure transport does not adequately address the question of who is at the other end of the subsequent connection. We trust these systems due to habit, based on the observation that "nothing bad has happened so far!" And even when something bad does happen, the lack of alternatives means that we are still largely coerced back in the same corruptible trust framework. The larger picture of the trustworthiness name space of the Internet is hardly reassuring.

Routing and Addressing

What about the address and routing infrastructure of the Internet? Just how well is this area performing in terms of a trustworthy infrastructure?

Here there are some obvious shortcomings in the picture. The continual stream of incident reports over cyber-attacks that were facilitated by deliberate corruption of the routing system do not inspire confidence and trust in the routing system. In addition, the fragility of the routing system is evident by the accidental leakage of routing information beyond its intended scope, causing conflicting routing information to result in misdirected packet forwarding. At the most basic level the issue here is the absence of a trusted reference model for routing, so there is no point of reference to allow a router to determine the veracity of dynamically-learned information. This is a deep problem and the absence of a secure framework for routing for the past thirty years is a good indicator of the inherent complexity of the problem. Constructing a framework trust in routing is at best an elusive objective.

There are four distinct elements of implicit trust in the Internet's routing system:

- Does every instance of the routing protocol used by every router conform to the same behaviour profile?
- Does the reachability information being passed through the operation of the routing protocol conform to the underlying topology of connectivity of the network?
- Will the routing system generating forwarding paths through the network that correspond to those computed by the routing protocol?
- Are the addresses used for identification of reachable endpoints authentic?

The first question, namely one of consistency across the massive diversity that is the Internet today is largely an article of faith as distinct from a testable proposition. The routing protocols are built according to standard specifications from the IETF, and from the perspective of the end user the degree to which all the routers along the end-to-end path are accurate and faithful implementations of this common standard is not directly verifiable.

For the second and third questions, the routing information used in these systems are self-relative, not absolute. There is no single routing protocol-generated topology map of the network that could be compared against a map assembled as a set of pairwise connections. With the exception of source-routed services the packet is passed through this stateless sequence of forwarding decisions where each decision is based on the packet's intended destination, not its source or the path that led to this router. External validation of these decisions becomes a prohibitively challenging task.

The final question is about the addressing system itself. Can we clearly authenticate an IP address as being usable in the context of the public Internet? The so-called "bogon" lists of IP addresses and AS numbers that are not duly registered with the address registries don't inspire any confidence that this is a robust system. The administrative framework of the address system is relatively opaque and the use of five regional address registries has resulted in differences in operational practices between the registries. Even simple questions, such as "which registry is authoritative for which IP address?", are not readily answerable in a verifiable manner. In the case where multiple registries claim to be the authoritative registry for the same address, then there is no higher level of authority that can resolve such a clash.

The use of a Resource PKI by the RIRs has improved the situation somewhat. If the registry system is trustable, then the registry publishes a certificate, using its private key to sign the certificate, that attest that the holder of a certain public key is the registered holder of a set of IP addresses and/or ASNs. This allows a user to validate an address holder's claim of address ownership by having the holder sign an address attestation using their private key. If the public key in the registry entry for that IP address can be matched against the attestation signature, then the credentials of the address holder can be validated.

However, there are still some troubling gaps in this story. How can a relying party determine which registry is the right registry to use? How well to the address registries operate their trust anchor and certification authority functions? What standards do they comply to? Who audits this compliance, and how often?

Networks are by their very nature dependent on trust. They replace direct physical interactions with transactions-at-a-distance and questions relating to the identity of the remote party and the authenticity of the transaction are intrinsic to the nature of networks. We are all forced into a relationship of trust in the operation of a myriad of elements within the networked environment and the operation of the supply chains that provided or manufactured these elements in the first place.

There are situations where additional elements or processes can be used to permit this trust to be validated, but this is by no means a universal response, or even a practical response in many cases. Personally, I'd classify the objective of a "trustworthy network infrastructure" as an aspirational target at best! We do a great job of verifiable trust in some situations and contexts in network infrastructure, a reasonable job in many cases, but there are still many other cases where the results are somewhat between mediocre and just plain useless!

Rakesh Kandula's presentation on trustworthy network infrastructure is online at the NANOG 89 site. The video can be found here.

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston AM, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net