



Open Networking & the Security of Open Source Software Deployment

A white paper presenting security considerations for practical
deployment

January 2021





Table of Contents	
Executive Summary	5
Welcome	7
Introduction	8
Open Source in Virtualised Open Networks	9
The Software Development Process	11
Shades of Open Source	13
Open Interfaces and Open Source Software Are Different	15
Two Perspectives: Systems and Component	16
Systems-level Approach	17
Component-level approach	19
Deployment Considerations	20
Layered Security Defence	20
<i>Security Standards</i>	21
<i>Industry Best Practice</i>	21
<i>National Regulations</i>	21
<i>Company Security Practices</i>	21
<i>Risk-driven Controls</i>	22
Networking Software	22
Virtualisation Layer	23
Open Radio Access Network Security Considerations	24
<i>Architectural Aspects</i>	24
<i>OSS Aspects</i>	25
<i>Areas for action</i>	25
A Broader Set of Infrastructure Considerations	26
Whole Systems Thinking	27
Hybrid Networks	29
Holistic Penetration Testing	30
Threat and Risk Assessment	31
Attack Trees	32
NIST 800-30	33
ETSI	33
MITRE ATT&CK®	34
Microsoft STRIDE	34
SAFECODE	34
DevSecOps	35
Control over the process	35
‘Shift-Left’	35
Some More Detail - Security Controls and Approaches	36

<hr style="border: 2px solid black; width: 100%;"/>	
Management Plane Security.....	36
<i>Administrator Access</i>	36
<i>Architecture</i>	36
<i>Bare Metal</i>	36
<i>Privileged Access Workstations</i>	37
Bottom to top security	38
<i>Root of Trust</i>	39
<i>A Zero Trust Approach</i>	40
<i>System Hardening</i>	41
<i>Virtualisation Layer Code</i>	42
<i>Software Composition Analysis</i>	43
<i>Common Vulnerabilities and Exposures (CVEs)</i>	44
Interoperability	45
<i>Interfacing</i>	45
<i>Application Programming Interface</i>	45
<i>Application Based Segmentation</i>	46
Cloud iNfrastructure Telco Task Force	47
Broader Cloud Considerations.....	48
<i>Cloud Root of Trust</i>	48
<i>ETSI</i>	48
<i>NIST</i>	48
<i>The Center for Internet Security (CIS)</i>	48
<i>Cloud Supply Chains</i>	48
GSMA Industry and Security Standards Activity Areas	50
Fraud & Security Working Groups	50
Securing the 5G Era.....	50
Telecommunication Information Sharing and Analysis Center	50
Coordinated Vulnerability Disclosure Programme.....	50
Security Accreditation Scheme	50
Network Equipment Security Assurance Scheme.....	51
Physical Security	52
Personnel Security	53
Bringing it together	54
Conclusions	62
Appendix A - Open Source Licensing	63

Executive Summary

This document introduces the concepts of Open Source Software (OSS) and open networking by exploring a variety of deployment scenarios within virtualised mobile networks. Wider and related aspects are discussed including an overview of the software development process. This theme is extended to identify various 'shades' of open source varying from unique new proprietary code developments through commercially-supported software packages including significant open source code and on towards open source community-supported software packages.

Before identifying the importance of both systems and component lifecycles, this whitepaper explains the differences between open interfaces and open source. Not only do these lifecycles operate at different cycle times, they also require different actions. The importance of layered security defences is discussed as well a number of broader security considerations such as whole systems thinking, hybrid networks, holistic penetration testing and threat & risk assessments. The emerging DevSecOps approach is discussed as is the concept of 'shifting left' security activities into earlier lifecycle phases to embed security through the lifecycle of a system.

Specific coverage is included addressing Open-Radio Access Network (O-RAN) Alliance security considerations through a discussion of two recently published reports from Ericsson¹ and the O-RAN Alliance². This is included as they reflect upon open source software security considerations in a rapidly evolving area where there is potential for significant future deployment.

In order to identify security actions that are relevant at the component level, a set of more detailed security concepts are presented.

Securing the equipment management plane is a vital area to protect service availability. Although individual platform layers may have security built-in, consideration is included covering 'Bottom to Top' stack security approaches such as Root of Trust, Zero Trust, Cloud Roots of Trust and System Hardening.

Content is included on the benefit of generating and maintaining a software bill of materials and the management of Common Vulnerabilities and Exposures. The security benefits of interoperability and interface testing are presented alongside application programming interfacing and application-based segmentation approaches.

Cloud considerations are included such as:

- The Cloud Infrastructure Telecom Taskforce
- ETSI's work on *offloading sensitive functions to a trusted domain*
- NIST's initiative to develop a practice guide for Trusted Cloud
- CIS Platform Benchmarking
- Cloud supply chains.

It is important to note the security activities and work undertaken within GSMA, to inform and support the industry with resources to raise the industry baseline for Telecommunications and its wider ecosystem.

A lifecycle approach to systems development and operation is described that integrates the range of security practices identified in this document and wider security best practice.

¹ <https://www.ericsson.com/4a4b77/assets/local/security/security-considerations-open-ran.pdf>

² <https://www.o-ran.org/blog/2020/10/24/the-o-ran-alliance-security-task-group-tackles-security-challenges-on-all-o-ran-interfaces-and-components>



In order to align with new technology approaches for OSS, there are many opportunities for action to create the best set of security considerations that incorporate proven existing security practices combined with new approaches.

Specifically:

- Where vendor software includes open source components directly within code or is included in a full stack supply, encourage vendors to update/patch upstream components quickly or enable operators to act directly.
- To incorporate a Software Bill Of Materials (SBOM) to ensure full visibility of the deployed code in use.
- Exploit the strengths of open source transparency through code inspection, Source Code Analysis (particularly to generate and validate an SBOM), dynamic application security testing and encouraging use of coding standards through both vendor-Software Development Life Cycles and Core Infrastructure Initiative.
- Where infrastructure virtualisation is delivered through a software package that is open source code-derived, use scanning tools to identify obsolete and vulnerable products and encourage a supply arrangement to enforce the ability to update out of date components within a stack.
- For infrastructure virtualisation, consider proving and re-using deployments with established industry benchmarks and common security-proven builds that have been extensively defined, tested and maintained. The Cloud iNfrastructure Telecom Taskforce (CNTT) has undertaken work in this area.
- Incorporate proven security methods that deliver 'Bottom to top' security to preserve the root of trust for the solution as a whole. Current equipment is often supplied from a single vendor, open networking is changing this and may mean there are different vendors involved in each layer.
- The O-RAN Alliance Security Group is defining security requirements to align to the specifications and interfaces. GSMA are keen to see and assist the O-RAN Security Group to drive the maturity of security specifications that will build confidence for scale deployments. These are important security considerations that require comprehensive design, feasibility and testing approaches that build maturity through practical experience.
- Consider the total operating environment into which open source code is deployed such that holistic security outcomes are considered across both new and existing infrastructures.
- Utilise a lifecycle approach such that security is designed-in, comprehensively tested in detail and in-context, deployed securely and then operated to maintain this security in-life.

Welcome

Welcome. This whitepaper offers a review of the security factors identified when deciding on deployment scenarios for open networking and deployment of open source software solutions within mobile networks.

A wide range of consideration is offered covering not just to the security aspects associated with using open source software but also, significantly, to the wider system and operational aspects necessary when considering the total operational network.

This whitepaper is undertaken as part of the broader GSMA Open Networking initiative where work is ongoing to define Minimum Viable Products and Use Cases. This document informs these models and provides high level considerations for security for mobile network operators developing their network capability and for potential vendors to build in and support beneficial security features.

The following MNOs are thanked for their support of the open networking and open source software security project: AT&T, China Mobile, China Unicom, Hutchison, MTN, Telefonica and Vodafone.



Introduction

There are many initiatives driving open architectures and virtualised telecoms infrastructure such as Telecoms Infrastructure Project (TIP), O-RAN Alliance, Linux Networking Foundation and the Open Networking Forum. The use of software from open source in a range of architectural deployments is rapidly increasing such as a software component running on virtualised infrastructure, to provide virtualised middleware, or within proprietary code implementation.

This area is of particular interest as mobile network operators seek to densify radio access networks in order to deliver new services. The ability to disaggregate the network and use virtualised components offers the potential to lower unit costs, increase vendor diversity, increase flexibility to grow or shrink services and enhance innovation potential.

OSS has a number of advantages, notably including that source code is accessible and subject to inspection, a wide community of developers can contribute and there is potential to accelerate telco cloud implementation. In contrast, there are various best practice steps that aim to 'make secure software' but none of these are mandated in the open source community whose main focus is functionality³. Hence, there is also security value in utilising proprietary code solution as it may have benefitted from secure code development practices. This whitepaper explores security practices that can make OSS deployments secure.

Whatever the change being implemented in a mobile network, it is rare that it is implemented as a 'green field' development. In many instances, the enhancement must co-exist and integrate with existing network infrastructure. It is for this reason that this whitepaper pays significant attention to the wider system issues within which any network change must exist.

Consideration has been given to the differing deployment arrangements for open source software as illustrated below.

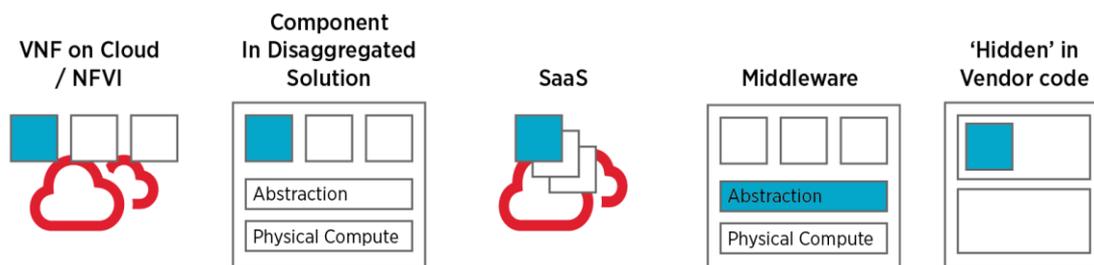


Figure 1 Open source software deployment arrangements

Open source software may be applied in a wide range of ways including:

- **as discrete code** (such as a Virtual Network Function (VNF) running on top of Cloud / Network Function Virtualisation Infrastructure (NFVI), or as virtualised Central / Distributed RAN Units on Cloud / NFVI);
- **as a component** within a disaggregated solution,
- **as part of the provision** of a wider Software as a Service (SaaS) provision. This may be found in a variety of deployments, e.g. as part of an open RAN or virtualised core deployment),
- **as middleware abstraction or virtualisation layer** between Commercial Off The Shelf (COTS) physical compute and the applications sitting on top. The applications may themselves be open source or proprietary in origin. This definition might be extended to include other software such as variants of Linux and Apache.
- **re-used** within vendor executable code. The fact that open source code is deployed may be obscured or 'hidden' as the executable code is difficult to inspect and source code may be difficult to obtain and inspect.

³ https://www.linuxfoundation.org/wp-content/uploads/2020/12/2020FOSSContributorSurveyReport_121020.pdf

Open Source in Virtualised Open Networks

Open source software can provide the middleware abstraction layer between COTS open-compute hardware and the applications sitting on top. The applications may themselves be open source or proprietary. This may be found in a variety of architectural deployments, such as part of an open RAN or virtualised core deployment.

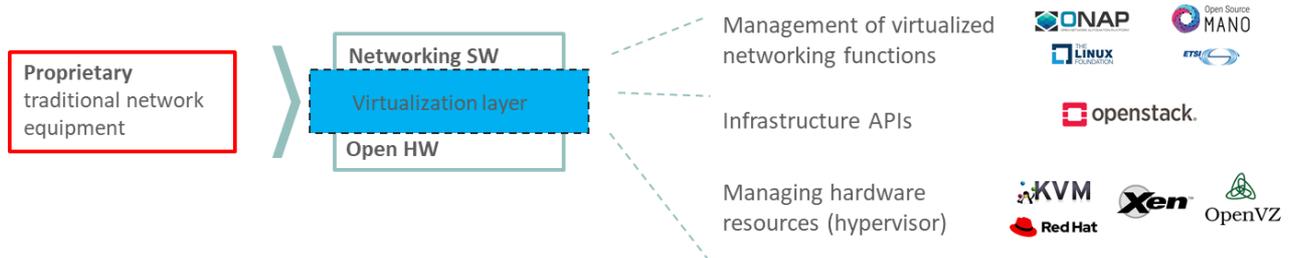


Figure 2 The move from integrated to virtualized

The drive for open networking has significant impetus driven by a number of factors including:

- To support 5G Cell densification in order to support low latency, high bandwidth and end-to-end network slicing.
- Offer potential to diversify the supply chain by expanding the potential number of vendor in the supply pool
- Offer potential to accelerate the roll-out of infrastructure and increase innovation.
- The new network architecture separates software from (general purpose) hardware using an intermediate virtualisation layer that exposes hardware capabilities in the same architecture used today by internet & cloud companies.

Components that are leveraging open source elements include:

- All distributions of Linux
- OpenStack distributions
- Software Defined Networking controllers
- Management & Network Orchestration
- Near-Real Time RAN Intelligence Controllers
- Virtualisation Hypervisors
- Service Orchestrators
- Ingress controllers
- Software based Load Balancers
- Application components:
 - Frontends
 - Middleware
 - Backends

Consideration has been given to the differing deployment arrangements for open source software as illustrated below.

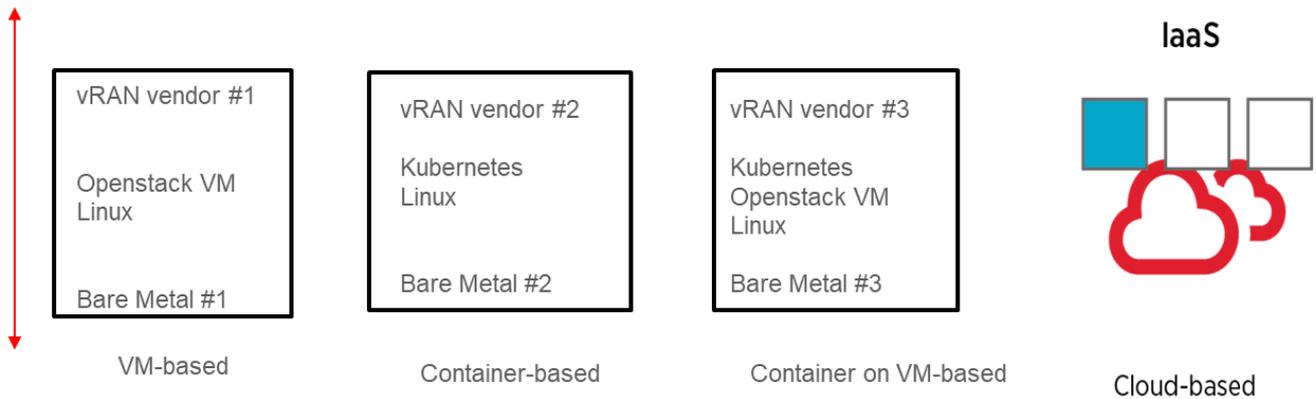


Figure 3 Some different deployment arrangements

Open source software may be applied in a wide range of ways⁴ including:

- As a virtual machine (VM) hosting application software such as vRAN capability(ies) or other virtualised network functions.
- As a container-based solution hosting application software such as vRAN capability(ies) or other cloud-native network functions.
- As a container-based solution built on top of a VM-based approach. There are efficiency considerations with this approach but has the benefit of being able to exploit the strengths of the VM approach.
- As cloud infrastructure providing the virtualisation to host a range of software capabilities for Infrastructure as a Service (IaaS).

⁴ See a good discussion at <https://pablo-montes.medium.com/installing-kubernetes-over-baremetal-or-virtual-machine-telco-workloads-8388c6f23ea5>

The Software Development Process

The software development of executable or binary code is illustrated in simplified form below. Source code is written in high-level human-readable languages such as C++. This can be newly written original code, can call upon or improve previously written code, use open source code or most likely some combination of all. Once the source code is completed, it needs to be compiled so that it is converted into a machine-readable code that will run on a specific compute platform. The executable code is the platform-specific code that actually runs to deliver the desired functionality. In practice, there are other steps in the process such as Interpreters and Linkers, which have been omitted for simplicity.

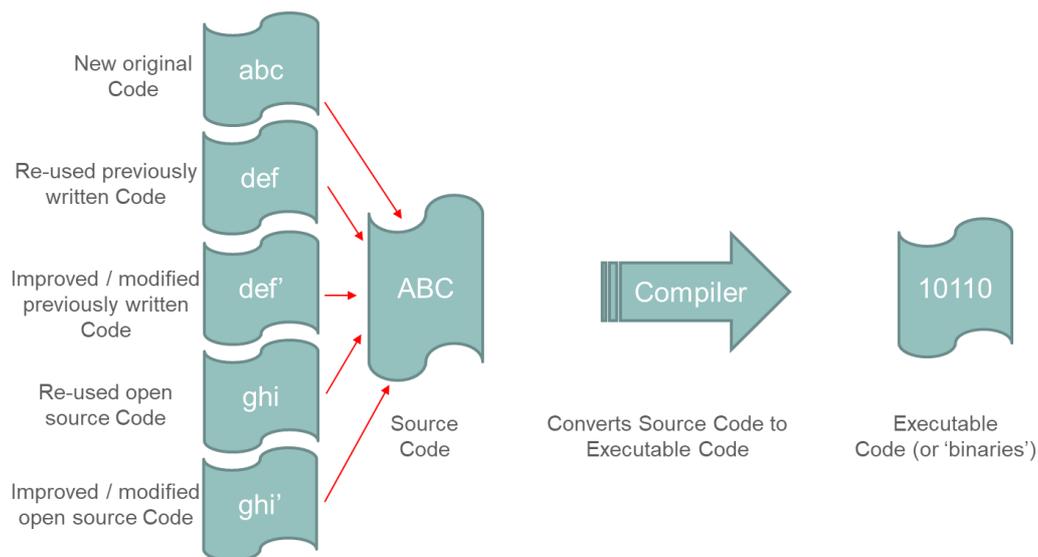


Figure 4 A software development process

Executable code is that normally provided by a vendor on the basis of a commercial offer. The associated source code is rarely made available due to concerns over commercial and intellectual property concerns and of code theft.

Open source code is available in its source format and as such is possible to inspect in detail through tools such as static and dynamic code analysis. The composition of the source code is ideally documented in detail to describe how the code works to assist in code maintenance and upgrade by other coders. The code composition can be recorded in a SBOM. The SBOM allows a detailed record of code components, especially re-used code, that can allow much improved support for the code when in-life. For example, should a new code vulnerability be spotted and published (such as Common Vulnerability Exposures (CVEs)) then an entity can check whether their code is affected by checking against the SBOM and can take appropriate remedial action (such as applying a patch, disabling the code, changing vendor).

A software package is a general purpose code bundle that can contain any type of files: executables, libraries, configuration files, etc. A library is a re-usable portion of code that may be included within other code, e.g. to expedite code development by removing the need to generate original code. It is important to know the entirety of the code deployed within a package including libraries, linkers, configuration files in order that the overall deployment is fully understood; this is why the SBOM topic above is important.

For proprietary executable code, the vendor will typically provide all the development resources (coders), follow their own company-specific software development coding practices (ideally benchmarked to the best in industry) and controlled according to their own configuration management processes. Support for the code is usually provided in a Maintenance contract with service level agreements.

For open source developed code, the main focus is typically to deliver required functionality and can be highly distributed by workforce and geographies. There is often little requirement for best practice development processes and for coding standards in general save for any that the community may agree to adhere to. Support for the code is varied. There are some well supported code bases that are contributed to by significant corporate groups. Elsewhere, support can depend entirely on the goodwill of the open source code developers and there is no guarantee of code fixes etc. As above, a major advantage of open source is that the source code is available for detailed inspection unlike vendor-specific executable code. This is also a disadvantage because attackers can equally inspect open source code to assess vulnerabilities.

An example of proprietary code re-using open source code can be described through the HCSEC Report in 2019⁵; when reviewing the Huawei code for an older LTE eNodeB product asserted “3.33 The report analysed the use of the commonly used and well maintained open source component OpenSSL. OpenSSL is often security critical and processes untrusted data from the network and so it is important that the component is kept up to date. In the first version of the software, there were 70 full copies of 4 different OpenSSL versions, ranging from 0.9.8 to 1.0.2k (including one from a vendor SDK) with partial copies of 14 versions, ranging from 0.9.7d to 1.0.2k, those partial copies numbering 304. Fragments of 10 versions, ranging from 0.9.6 to 1.0.2k, were also found across the codebase, with these normally being small sets of files that had been copied to import some particular functionality. There were also a large number of files, again spread across the codebase, that had started life in the OpenSSL library and had been modified by Huawei”.

Case study: Heartbleed

As an example of the downside of the re-use of open source code, consider the cyber security flaw called Heartbleed. Refer to New Zealand National Cyber Security Centre coverage: “OpenSSL versions 1.0.1 through 1.0.1f contain a flaw that allows an attacker to retrieve private memory of an application that uses the vulnerable OpenSSL library. The bug commonly known as Heartbleed, allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This potentially compromises the secret keys used to secure internet communication, the names and passwords of the users and the actual content. Exploit code for this vulnerability is publicly available.”

OpenSSL is the same code identified in the Huawei code described above. It is understood this is now remediated. CVE-2014-0160 is the official reference to this bug. CVE is the Standard for Information Security Vulnerability Names maintained by [MITRE](#).

⁵ HUAWEI CYBER SECURITY EVALUATION CENTRE (HCSEC) OVERSIGHT BOARD [ANNUAL REPORT 2019](#)

Shades of Open Source

It is relatively rare to conceive of 'pure' proprietary code such is the extensive use of open source software components in proprietary code. The amount of open source code will vary by application but can be significant. This is illustrated below.

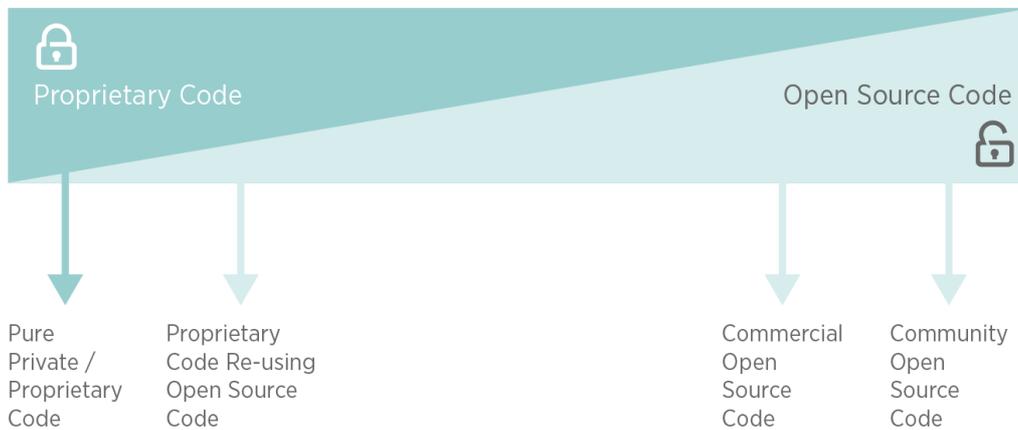


Figure 5 Different shades of open source

Proprietary code developers aim to speed delivery of their products by re-using open source code. Although the open source code may be small compared to the proprietary code, the proprietary code developer must recognize the continuing responsibility for the entire code base, proprietary and open source. These responsibilities include keeping track of open source code dependencies and making updates as they become available from the open source community providing the open source code.

Community open source code is produced from an open source community and entirely supported from within that community. Packages are free to download within the terms of open source licensing (see also Appendix A). There is often significant a rapid churn in open source community developments. This can result in a significant number of 'dead' or inactive code branches which are unlikely to attract further code development and support. In contrast, active branches will benefit from enhancements and bugs fixes. Whilst certain code branch functionality may seem attractive, it is important to understand the support, development and coder quality associated with it to ensure there is longevity to the code deployment.

Commercial open source code is open source code often produced from code developed by the commercial entity and also with other contributors. Commercial open source code is not the same as proprietary – it is often free to download in both source and executable forms. Both commercial and community approaches have advantages; one key differentiator is in the area of support / bug fixes. With a Community support arrangement, the software user is dependent on the community to generate the code fix / update to a non-deterministic timescale. Commercial open source can often be backed by a service level agreement to integrate newly developed open source software, update the software with the latest security patches and ensure that modifications to the software do not disrupt user operations. The service agreement is based around a service offer, i.e. it does not imply any ownership of the underlying code itself.

One other key differentiating aspect is the governance model for open source projects. Community open source tends to embrace an open governance model where decisions on features, future releases and code inclusion are decided by a diverse community elected leadership team with very well-defined operational procedures. Commercial projects can be established in open governance mode or are just simply managed by the key contributors from the commercial entities involved.



For commercial open source, the customer can usually rely on professional configuration of the software and does not need to have in-house expert knowledge on open source. Additionally, there may be liability and warranty provisions from a supplier when using open source whilst in existing open source licenses there is almost certainly an exclusion of any warranty or liability. The commercial supplier can provide this warranty on a commercial basis. In addition, commercial open source providers can offer expert support for the software, consultancy in its application, provide training people for software administrators, certify the software to operate on industry-standard hardware platforms and provide integration services that help customers deploy and operate their open-source software. These support arrangements can be very attractive should a company wish to gain more certainty in support over a five to 10-year service and investment lifecycle. Packaged open source distributions can include a range of additional components such as installers, linkers and utilities that may ease deployment but may also install unwanted / unneeded packages. Secure removal or deletion of these components should be considered in order to reduce the attack surface.

Open Interfaces and Open Source Software Are Different

The drive by TIP, O-RAN and others to define more granular, open and interoperable solutions has focused on defining interfaces between different network functional blocks and in some cases split functions into smaller blocks with and defining interfaces at such levels. Some interfaces have been specified outside of international standards forums (e.g. 3GPP), are proprietary between different components from the same manufacturer or have been industry developed standards that are implemented in a proprietary manner (e.g. Common Public Radio Interface (CPRI)).

Developing specifications in a fully open manner allows a range of new vendors to develop interoperable solutions in a more competitive manner. This does not necessarily mean that the solutions will be delivered through open source software. As described earlier, open source software is very likely to form part of the *source* code but the deployed, available *executable* code may still be proprietary in nature. The executable code implements open interfaces and contains open source software but is proprietary in nature.

The diagram below presents this concept for a stylized 5G virtualised open network:

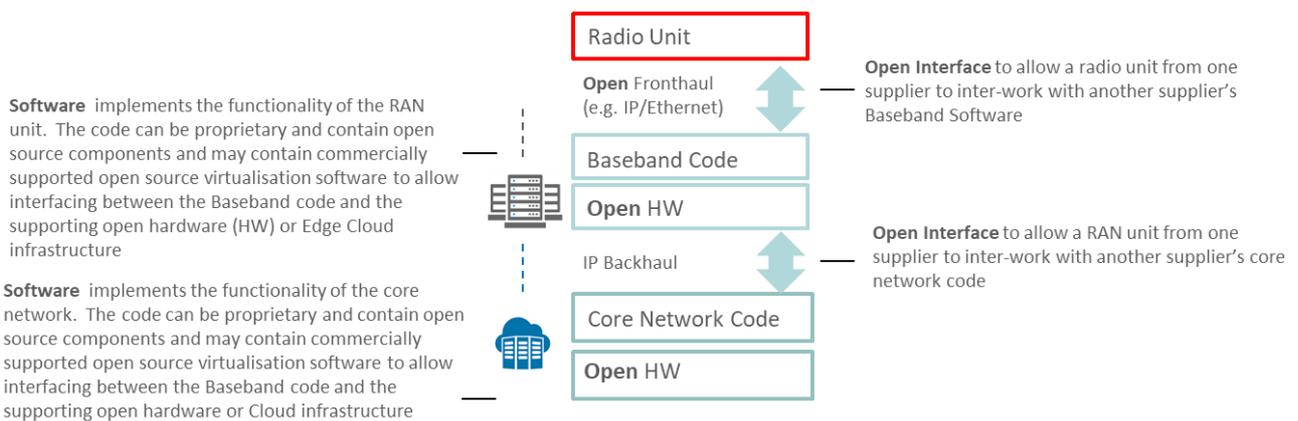


Figure 6 Software and open interfaces

The diagram below illustrates a range of potential code types within a generic system. Pure open source code is fully inspectable, pure proprietary code is non-inspectable as might be proprietary code with integrated open source libraries. Conversely, proprietary code may have benefitted from established software development lifecycle approaches. It is important to note that whilst the Application Programming Interface (API) design / specification may be open it does not mean the implementation is open source.

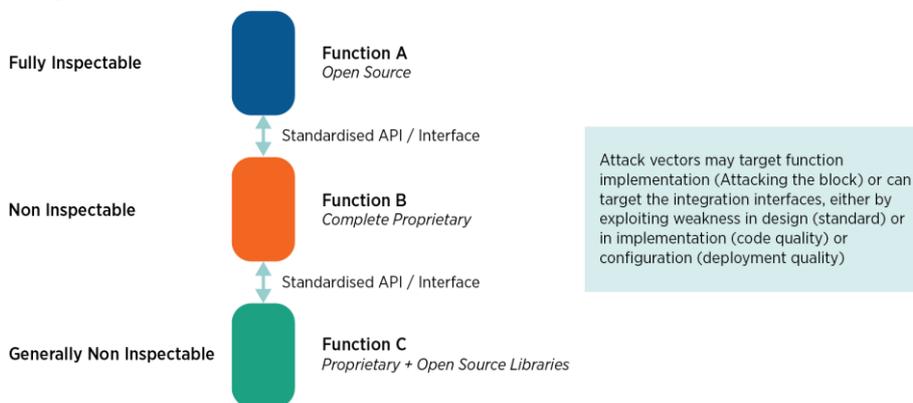


Figure 7 Code inspection

Two Perspectives: Systems and Component

The implementation of new capabilities can occur in varying levels of scale and at different parts of the network architecture. Accordingly, the processes for managing security will vary from a *systems* level approach (such as upgrading part of the RAN to a virtualised solution) to a more *component* level change (e.g. adding a new Virtual Network Function / Cloud-native Network Function). Operators may need to have different security approaches for both *systems* and *component* level network change so they can secure the open source software used within their networks

There are consistent themes across these references including a lifecycle approach to structured development. For open source code, some of the early stages (including the Coding Stage) of the lifecycle are undertaken by the developer community and the ability to influence security outcomes outside of the community is limited. Instead, focus can be considered across the rest of the lifecycle and in the longer-term by beginning to influence developer education and skills development.

This report therefore outlines two perspectives on managing and implementing change:

- Systems
- Component

When considering security controls and mitigations, it is useful to assess them within these perspectives.

For example, if consideration is being given to mandating the need for a SBOM then this might be achieved at the systems level through a contractual requirement placed on a systems integrator whilst at the component level this might be achieved through implementing a software composition analysis tool. The same control is achieved through differing means.

NIST have published a Cybersecurity White Paper:⁶ Mitigating The Risk Of Software Vulnerabilities By Adopting a Secure Software Development Framework. It makes the point that:

“Few software development life cycle (SDLC) models explicitly address software security in detail, so secure software development practices usually need to be added to each SDLC model to ensure the software being developed is well secured. This white paper recommends a core set of high level secure software development practices called a secure software development framework (SSDF) to be integrated within each SDLC implementation. The paper facilitates communications about secure software development practices among business owners, software developers, project managers and leads, and cybersecurity professionals within an organization. Following these practices should help software producers reduce the number of vulnerabilities in released software, mitigate the potential impact of the exploitation of undetected or unaddressed vulnerabilities, and address the root causes of vulnerabilities to prevent future recurrences. Also, because the framework provides a common vocabulary for secure software development, software consumers can use it to foster communications with suppliers in acquisition processes and other management activities.”

⁶ Mitigating The Risk Of Software Vulnerabilities By Adopting a Secure Software Development Framework: April 23, 2020 <https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04232020.pdf>

Systems-level Approach

The level of change (procurement, implementation) is at the *systems* level where the detailed view of change is indirectly controlled through higher-level activities. Change happens over a longer period (months) and applies to parts or all of a mobile network (e.g. RAN, Transport, Core etc.).

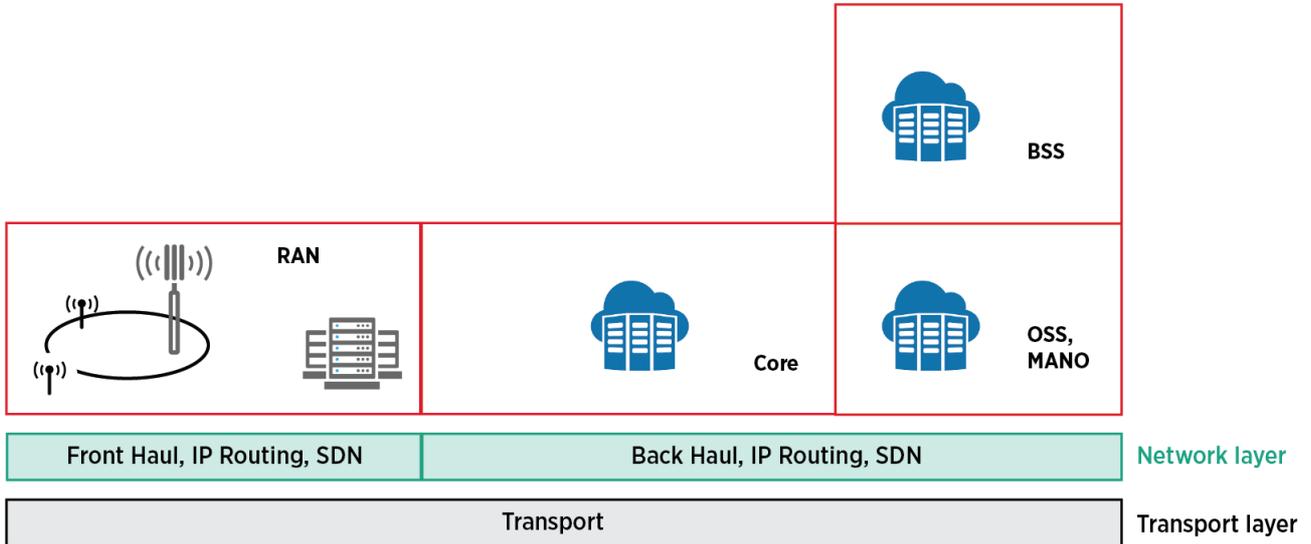


Figure 8 A high level view of a mobile network

The process steps through which change is effected are illustrated below in two lifecycle examples. There are a variety of actions at each step that help deliver security outcomes. We aim to capture a set of best practice security steps to take at each stage that will drive a set of strong security outcomes which are linked to a component-level change cycle (described later).

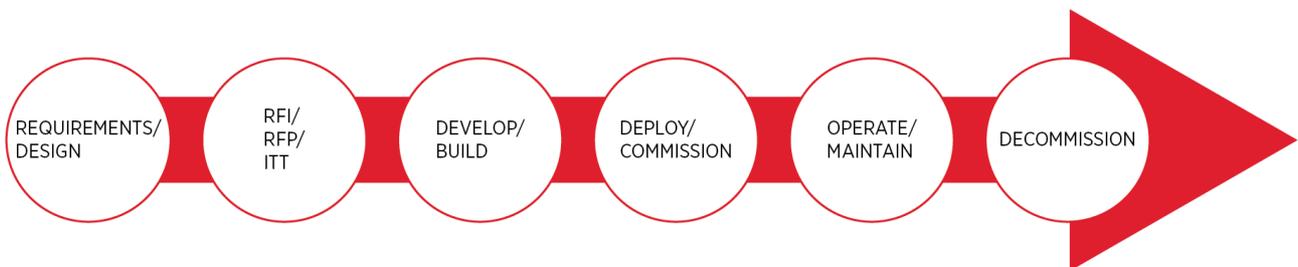


Figure 9 A systems lifecycle

Security considerations can and should be built within this lifecycle, aligned to its progress or through other approaches such as gate approvals.

In practice, delivery can be managed directly by the operator or by engaging a Systems Integrator (SI) or lead vendor. Using a systems integrator can provide a single point of responsibility for delivery and integration of different vendor equipment / software / services etc but can also form a long-term in-life reliance on the SI for support and maintenance etc. A more direct approach exposes increased detail of integration operations, direct risk management of deployment issues and provides in-house expertise for the system when in-life.

For example, implementation of a new RAN solution may involve several technology providers. Example non-exhaustive set of providers illustrated below.

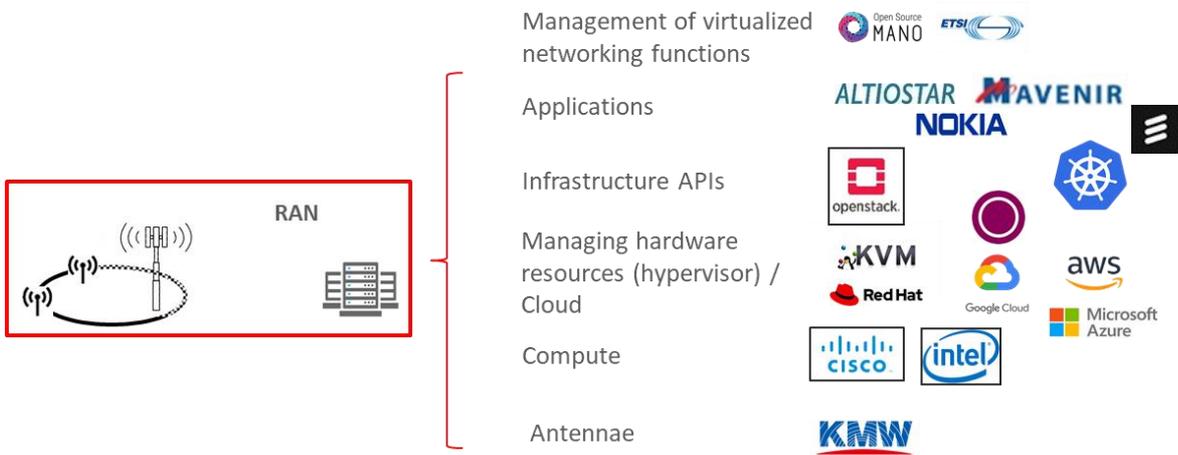


Figure 10 An example RAN implementation

To manage the delivery of this example may involve a wider range of parties that need co-ordination and management. This diagram is an example arrangement illustrating the involvement of a systems integrator.

In practice, the operating system may involve a range of other parties such as national regulators, Mobile Network Operators (MNOs) Shareholders, MNO Staff and skill interests, physical tower infrastructure and in-life managed service provider considerations.

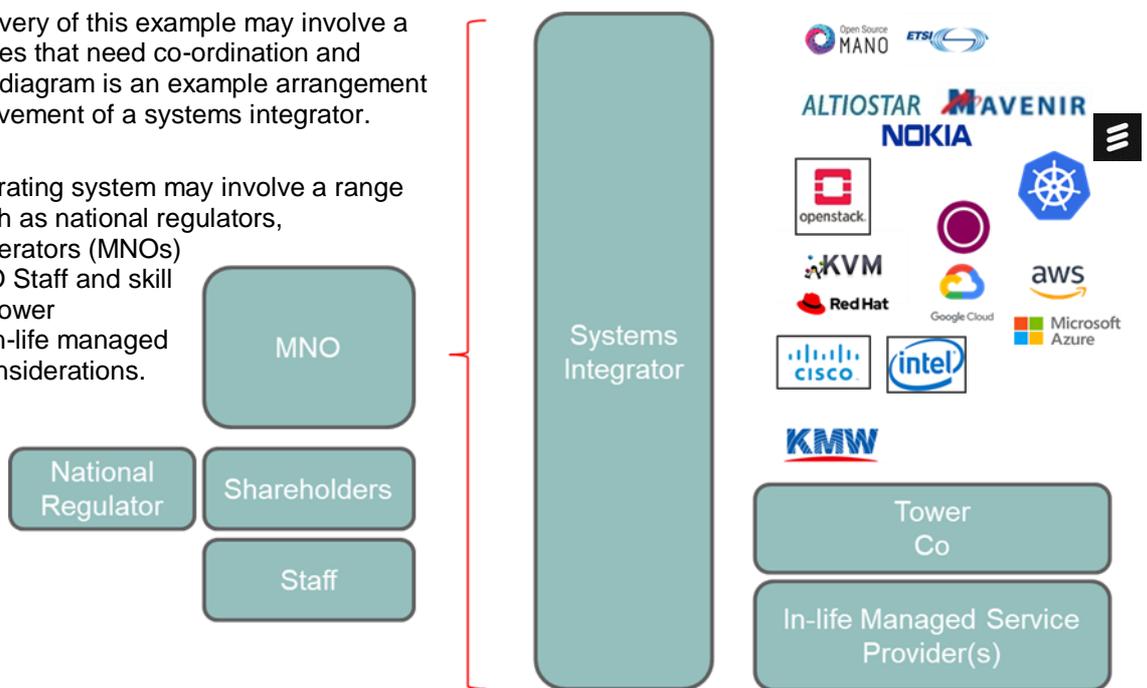


Figure 11 The use of a systems integrator

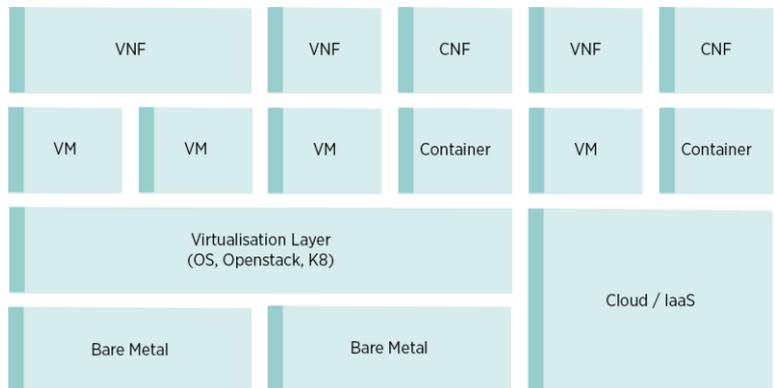
Component-level approach

The level of change (detailed capability and code delivery) is at the *component* level where the detailed level of activity is influenced directly.

Non-exhaustive examples of these components are:

- A Virtual Network Function
- A vRAN Control Unit
- A Cloud-native Network Function
- A middleware virtualisation layer

Figure 12 Some component combinations



These changes can happen quickly (days) with different versions of code developed on collaborative platforms and lifecycles may have a short cycle time due to emerging approaches such as Continuous Integration / Continuous Deployment (CI/CD) and more integrated Development, Security & Operations (DevSecOps) or (DevOps; shown below).

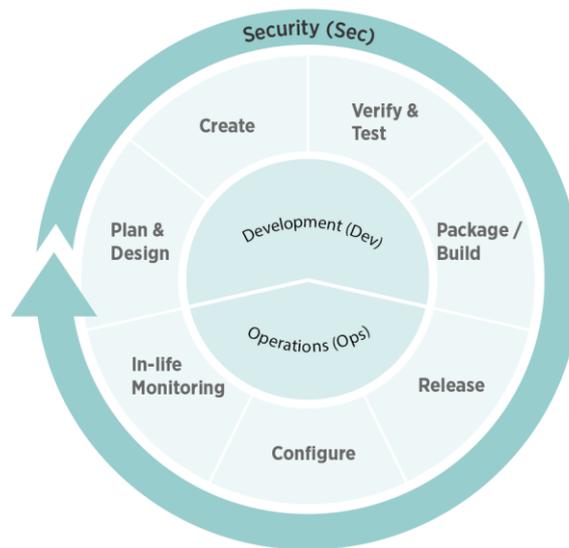


Figure 13 A component lifecycle

Examples of the process steps through which change is effected are illustrated below. There are a variety of actions, tools and best practice that help deliver security outcomes at each step. We aim to capture a set of best practice security steps to take at each stage that will drive a set of strong security outcomes.

Deployment Considerations

It is useful to consider the differing approaches to deployment of any solution stack. Any effective security strategy will deploy a layered defence informed by the threat landscape, budget, installed base, skills, risk appetite, regulation etc. In this section, consideration is given to approaches for networking software and the supporting virtualisation layer. The research undertaken to build the security evidence base for this project is published in the associated GSMA project document⁷: and is intended to offer a summary of some of the key information sources researched, so that the reader can relatively quickly gain an overview of open source software security considerations and undertake their own research into the referenced sources.

Layered Security Defence

A security strategy may be composed of multiple layers as shown below. The combination of security controls taken from each layer build to deliver a bespoke security solution for every operator. Security defences can be built on the controls and mitigations delivered from each previous security layer. Efficient and cost effective security approaches can be delivered by matching security controls to the threat model, understanding the security benefits built-in by lower level security standards and by customising the security decisions in the higher-level security levels. This is especially true where compliance with national regulations may have already mandated some security considerations. The resulting set of security approaches builds the overall security design.



Figure 14 A layered security defence

⁷ GSMA Document, Open Source Software Security Research Summary, Jan 2021

Security Standards

Globally, there are a wide range of International Standards Specifications and organisations where community and industry standards are specified. Ericsson present information on the role of security standards topic⁸.

GSMA author and contribute to industry best practice, international standards development and operational considerations through a range of activities including Telecommunication Information Sharing and Analysis Center (T-ISAC), Co-ordinated Vulnerability Disclosure (CVD), Security Accreditation Scheme (SAS), Network Equipment Security Assurance Scheme (NESAS) and Fraud & Security Working Groups (FASG).

Together these form a strong foundation on which to build stronger defences. For example, this approach is noted in the GSMA NESAS description. NESAS provides a security baseline to evidence that network equipment satisfies a list of security requirements and has been developed in accordance with vendor development and product lifecycle processes that provide security assurance. NESAS is intended to be used alongside other mechanisms to ensure a network is secure, in particular an appropriate set of security policies covering the whole lifecycle of a network. The scheme should be used globally as a common baseline, on top of which individual operators or national IT security agencies may want to put additional security requirements.

Industry Best Practice

There are a range of industry best practices that can be adopted including GSMA, National Institute of Standards and Technology (NIST), European Union Agency for Cybersecurity (ENISA), National Telecommunications and Information Administration (NTIA), Center for Internet Security (CIS), Open Web Application Security Project® (OWASP), Open Standards, Open Source (OASIS), national cyber security organisations, Building Security In Maturity Model (BSIMM), Cloud Native Computing Foundation (CNCF), the Linux Foundation, SAFECode and CNTT. This area is explored in greater depth in the associated GSMA project document covering the security research evidence base⁹. These offer a set of security principles that can be exploited, moulded and applied to enhance network security in a bespoke manner to reflect each operator's approach.

National Regulations

There are an increasing range of national regulations covering cloud, Internet of Things (IoT), data protection and network security. For example, the UK's Supply Chain Review¹⁰ has resulted in new legislation to be enforced through the national regulator, OFCOM, with high fines for non-compliance. Strict national controls will necessitate a design, configuration and operational response from operators that should be aligned to company best practices. Alignment allows the maximum benefit to be extracted from mandated controls and then additional security measures can be established on top.

Company Security Practices

Every operator will have established security approaches, procurement requirements, penetration test schemes, known improvement activities and Security Operations experience that have been shaped and refined over time. These will reflect the installed network and can be improved as network enhancements are delivered.

⁸ <https://www.ericsson.com/en/blog/2020/6/security-standards-role-in-5g>

⁹ GSMA Document, Open Source Software Security Research Summary, Jan 2021

¹⁰ <https://www.gov.uk/government/news/new-telecoms-security-law-to-protect-uk-from-cyber-threats>

Risk-driven Controls

Given that a strong security base will have been established from the previously described security approaches, a bespoke risk management activity can be used to identify and assess any residual areas of weakness or tactical mitigations that may enhance the overall security posture.

Networking Software

One approach for networking software is to utilise commercial companies to supply proprietary networking software such as VNFs, Central Unit (CU), Distributed Unit (DU) etc. The approach can be aligned to the Proprietary Code Re-using Open Source code model described earlier. Suppliers may include more established vendors or an increasing range of alternative vendors seeking to build a business on virtualised infrastructure. This perpetuates the current proprietary code supply model but offers an opportunity to introduce new vendors' software development processes and resultant code.

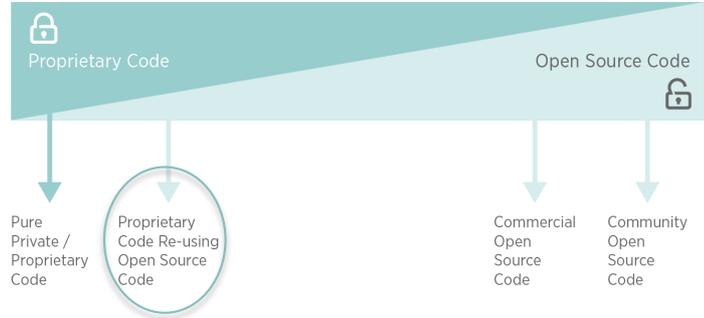


Figure 15 Proprietary Code Re-using open source code

It is noted that it can take a long time for telecom vendors to update/patch upstream components in their products due to the long development cycles for complex telecom products. This issue applies whether the upstream component is open source or not. This issue is more visible in the case of open source components as it is usually easier for MNOs to detect that a vendor is using an outdated component when that component is a well-known open source OS or library than when it is a piece of proprietary code.

Virtualisation Layer

One approach, for virtualisation software such as OpenStack, Kubernetes and operating system distributions, is to use commercial companies to supply distributions with open source origins. The approach can be aligned to the Commercial Open Source Code model described earlier. This approach can be attractive as it has potential to offer more contractual certainty for extended (5-10 years) support arrangements for deployments.

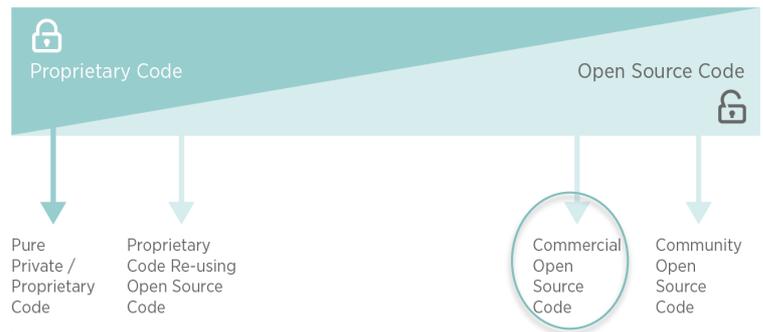


Figure 16 Commercial Open Source Code

Vendors may have long term support agreements with open source vendors and continue to use the versions long past the public support date. Scanning tools may then identify these products as obsolete and vulnerable and be unable to determine whether 'backported' patches have been applied to secure the versions. Due to this supply chain arrangement, contractual strength is an approach to enforce the ability to update out of date components within a stack.

The attraction of using more mainstream vendor distributions of code is understandable as it plays to mitigate against support, indemnity and integration risks. However, the distributions are sometimes slower in pace (to build in the very advantages required) which can lead to vulnerabilities to be exposed for longer periods (noted in previous section). Equally, it can provide a single point of focus for operators to leverage a particular software fix to be progressed (not always possible where there is reliance on community support).

The software skills and scale of commercial open source coders must be credible within the open source community for code to be accepted into a release (less of an issue if the code updates are within the vendors own code fork but code quality is still vitally important). SAFECODE¹¹ have generated a useful set of questions that can be used as part of any assessment of the effectiveness of contribution into open source communities.

¹¹ Managing Security Risks Inherent in the Use of Third-party Components at https://safecode.org/wp-content/uploads/2017/05/SAFECODE_TPC_Whitepaper.pdf

Open Radio Access Network Security Considerations

Two recent publications offer insight into security considerations for open source software and networks for open radio access networks.

The O-RAN Alliance¹² and Ericsson¹³ have published material addressing some of the security considerations relating to the delivery of security within O-RAN specifications, broadly into architectural and OSS topics.

The areas identified in these papers cover a range of security areas including:

- Architecture
 - Expanded threat surface [as a result of new interfaces in the O-RAN Architecture]
 - Bridging of management traffic to allow end to end management
 - Threat to the security trust chain introduced by decoupling of functions [a topic covered within this report]
 - Attack vectors back into the network core [to an extent this is an existing threat area with a range of security mitigations]
 - The need for mutual authentication, appropriate encryption protocols and access control.
- Open Source
 - Security vulnerabilities associated with Near-Real Time RAN Intelligent Controller (Near-RT RIC) [The Near-RT RIC is currently an OSC open source software implementation based on code developed by Nokia and AT&T. Samsung and HCL contributions started in summer 2020. O-RAN note the intent to develop application authenticity controls, software isolation techniques, secure standardised interfaces, testing methodologies, and access controls]
 - Practice a higher level of due diligence for exposure to public exploits from use of Open Source code [a top covered within this report].
- Functional Conflicts
 - xApp Conflict [O-RAN note the intent to develop application authenticity controls, software isolation techniques, secure standardised interfaces, testing methodologies, and access controls.

Architectural Aspects

The O-RAN alliance identify some architectural considerations such as “*The openness and disaggregation of O-RAN has many positive effects on security. Open interfaces are more transparent than black-box implementations, facilitating the alignment with security standards and best practices.*” Other areas of architectural interest observed in the publications include the consideration of an increased threat surface, the need to secure management interfaces and the need for threat modelling and risk analysis. O-RAN find an area for further consideration “*The strict latency requirements on the RAN need to be considered when implementing security controls, such as encryption, on the Open Fronthaul Interface.*” Additionally, O-RAN see “*The separation of the O-DU and O-RU introduces a potential new attack surface in the RAN: the open fronthaul interface operating the lower layer split (LLS) interface. The STG is currently studying the threats to this interface to gain a thorough understanding that will drive the specification of security controls on the interactions between O-DU and O-RU*”

¹² <https://www.o-ran.org/blog/2020/10/24/the-o-ran-alliance-security-task-group-tackles-security-challenges-on-all-o-ran-interfaces-and-components>

¹³ <https://www.ericsson.com/4a4b77/assets/local/security/security-considerations-open-ran.pdf>

OSS Aspects

Ericsson find that *“Industry has recognized that Open Source code introduces security risks. Open Source vulnerabilities are publicly available on the National Vulnerability Database (NVD). While this is intended for developers to disclose vulnerabilities, it is also used by hackers to exploit those vulnerabilities. Vulnerabilities frequently propagate as developers re-use free open source code enabling backdoors to attacks”*. The O-RAN alliance find the increased reliance on open source software in modern telecom platforms increases the Open RAN dependence on secure development practices within open source communities. Additionally, the Security Task Group (STG) will require the O-RAN Software Community (OSC) to adopt industry best practices in the OSC development pipeline. The OSC is adopting the Linux Foundation Core Infrastructure Initiative (CII) Badging security framework that defines the expected application security controls, change management practices, and security vulnerability tests that lead to more secure use and development of open source code.

Areas for action

Both publications offer a range of wider security considerations and recommend some areas for action. O-RAN see *“Network functions such as the Near Real-Time RIC, O-CU-CP, O-CU-UP, and O-DU, implemented as containerized microservices can leverage cloud native security advances such as hardware resource isolation, automatic reconfiguration, and automated security testing, which can improve both open source vulnerability management and security configuration management”*. It is understood work is ongoing within O-RAN to further specify and develop additional security requirements. They state, *“Recognizing the possible security challenges and the criticality of a secure RAN, the O-RAN ALLIANCE is following the 3GPP security design practices of rigorous threat modeling and risk analysis to identify security requirements and solutions that enable O-RAN to provide the level of security expected by the industry and 5G users.”* O-RAN products should be held to comparable assurance criteria as 3GPP 5G products such as GSMA NESAS/3GPP Security Assurance Specifications (SCAS) evaluations. It is also understood that Ericsson is working within the O-RAN Alliance STG to help standardise mitigations for the identified security risks.

Ericsson see that *“It is important to implement security best practices in a multi-vendor environment using Open Source code to build open, interoperable, secure network systems. This enables vendors and network providers to minimize the number of vulnerabilities and quickly respond in case a new vulnerability is found or exploited. These best practices should be implemented by each vendor at the individual product level and by the service provider at the network level:*

- *Life Cycle Management (LCM) with early integration of security to implement “security by design”*
- *Continuous development and continuous integration (CD/CI) with continuous regression testing and software security auditing*
- *Supplier Relationship Management with an inbound development process and strict security controls for FOSS*
- *Trust stack with software anchored to reliable, trusted supply chains and trusted operations with well-defined processes to reduce risk*
- *Vulnerability management with intelligence to continuously track, identify and re mediate vulnerable applications*
- *Multi-vendor system integration (SI) with continuous verification to ensure all vendors share the same interpretation and implementation of functions”*

It is recommended that there is consistent application of these principles by all vendors / code developers. These are important security considerations that require comprehensive design, feasibility and testing approaches to build maturity through practical experience. To a degree, this is to be expected as this is a typical experience for new specifications and implementation approaches. The precise security details of any particular specification will vary but the generic security considerations explored in this report are applicable. Although these reports are focused on O-RAN implementation, the security principles are relevant in a wider context and have proven solutions.

A Broader Set of Infrastructure Considerations

Detailed configurations and security considerations are increasingly being created and codified. This section draws out a range of the broader infrastructure considerations.

Containerisation¹⁴ is an Operating System (OS) level virtualisation technology. Containers are packages that rely on virtual isolation to deploy and run applications that access a shared OS kernel without the need for VMs. Containers hold the components necessary to run desired software. These components include files, environment variables, dependencies and libraries. The host OS constrains the container's access to physical resources, such as CPU, storage and memory, so a single container cannot consume all of a host's physical resources.

Containers are well-adapted to work with microservices, as each service that makes up the application is packaged in an independently scalable container. For example, a microservices application can be composed of containerised services that generate alerts, log data, handle user identification and provide many other services. Each service operates on the same OS while staying individually isolated. Each service can scale up and down to respond to demand. Cloud infrastructure is designed for this kind of elastic, unlimited scaling.

The cloud native concept is first introduced to Service-Based-Architecture networks and characteristics such as fine tuning, service customisation, high throughput are key enablers for virtualised infrastructure, which will see more effective execution, higher deployment density and scalability. ETSI's defined NFV architecture, NFVI, supports 6 types of virtualisation technologies, the foundations of which are VMs and containers. Containers and microservices are the future evolution of NFV cloud native and security is a significant consideration for their rollout. For example, host OS security is a typical container security threat as the lack of isolation from the shared host OS may introduce a potential threat. Because containers share a host OS, the obvious security threat is that the entire system can be more easily accessed and attacked when compared with hypervisor-based virtualisation. The container security threats also include container image file security, container orchestration security, container lifecycle management security, container run time security, etc. In order to facilitate the rollout of virtualised networks and services, security technologies to address these threats need to be considered in a timely manner.

Consideration is therefore needed at the component level within the container (e.g. code level), the host operating system level (e.g. resource security) and the architecture level (e.g. workload separation).

¹⁴ See GSMA FS.40 CR1001 5G Security Guide

Whole Systems Thinking

Consider a simplified mobile network presented below.

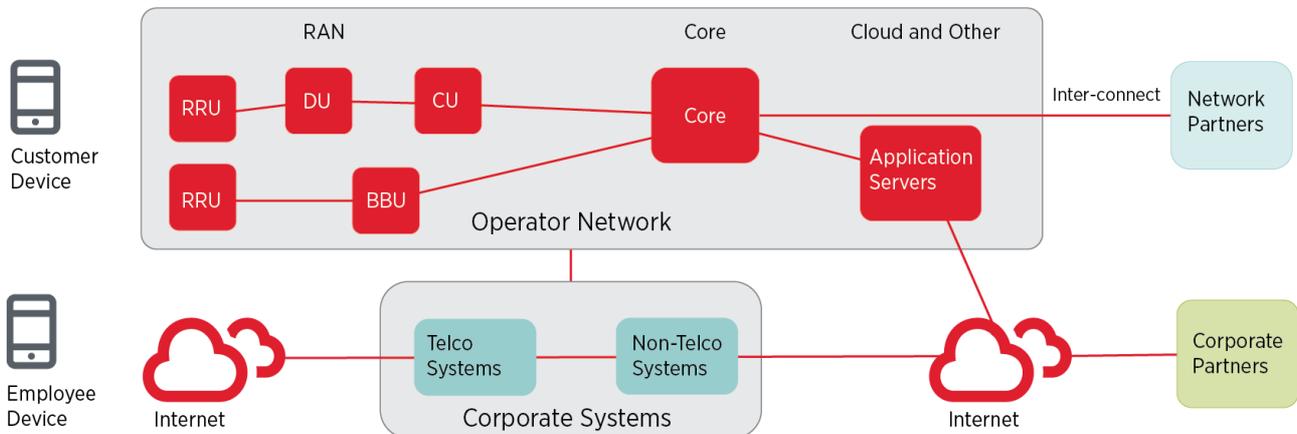


Figure 17 A simplified mobile network

To administer and manage a mobile network there are a wide set of telecoms, information technology (IT) and physical considerations. In addition to an operational mobile network, there are often a number of corporate IT systems that enable the broader operation such as corporate intranet, email, instant messaging, staff systems such as timesheets, sales systems etc.

These systems are accessed by a range of employee devices and used by the full range of staff functions including system administrators for the operational network. A range of wider corporate partner connections are often in place to provide access to wider IT, internet and cloud services, e.g. Salesforce, but also can provide access to enable managed service providers to remotely access the network. Crucially, there can be a connection between the corporate systems and the operator network which can provide an attack vector. Security of corporate IT systems may be treated differently / less favourably than the systems used in the operational network and hence lateral movement from a different security zone to a high secure zone is a concern.

The operational network is accessed by a range of customer devices. Disaggregated access networks enable a range of deployment options, configurations, equipment approaches and cost bases but can also present a wider attack surface to protect. This can take the form of new exposed interfaces such as eCPRI but also within the bottom to top deployment stack of any particular function such as a DU, where application software is exposed to a virtualisation layer and in turn to open hardware / bare metal compute (see Figure 18).

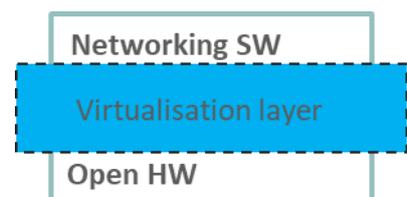


Figure 18 A disaggregated network component

A wide range of attack vectors can be identified when considering the complete system of both operator network(s) and the associated corporate IT systems.

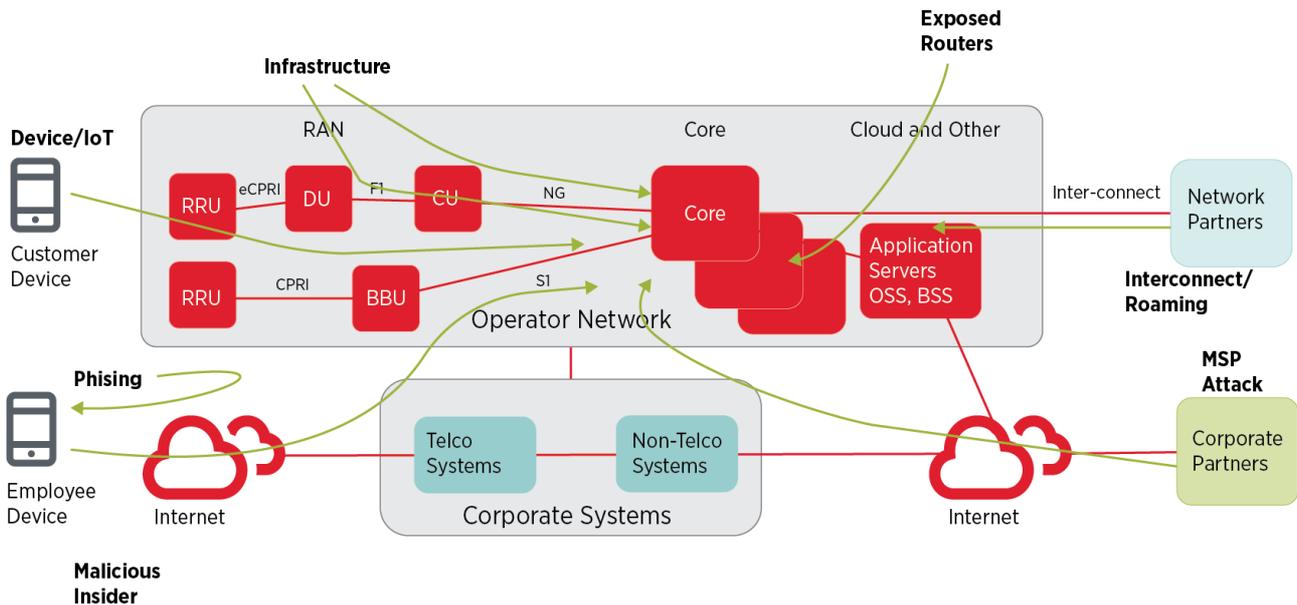


Figure 19 Network attack vectors

Any product (containing OSS or not) needs to protect itself from a range of attack vectors. There are a number of attack vectors presented and each requires strong security controls and processes to minimise the likelihood and impact of any attack:

- **Phishing attacks:** well-engineered and styled phishing attacks continue to have a finite success rate in penetrating perimeter defences. Consequently, anti-phishing campaigns and well architected internal network controls making lateral movement more difficult are important activities
- **Malicious Insider:** in a similar manner, internal controls, least privilege, strong authentication and employee vetting make it harder for a malicious insider to gain traction
- **Managed Service Provider attack:** remote compromise of a managed service provider offers a potential attack vector. Strong vetting, least privilege and trust domains form part of any defence.
- **Inter-connect / Roaming / Internet Signalling and DDOS attack:** this attack vector is well documented and attracts significant coverage in GSMA Security documents
- **Exposed routers and servers:** a network operator will have a significant estate of vendor equipment, router and server infrastructure. It is important to have a strong grasp of the inventory of equipment in order that it can be managed and protected. Once the equipment fleet is identified, it must be protected and configured against attack. This is particularly true for any internet-exposed management interfaces. Legacy equipment can use protocols with limited in-built security, e.g. Telnet. These exposed interfaces must be configured to use secure protocols or have additional security controls such as VPN protection to reduce the likelihood of success for an adversary attack. This applies to virtualised deployments in the same sense, in that bare metal compute, storage and network devices must be protected. Additionally, unused management protocols, internet services and accounts can be disabled to limit attack opportunities.
- **Infrastructure Attack:** physical attack of network infrastructure, e.g. at Cell Site or Data Centres can be minimised through physical protections as well as access controls, alarming etc. A further layer of defence is to ensure management and other equipment interfaces are suitably protected to prevent onward attack within the wider network. Physical attacks on RAN can also include Joint Test Access Group¹⁵ (JTAG) attack and Serial management ports compromise, etc

¹⁵ JTAG is a common hardware interface that provides a method to communicate directly with the chips on a computer board

-
- **Device attack:** with increasing access bandwidth and a range of malware attacks on device, protection must be considered against device-based network attacks (e.g. signalling 'storms', Denial of Service attacks, IoT Compromise) back into the network. Additionally, devices themselves may be subject to individual attack
 - **Air interface attack** onto Radio Access Network or further in the wider network.
 - **Supply Chain** (not shown) where equipment / software experiences interference in the process of supply / deployment.

One critical security aspect is the link between the corporate and operator networks as it provides an attack vector into the operational network. Good security practices can mitigate this risk through secure networks, strong authentication and least privilege practices alongside strong privileged access management (PAM).

Approaches such as Zero trust, Roots of trust (see later section) and Trust Domain Separation¹⁶ are also important security concepts (an area explored in the Management Plane section of this report). Strong security controls in this area can significantly reduce the attack surface for phishing, malicious insiders and external attacks via corporate partner arrangements¹⁷.

Hybrid Networks

As networks evolve, focus is applied to introducing new capabilities (such as higher bandwidths and low latency) that in turn require virtualised infrastructure and network functions. This can sometimes be at the expense of legacy equipment still in service but often builds on top of existing infrastructure (like Non Stand Alone 5G where 5G New Radio is built on top of a 4G core).

Newer and more complex systems will introduce new vulnerabilities but there is a strong focus on deploying these new systems securely¹⁸. These *hybrid* networks require consideration as a *whole system* as security weaknesses in legacy equipment can provide an attack vector into newer systems. For example, older Physical Network Functions (PNFs) may need to trust newer VNFs and both PNFs and VNFs will be susceptible to differing security vulnerabilities, yet both must work coherently and securely¹⁹.

Hence, it is important to consider common aspects within the total network architecture such as the management plane, underlying transport, shared backplanes, internet-exposed infrastructure and servers. Accurate asset inventory and internal controls that limit lateral movement and escalation of privileges are part of an effective defence. Knowledge of the complete asset base (e.g. servers, services, routers, storage) and address space (e.g. IP address ranges, Border Gateway Protocol (BGP) arrangements, Signalling ranges) is key to support a comprehensive approach to total asset management. This should include supporting services such as Domain Name Service (DNS), Dynamic Host Configuration Protocol (DHCP) and Active Directory®.

¹⁶ ETSI GS NFV-SEC 026: "Isolation and trust domain specification"

¹⁷ See also, GSMA documents FS30 Security Manual and FS31 Baseline Security Controls

¹⁸ See Communications Security, Reliability and Interoperability Council [CSRIC VII Report on Risk to 5G from Legacy Vulnerabilities and Best Practices for Mitigation](#) June 2020

¹⁹ Explored in s5.17 of 3GPP TR 33.848 V0.5.0 Technical Specification Group Services and System Aspects; Security Aspects; Study on Security Impacts of Virtualisation (Release 16)

Holistic Penetration Testing

Often penetration and vulnerability testing is undertaken against specific platforms and services, yet the potential to move laterally and escalate privileges within and across the entire system-estate is a major consideration for would-be adversaries. In recent years, holistic telecoms²⁰ and banking²¹ schemes have been developed to provide controlled, bespoke and intelligence-led security testing of networks. The premise is that testing mimics the realistic approaches of threat actors and can explore a range of the attack vectors identified in the previous section. Approaches focus on sophisticated and persistent attacks on critical systems and essential services, across a range of scenarios, with priority given to the systems identified as systemically important. The level of *system* might vary from the entire network infrastructure down to a Core Network focus or lower levels to infrastructure implementations. The idea is to adopt an ethical security test approach to identify operational security weaknesses in a controlled and safe manner in advance of any external attack.

Intelligence can:

- identify potential attack actors and attack tools
- provide targeting reports to identify key administrator personnel that may be subject to a phishing or other attack
- identify other open source material (such as equipment vendors and products in use) useful to an attacker in formulating their attack strategy.

Adopting an intelligence-led holistic penetration test approach means the process can adapt to changing threats. It can also:

- exercise business processes to detect, respond and recover from attacks
- explore the practical and real-world total system attack surface that may be exploited by a motivated adversary
- improve Board-level engagement and security profile
- evidence real security weaknesses to provide real data for business cases for new security investments and to effect remediation
- evidence the effective implementation of previous security investments
- provide an associated benefit in testing the security arrangements for customer data privacy arrangements
- be delivered in 4 Phases: Initiation, Threat Intelligence, Adversary Simulation, Remediation and Improvement.

Consideration can be given to adoption of this threat intelligence-led holistic penetration approach and undertaking practical assessments from the design/build stages before the services go into operations. When the deployment is in operation, there is the need to carry out frequent (yearly or when there is a significant update to the service) penetration tests in addition to the regular (monthly, quarterly etc.) vulnerability and compliance scans.

GSMA have considerable threat intelligence through T-ISAC, CVD, SAS, NESAS and Fraud & Security Working Groups²².

²⁰ UK Office of Communications; <https://www.ofcom.org.uk/phones-telecoms-and-internet/information-for-industry/network-security-and-resilience/our-work>

²¹ European Central Bank TIBER-EU scheme <https://www.ecb.europa.eu/paym/cyber-resilience/tiber-eu/html/index.en.html> & the Bank of England CBEST scheme <https://www.bankofengland.co.uk/-/media/boe/files/financial-stability/financial-sector-continuity/cbest-implementation-guide.pdf>

²² <https://www.gsma.com/security/>

Threat and Risk Assessment

Threat and risk assessment can be applied to understand the overall security of the solution. This can be applied in a Secure by Design approach²³ across the system development lifecycle from idea/concept to operations.

1. Analysis & Planning – Impact Assessment of architectural design choices and overall cost base
2. Procurement – assess risk coverage provided by responses and likely cost of residual risk. This in turn, supports the development of a business case and selection
3. Design & Build – use threat modelling in architecture and design guidance and reviews, by identifying threats and mitigations of vulnerabilities²⁴
4. Validation – Penetration Testing to verify approach
5. Deployment – use threat assessment to inform approval to go live after vulnerabilities and risk have been fixed, baseline regression testing, defect management plan tracking etc.
6. Operations & Monitoring – continuous security assurance activities, risk management, security awareness and training.

This is illustrated in a system lifecycle approach below.

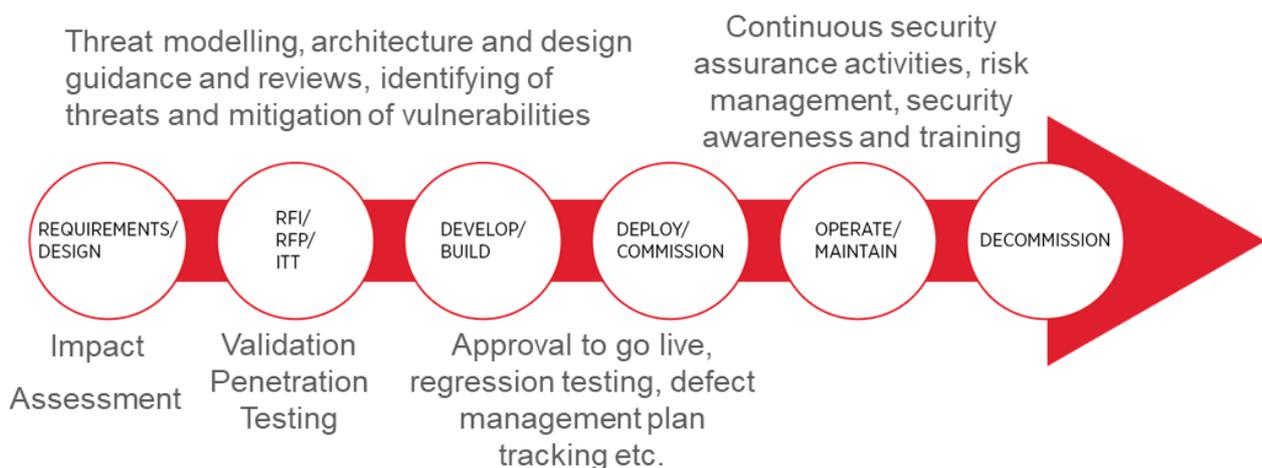


Figure 20 Threat and risk assessment applied through the systems lifecycle

Threat and Risk assessment allows consideration of the most likely and impactful risks considering the technical security threats to the business given its open source code choice, vendor selection, network architecture, software builds, etc. By focusing on the areas of threat / risk, a business can examine the gross risk likelihood and impact. By considering the effect of existing controls and mitigations, a net risk position can be determined. A review of this net risk position can assess whether the risk profile is within the company risk tolerance or whether additional controls and mitigation activity is required to further reduce the net risk position. This risk quantification activity, can directly feed into the internal business case for investment, such as investing \$1m in new security controls might reduce the threat impact by \$10m. Once

²³ https://www.ey.com/en_gl/consulting/how-to-manage-cyber-risk-with-a-security-by-design-approach & <https://iapp.org/resources/article/security-privacy-by-design-principles-sp/>, <https://www.sans.org/reading-room/whitepapers/awareness/security-design-systems-road-map-approach-39370>

²⁴ GSMA FS.33 CR1001 Network Function Virtualisation (NFV) Threats Analysis

the set of Controls and Mitigations is agreed, these can be built into the delivery programme for new designs or for new change programmes of existing infrastructure.

There are a range of risk assessment approaches that are discussed below.

Attack Trees

The UK's National Cyber Security Centre (NCSC) has published a report²⁵ examining the systematic analysis of a telecoms network from an *attack tree* perspective. The idea being to identify possible attack approaches from an attacker perspective and then break these down into different categories of attack type which can then be considered against the effects of existing controls. The approach began by drawing upon existing threat and attack data, global attacks on telecoms systems, practical industry security practitioner input and international security standards. From this data, a series of attacks was pulled together into 'attack trees'. Each attack tree was considered for their relative risk of success and likelihood. From this analysis, the most important risks can be listed. Security controls and mitigations can then be considered in order that the net risk position is at an acceptable level for the business. The NCSC report highlights 5 areas of the highest scoring attack vectors:

- exploitation via the operators' management plane
- exploitation via the international signalling plane
- exploitation of virtualised networks
- exploitation via the supply chain
- loss of the national capability to operate and secure our networks (dependency).

Adopting an attack tree assessment of a given network design can identify specific and localized security risk areas for consideration against design improvements or validation of the overall design. This specificity can add a custom security design that identifies new or specific threats that might not be addressed by existing control areas. This process is particularly powerful if it is maintained over the lifetime of the system under consideration with regular reviews that reflect changing threats, broader system enhancements, load data and security patching of the system.

²⁵

<https://www.ncsc.gov.uk/files/Summary%20of%20the%20NCSCs%20security%20analysis%20for%20the%20UK%20telecoms%20sector.pdf>

NIST 800-30

The National Institute of Standards and Technology publish a comprehensive guide²⁶ for conducting risk assessments. The guide identifies a range of useful content and considerations for consistent risk assessment including:

- Fundamentals of risk management
- Process for conducting and maintaining
- Threat sources and threat events
- Vulnerabilities and predisposing conditions
- Likelihood
- Impact
- Risk Determination.

ETSI

ETSI have a range of useful documents including a Technical Specification²⁷ for Method and pro forma for Threat, Vulnerability, Risk Analysis (TVRA). This contains useful content covering:

- TRVA Method
- TRVA Process
- Risks
- Security counter-measures identification
- Counter-measure Cost-Benefit Analysis.

A broad set of 3GPP security design practices²⁸ include:

- 33.401 - 3GPP System Architecture Evolution (SAE); Security architecture
- 33.501 – Security architecture and procedures for 5G System
- 33.511 – Security Assurance Specification (SCAS) for the next generation Node B (gNodeB) network product class.

²⁶ <https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final>

²⁷ https://www.etsi.org/deliver/etsi_ts/102100_102199/10216501/05.02.03_60/ts_10216501v050203p.pdf

²⁸ <https://www.3gpp.org/DynaReport/33-series.htm>

MITRE ATT&CK®

The MITRE ATT&CK®²⁹ framework is a knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community. The framework covers a range of tactics:

- Reconnaissance
- Resource Development
- Initial Access
- Execution
- Persistence
- Privilege escalation
- Defense evasion
- Credential access
- Discovery
- Lateral movement
- Collection
- Command and control
- Exfiltration
- Impact.

Microsoft STRIDE

Microsoft's long established STRIDE³⁰ approach to threat modelling remains a useful approach. STRIDE is an acronym for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. The concept is to decompose your system into relevant components, then analyse these for susceptibility to the threats, and then mitigate those threats. The desired security characteristics map directly to the threat areas:

- Authentication maps to Spoofing
- Integrity maps to Tampering
- Non-repudiation maps to Repudiation
- Confidentiality maps to Information Disclosure
- Availability maps to Denial of Service
- Authorisation maps to Elevation of Privilege.

SAFECODE

SAFECODE have published a useful discussion document³¹ that discusses threat modelling, identifies various methodologies and within the context of a lifecycle approach.

²⁹ <https://attack.mitre.org/>

³⁰ <https://docs.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach>

³¹ https://safecode.org/wp-content/uploads/2017/05/SAFECode_TM_Whitepaper.pdf

DevSecOps

Control over the process

Operators may move more towards a Development, Security, Operations (DevSecOps) processes that more closely integrate security considerations in the software builds. The advantage of this approach is to increase the speed of code deployment into live networks. Traditionally, there were separate Production (Development) networks and Live (Operations) networks. This allowed a partitioning of technical risk as development code could be tested away from live networks, thus de-risking new deployments.

As development and operations become more closely linked it allows a faster cycle time of code development and deployment and the potential to deploy smaller incremental code changes. This comes with some additional risk too, in that new code may have bugs that become deployed in the live network.

The interaction of operators within the DevSecOps process will be ever more important, especially when there are differing priorities, toolsets, risk appetites and processes. This will need additional consideration as different DevSecOps processes will be aligned to different supplier software coding practices with the intent to increase the velocity of code deployment into live operational networks.

Further thought is needed to corral the differing operator interests into the supply chain of vendors. Protecting the code repositories and development environments are key areas of focus. OWASP have a project attempting to capture the maturity of DevSecOps approaches through the DSOMM project³².

'Shift-Left'

One of the approaches to implement bringing the Security (Sec) part of DevSecOps is the concept of shifting the security review process "left" or earlier in the software development lifecycle³³.

This means the operator security team getting involved in the earlier system lifecycle stage such as the design phase, i.e. not leaving security involvement to a late stage when it can be harder to substantively improve security. A security review can be added as part of the release 'gate' factor for the design to move on to the development stage. Another example is to develop an automated testing process that can identify common security vulnerabilities and run continuously over the test period. The security team can be involved in the design and development of this testing.

³² <https://owasp.org/www-project-devsecops-maturity-model/>

³³ <https://cloud.google.com/solutions/devops/devops-tech-shifting-left-on-security>

Some More Detail - Security Controls and Approaches

Whilst earlier sections have considered the broader considerations and systems aspect within which open source based solutions may reside, this section focuses more on some specific security controls and approaches that are considered particularly helpful.

Management Plane Security

The management plane of a network is a very powerful part of network control and configuration of infrastructure. Consequently, protection of management plane security is a priority in order to protect network availability, integrity and confidentiality. A previous section highlighted the link between corporate and operational networks. In practice, this link is sometimes achieved through use of the same standard corporate device to both undertake corporate productivity activity (email, Intranet access) and administrative actions with the operational network. Thus, compromise of this device can provide an attack vector or bridge into the operational network.

This area of priority was explored in a report³⁴ identifying the need for a Privileged Access Workstation (PAW) approach. The report identifies the need for additional national regulations to mandate a series of 'Telecoms Security Requirements' (TSRs) that would segregate access to these important activities. There is merit in considering some of these controls in more detail.

Administrator Access

Administrators for network equipment and services can be specifically authorised. Operators can enforce the principle of *least privilege* and *separation of duties* on their privileged users, explicitly authorising their system access and requiring additional multi-factor authentication per intervention.

Architecture

The management plane is the highest trust domain due to its direct and powerful control of the network. As such, it can be isolated from other networks; especially internet-connected networks.

It is important to create dedicated secure out-of-band management zones for management of operator networks. Access to the operator RAN or Core Networks can be possible through mechanisms such as remote access methods like Citrix etc, but these must have strong controls in the own right.

Additional controls such as Privilege User Access Management, Logging and Monitoring and well-defined User Access Management processes amongst others must be implemented.

Segregation of the management plane³⁵ can be put in place to separate access to different vendor equipment types, thus limiting the potential for a single-access compromise to feed across to a wider effect. Connections can be further protected through use of secure, encrypted and authenticated protocols.

Bare Metal

Later in this section and report, additional security considerations (e.g. Trusted Platform Modules) are identified that are important considerations in the selection of hardware (bare metal). The selection of a specific hardware vendor and platform are informed by vendor responses to the mandated security requirements to support security functions such as Trusted Platform Modules.

³⁴ UK National Cyber Security Centre Report: Summary of the NCSC's security analysis for the UK telecoms sector <https://www.ncsc.gov.uk/files/Summary%20of%20the%20NCSCs%20security%20analysis%20for%20the%20UK%20telecoms%20sector.pdf>

³⁵ <https://www.cyber.gc.ca/en/guidance/network-security-zoning-design-considerations-placement-services-within-zones-itq-38> & <https://www.sans.org/reading-room/whitepapers/bestprac/infrastructure-security-architecture-effective-security-monitoring-36512>



This process can be extended to consider the hardware configuration especially management interfaces and the placement/zoning of these infrastructure components in the various operator environments (test, pre-production, production etc.),

Privileged Access Workstations

The concept of a PAW is to ensure that the management plane is only accessible from trusted and authenticated accesses. As such controls can be considered such as:

- Ensuring a PAW has no internet access with no wider access to corporate systems and services
- The PAW can be built with 'bottom to top' (see next section) security principles in mind such as secure boot, boot-attestation, data-at-rest encryption backed by a hardware root-of-trust
- The PAW can be further hardened through ensuring up-to-date security patching of both applications and underlying operating systems and can be configured to prevent the execution of unauthorised code
- It is recognised that operationally this can result in the need for a second laptop for engineers / administrators and investment in network separation, however, the consequent improvement in security controls is significant.

Bottom to top security

Consider, below, a generic implementation with an underlying open hardware platform supporting a virtualisation layer (e.g. OpenStack) in turn supporting multiple VNFs each from differing vendors / supply routes.

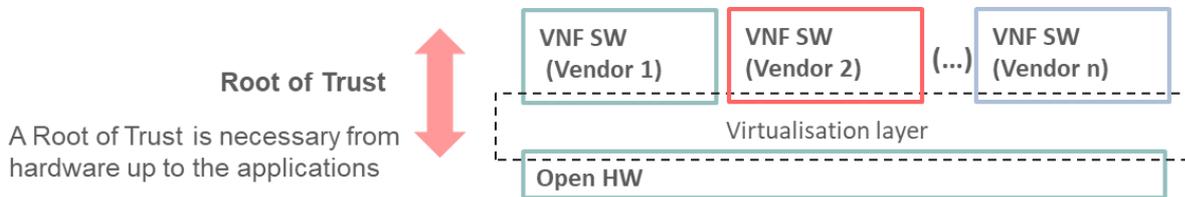


Figure 21 The bottom to top root of trust

Software implements the functionality of the unit. The code can be proprietary and contain open source components and may contain commercially supported open source virtualisation software to allow interfacing between the code and the supporting open hardware or Cloud infrastructure. Differing architectural decisions result in variances in the levels of abstraction and separation between workloads within virtualisation fabrics. Each respective layer may have security controls yet it is important that the implementation works together to implement a coherent security solution.

A coherent security arrangement is vital to protect against attack of the infrastructure fabric, hypervisor and management and orchestration (MANO). Poor security could enable an attacker to adversely influence the entire virtualisation fabric, including all hosts and virtual workloads.

A VM can provide a full abstraction layer between each virtual host and the supporting hardware. This abstraction layer provides each VM with their own resource pools, with usually no shared resources with other VMs. In some instances, the hypervisor can provide short-cuts to allow specific VMs to address the hardware directly. These short-cuts can have a notable effect on the security boundaries of the virtualised solution as a VM may now be able to directly access and control physical hardware without any of the hypervisor's security controls. It is advisable to treat these arrangements as their own specific lower trust domain.

Containers provide a process-level separation between workloads which can make them fast and cheap to deploy. The underlying kernel and resource scheduling is shared between every container running on the host within the same trust domain. However, a single kernel-level vulnerability might allow an attacker to impact the underlying host and as such all concurrent containers.

Root of Trust

Preservation of a root of trust is essential in protecting the integrity of the solution from 'top to bottom'. The correct workload code should be running through the correct virtualisation platform (with any appropriate workload placement) through operating system security functions (such as SELinux) from any underlying trust arrangements. A range of associated approaches apply in this space including Secure Boot³⁶, Remote Attestation³⁷ and Trusted Platform Modules (TPM).

The Cyber Security Body of Knowledge Hardware Security Knowledge Area³⁸ explores a number of these aspects and there is also content from the Trusted Computing Group³⁹. Infrastructure design can seek to include these approaches as design criteria and can then be considered on the extent to which given technical designs and vendor products support these approaches.

A TPM is a hardware component that can provide additional security capabilities but is not always implemented by all manufacturers. The TPM supports hardware-based cryptographic operations. System security functions can then leverage the TPM for a range of purposes notably ensuring Secure Boot.⁴⁰

Some vendors are considering introducing White Box Cryptography⁴¹ (WBC) which uses encryption and obfuscation to secure software keys and critical data inside their RAN applications. Suitable risk assessment will assist in deciding the degree to which this is a suitable security approach and the extent to which additional controls may be required.

³⁶ Secure Boot allows the system to boot into a defined and trusted configuration

³⁷ Remote attestation allows a trusted platform to present reliable evidence to remote parties about the software that is running

³⁸ https://www.cybok.org/media/downloads/Hardware_Security_issue_1.0.pdf

³⁹ <https://trustedcomputinggroup.org/>

⁴⁰ Extract from draft CNTT Cloud Infrastructure Reference Model:

Static Root of Trust Measurement (SRTM) begins with measuring and verifying the integrity of the BIOS firmware. It then measures additional firmware modules, verifies their integrity, and adds each component's measure to an SRTM value. The final value represents the expected state of boot path loads. SRTM stores results as one or more values stored in PCR storage. In SRTM, the CRTM resets PCRs 0 to 15 only at boot. The Platform Configuration Register (PCR) is a memory location in the TPM used to store TPM Measurements.

Using a Trusted Platform Module (TPM), as a hardware root of trust, measurements of platform components, such as firmware, bootloader, OS kernel, can be securely stored and verified. Cloud Infrastructure operators should ensure that the TPM support is enabled in the platform firmware, so that platform measurements are correctly recorded during boot time.

In Dynamic Root of Trust for Measurement (DRTM), the RTM for the running environment are stored in PCRs. Cloud Infrastructure operators must ensure that remote attestation methods are used to remotely verify the trust status of a given Cloud Infrastructure platform. The basic concept is based on boot integrity measurements leveraging the Trusted Platform Module (TPM) built into the underlying hardware.

⁴¹ <https://www.cryptoexperts.com/technologies/white-box/> A white-box technology consists of a program-generating compiler that, for some specific cryptographic algorithm, takes as input a secret key and produces a white-box secure program that implements the cryptographic algorithm with the specified secret key. Anyone in control of the generated program can execute it on any input and get the expected output, but is unable to learn anything more than such input-output pairs. The white-box program remains unintelligible and securely hides the secret key, just as trusted hardware would.

A Zero Trust Approach

The tenets of Zero Trust are described below in italics from source⁴² :

- *All data sources and computing services are considered resources.*
- *All communication is secured regardless of network location.*
- *Access to individual enterprise resources is granted on a per-connection basis.*
- *Access to resources is determined by dynamic policy - including the observable state of client identity, application, and the requesting asset - and may include other behavioral attributes.*
- *The enterprise ensures that all owned and associated devices are in the most secure state possible and monitors assets to ensure that they remain in the most secure state possible.*
- *All resource authentication and authorization are dynamic and strictly enforced before access is allowed.*
- *The enterprise collects as much information as possible about the current state of network infrastructure and communications and uses it to improve its security posture.*

The National Institute of Standards and Technology (NIST) Special Publication⁴³ describe a Zero Trust Architecture as follows:

A zero trust architecture (ZTA) uses zero trust principles to plan industrial and enterprise infrastructure and workflows. Zero trust assumes there is no implicit trust granted to assets or user accounts based solely on their physical or network location (i.e., local area networks versus the internet) or based on asset ownership (enterprise or personally owned). Authentication and authorization (both subject and device) are discrete functions performed before a session to an enterprise resource is established.

Adopting a zero trust approach⁴⁴ for the creation of Trust Relationships between trust domains, between and within the system, can add additional control layers to limit lateral movement and cascaded compromises.

⁴² From: COMMUNICATIONS SECURITY, RELIABILITY, AND INTEROPERABILITY COUNCIL VII - REPORT ON RISKS INTRODUCED BY 3GPP RELEASES 15 AND 16 5G STANDARDS; September 16, 2020

⁴³ National Institute of Standards and Technology (NIST) NIST Special Publication 800-207 Zero Trust Architecture

⁴⁴ See also <https://www.ncsc.gov.uk/blog-post/zero-trust-principles-beta-release>

System Hardening

Considerations for 'hardening' the deployed system can include:

- All compute infrastructure to support a root of trust and secure boot
- Architecturally adopt the principles of zero trust and trust domain separation
- From a secure by design⁴⁵ principle, adopt the principles of least privilege and strict limitation of the necessity for allowing root access
- Remove or disable unnecessary services, applications and network protocols, configuring resource controls and testing the security of the Operating System.
- Ensure that all the platform's components are kept up to date with the latest patching
- Strictly control access to resources and protect them from malicious access by utilising and configuring suitable operating system security modules
- All systems part of infrastructure to support password hardening
- Regular monthly, quarterly etc. hardening checks and patching (vulnerability scanning) compliance scanning

Many of these concepts are explored in the CNTT⁴⁶ Cloud Infrastructure Reference Model that is currently under development.

⁴⁵ <https://www.ncsc.gov.uk/information/secure-by-default-platforms> & <https://blog.threatpress.com/security-design-principles-owasp/>

⁴⁶ CNTT NG.126 CR1001 <https://cنت-n.github.io/CNTT/doc/>

Virtualisation Layer Code

Consideration is needed to the virtualisation layer of code that can, in many cases, be of open source code origin. As discussed in previous reports, this has both positive and negative security considerations.



Figure 22 The virtualisation layer

Vendors of packaged solutions may, in turn, have long term support agreements with open source vendors (e.g. Red Hat) and continue to use the versions long past the public support date. Subsequently, scanning tools can then identify these products as obsolete and vulnerable and cannot determine whether backported patches have been applied to secure the versions.

In the Research paper⁴⁷, one of the approaches that can help to mitigate this is ensuring an SBOM is available. The National Telecommunications and Information Administration⁴⁸ defines an SBOM as:

“a formal record containing the details and supply chain relationships of various components used in building software An SBOM is effectively a nested inventory: a list of ingredients that make up software components. An SBOM identifies and lists software components, information about those components, and the relationships between them.

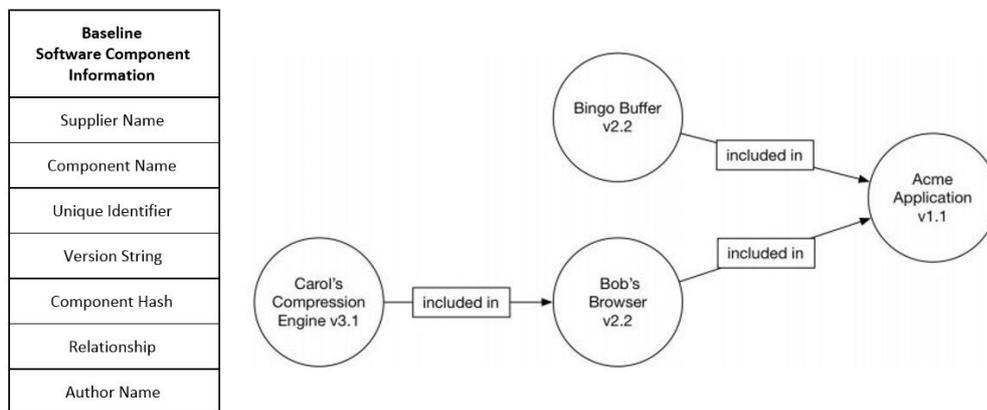


Figure 1: The baseline SBOM includes components in their assembled relationship. Each component has enough information to “uniquely and unambiguously identify” it (left), and the relationship of what upstream or child components are “included in” downstream or parent components (right).

Data standards exist today that can capture this SBOM data. These include SPDX, SWID, and CycloneDX.” The NTIA SBOM website contains a range of useful information on establishing an SBOM, a report on data formats and a proof of concept paper.

When a new vulnerability is discovered it is vital to know whether it is applicable to your own network and where in the infrastructure it is deployed. Systematic inventory that includes deployed software increases cyber security and can lower operational cost through efficient remediation actions. The attraction of using more mainstream vendor distributions of code is understandable as it plays to mitigate against support, indemnity and integration risks. However, the distributions are sometimes slower in pace (to build in the very advantages required) which can lead to vulnerabilities to be exposed for longer periods. Equally, it can provide a single point of focus for operators to leverage a particular software fix to be progressed (not always

⁴⁷ GSMA document Open Source Software Security Research Paper December 2020

⁴⁸ NTIA <https://www.ntia.gov/SBOM>

possible where there is reliance on community support). Hence, mandating and updating an SBOM for deployed software and obtaining it in a standard data format can help support better security responses.

Software Composition Analysis

Software Composition Analysis (SCA) is an increasingly important approach to code assessment. Techniques vary from:

- Static code testing
- Composition Analysis
- Fuzz testing
- Dynamic Application Security Testing

The earlier GSMA Report⁴⁹ included reference to a number of organisations active in considering SCA approaches. These included:

- The OWASP Dependency Check⁵⁰
- SAFECODE's Report: Managing Security Risks Inherent in the Use of Third-party Components⁵¹
- Linux Networking Foundation's OpenChain tool⁵²
- Synopsys' SCA tools⁵³ and the Open Source Security and Risk Analysis paper⁵⁴
- Whitesource's SCA: how to choose the right solution⁵⁵ and the Complete Guide to open source security⁵⁶

CNCF have an interesting tool – FOSSA. FOSSA claims to manage 'Visibility of 3rd party code', 'prioritise problematic dependencies', automatically compile compliance reports' (e.g. Bill of Materials) and 'Streamline license & vulnerability remediation'.

⁴⁹ GSMA Report Open Source Software Security - Research Summary; v1.0 Aug 2020

⁵⁰ Dependency-Check is a Software Composition Analysis (SCA) tool that attempts to detect publicly disclosed vulnerabilities contained within a project's dependencies. It does this by determining if there is a Common Platform Enumeration (CPE) identifier for a given dependency. If found, it will generate a report linking to the associated CVE entries.

⁵¹ https://safecode.org/wp-content/uploads/2017/05/SAFECODE_TPC_Whitepaper.pdf

⁵² The LF Openchain initiative maintains the industry-standard for the key requirements of a quality open source compliance program. It seeks to make open source license compliance simpler and more consistent

⁵³ <https://www.synopsys.com/software-integrity.html>

⁵⁴ <https://www.synopsys.com/blogs/software-security/5-open-source-trends-2020-ossra/>

⁵⁵ <https://resources.whitesourcesoftware.com/white-papers-datasheets/how-to-choose-an-open-source-management-solution>

⁵⁶ <https://resources.whitesourcesoftware.com/white-papers/the-complete-guide-on-open-source-security>

Common Vulnerabilities and Exposures (CVEs)

The National Vulnerability Database (NVD)⁵⁷ is a really valuable tool for understanding the vulnerabilities in the code that an enterprise uses but did not write themselves, especially open source and general purpose third party code. The U.S. government repository of standards-based vulnerability management data enables automation of vulnerability management, security measurement, and compliance. The NVD includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics. The NVD performs analysis on CVEs that have been published to the CVE Dictionary⁵⁸. NVD analyse CVEs to produce impact metrics (Common Vulnerability Scoring System CVSS), vulnerability types (Common Weakness Enumeration (CWE), and applicability statements (Common Platform Enumeration (CPE), as well as other pertinent metadata. CVSS scores can be used by operators in contracts to mandate a service level for vendors to respond for bug fixes in an acceptable period of time. Often this takes the form of CVSS score bands aligning to different Service Level Agreement (SLA) responses; higher CVSS bands mandate a faster response time.

Additional considerations here include:

- Scan software image files for known vulnerabilities in repositories before they get deployed into production
- Scan processes and containers in the runtime environment and block vulnerable processes and prevent vulnerable containers from being exposed
- Scan configuration files for any unauthorised changes.

Some industries have found that it can be effective to offer a 'bug bounty' programme where rewards are offered to those spotting and reporting code bugs and / or vulnerabilities.

⁵⁷ <https://nvd.nist.gov/>

⁵⁸ <https://cve.mitre.org/>

Interoperability

Integration and test is valuable to ensure the security and integrity of the solution when considering that a given platform may host a number of different functions from different vendors (as shown below).

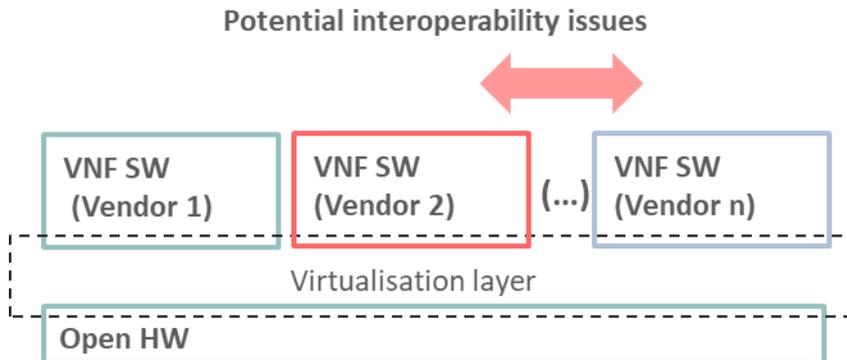


Figure 23 Potential security interoperability issues

Lack of standards can make it difficult to integrate networking software apps (e.g. microservices) from different suppliers on the same open hardware. Open source projects may be a solution for this, as they become “de facto standards” for the virtualisation layer (see later section on CNTT).

Interfacing

Interfacing arrangements include 3GPP Defined, Proprietary, Consortia Defined (e.g. eCPRI), Open (e.g. O-RAN) and practical deployment considerations such as Open Fronthaul Gateways. 3GPP interfaces have high scrutiny and ‘secure by design’ approaches with a relatively long cycle time. Community interfaces currently have limited constructs to evaluate and expose to security review. Interfaces that move to open specification from proprietary may not have the ‘wrapper’ of a singular supplier, i.e. the proprietary code may benefit from a wider security wrapper that may be absent in a different specification. A range of specification approaches are noted below:

- International Standards Interface Specs (e.g. ETSI, 3GPP, IETF)
- Community/industry specified (e.g. eCPRI)
- Proprietary solution (e.g. vendor internal specification).

Application Programming Interface

The Cyber Security Body of Knowledge section on Software Security Knowledge Area⁵⁹ includes a useful overview as follows:

An Application Programming Interface, or API, is the interface through which one software component communicates with another component, such as a software library, operating system, web service, and so forth. Almost all software is programmed against one or more pre-existing APIs. An API comes with an (explicit or implicit) specification/contract of how it should be used and what services it offers, and just like the contracts we considered in previous subsections, violations of these contracts can often have significant consequences for security. If the client of the API violates the contract, the software system again enters an error-state, and the further behaviour of the software system will depend on implementation details of the API, and this may allow an attacker to break the security objective of the overall software system.

The OWASP⁶⁰ Top10 for API security is a useful source for implementation security vulnerabilities.

⁵⁹ https://www.cybok.org/media/downloads/Software_Security_issue_1.0_1M7Kfk2.pdf

⁶⁰ <https://owasp.org/www-project-api-security/>

Approaches to address these concerns include comprehensive API-testing and error-state handling, use-case testing, 'plugfests', architectural considerations such as limiting deployments to parts of network infrastructure, threat and risk based modelling, re-use of proven design combinations and strong systems integration testing.

Application Based Segmentation

In virtualised environments remote connections are typically terminated upon the entry into the data center rather than within the virtualised network. Therefore, Service Providers must strongly police the connectivity by adding new controls.

One method to securely deploy application-based segmentation is to enable identity checks. In this method, a process authorisation is injected in-between the application components that are triggered for each communication attempt. As an example, all connections between front-end and back-end processes would be authorised and monitored according to a security policy.

Palo Alto⁶¹ describe: *Identity-Based Microsegmentation protects cloud applications from attack by authenticating and authorizing all communications with a cryptographically-signed identity assigned to every workload. Microsegmentation alleviates reliance on unmanageable, error-prone policies based on IP addresses. It enforces a distributed, homogeneous security policy per workload independent of network or infrastructure configuration, enabling uniform security orchestration across multi-cloud environments.*

⁶¹ See <https://www.paloaltonetworks.com/prisma/cloud/identity-based-microsegmentation>

Cloud iNfrastructure Telco Task Force

CNTT was incubated in early 2019 through a partnership between GSMA and the Linux Foundation as a global open source taskforce comprised of industry-leading network providers and NFVI/VNF suppliers. CNTT provides standardised infrastructures for both virtual machine-based and cloud native network functions, making it possible to deploy multiple network functions without having to create new infrastructures for each. This standardisation enables providers to shorten deployment and onboarding from weeks and months to hours and days, reducing costs and accelerating digital transformation. From a security standpoint, the ability to invest significant testing and validation into a smaller number of infrastructure combinations then re-use this standard build in other deployments can lower the overall security risk across a portfolio of deployments⁶².

All of this had led to a growing awareness of the need to develop more open models and validation mechanisms to bring the most value to telco operators as well as vendors, by agreeing on a standard set of infrastructure profiles to use for the underlying infrastructure to support VNF applications across the industry and telecom community at large. To achieve this goal, the cloud environment needs to be fully abstracted via APIs and other mechanisms to the VNFs so that both developers of the VNF applications and the operators managing the environments can benefit from the flexibility that the disaggregation of the underlying infrastructure offers.

One of the main targets of the CNTT is to define an agnostic cloud infrastructure, to remove any dependencies between workloads and the deployed cloud infrastructure, and offer infrastructure resources to workloads in an abstracted way with defined capabilities and metrics. This means, operators will be able to host their Telco workloads (VNFs/CNFs) with different traffic types, behaviour and from any vendor on a unified consistent cloud infrastructure. The use of a consistent approach such as CNTT has the advantage of being replicable across multiple deployments, consistent documentation and skills. CNTT have 2 Reference Architectures : OpenStack-based and Kubernetes-based⁶³. GSMA document Cloud Infrastructure Reference Model⁶⁴ contains significant technical detail on these considerations and is recommended reading.

⁶² Assuming similar risk appetite, threat exposure, etc

⁶³ https://cntt-n.github.io/CNTT/doc/ref_arch/

⁶⁴ GSMA Official Permanent Reference Document NG.126, Cloud Infrastructure Reference Model Version 1.0 11 November 2020

Broader Cloud Considerations

Cloud Root of Trust

There are various options for a root of trust approach in cloud infrastructure. For example, the root of trust in AWS EC2 is Nitro⁶⁵ and Google Cloud has a Shielded VM⁶⁶ approach, which provides secure and measured boot and vTrusted Platform Modules. The cloud-native approach is slightly different as less focus is placed on a single root of trust because other aspects of the architecture can be better protected. For example, the cloud platform's control plane provides each instance with cryptographic credentials for that VM's identity. This can then be used to get secure access to cryptographic keys and managed secrets. This allows use of the control plane as a root of trust, since that then provides access to things like the certificates used to authenticate and encrypt network traffic. The same identity can also be used to gain secure access to managed cloud services, such as managed databases, storage services, and message queues.

ETSI

The ETSI standard *Interface to offload sensitive functions to a trusted domain*⁶⁷ provides extra security requirements for public clouds to offer operators the option of running public telecom network functions in public clouds. The standard provides extra security for sensitive functions down to individual Virtual Machines. It introduces a trust hierarchy onto the flat admin architecture of public clouds so that only a subset of telco engineers or processes can access these sensitive functions. See for further explanation *ETSI Secure Public Clouds for Telcos*⁶⁸. See also ETSI Container Security Specification⁶⁹.

NIST

Additionally, the National Cybersecurity Center of Excellence (NCCoE) at NIST have an initiative developing a practice guide for Trusted Cloud: VMware Hybrid Cloud IaaS Environments⁷⁰.

The Center for Internet Security (CIS)

The Center for Internet Security⁷¹ also has a range of useful benchmarks for a range of platform approaches including Google Cloud, Oracle Cloud, Microsoft Azure, Kubernetes, Docker, Amazon Web Services, Red Hat Linux, VM Ware and Ubuntu Linux. These benchmarks can be used to validate infrastructure is configured as securely as possible. There are open source⁷² and commercial⁷³ tools that can check environments against the recommendations defined in the CIS Benchmark to identify insecure configurations. This baseline of controls can be used to build on to ensure the best possible secure configurations are deployed for a range of areas such as levels of privilege, image deployment options, enabling Transport Layer Security (TLS) and limiting port exposure.

Cloud Supply Chains

It is important to consider the supply chains used by Cloud service providers and other aspects such as data localisation regulations and performance and security of the cloud infrastructure. A recent report⁷⁴ discusses this in the context of geopolitics of cloud computing (see global distribution graphic below).

⁶⁵ <https://aws.amazon.com/ec2/nitro/>

⁶⁶ <https://cloud.google.com/shielded-vm>

⁶⁷ ETSI TS 103 457 CYBER; Trusted Cross-Domain Interface: Interface to offload sensitive functions to a trusted domain

⁶⁸ HardenStance Briefing No.22, 28th March 2019 "ETSI Secures Public Clouds for Telcos"

⁶⁹ ETSI GS NFV-SEC 023: "Container Security Specification"

⁷⁰ See preliminary draft at <https://www.nccoe.nist.gov/projects/building-blocks/trusted-cloud/hybrid>

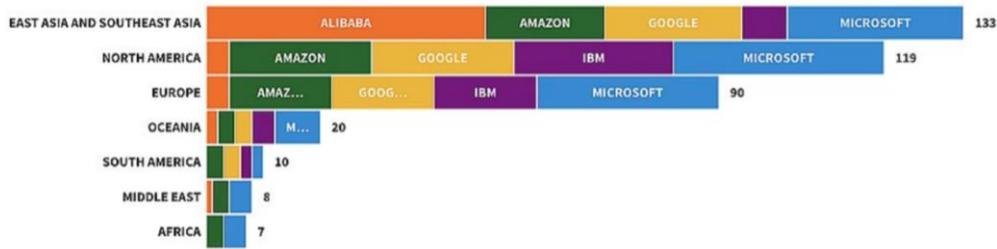
⁷¹ <https://www.cisecurity.org/resources/page/4/?type=benchmark>

⁷² E.g. <https://github.com/docker/docker-bench-security>

⁷³ See <https://www.cisecurity.org/cis-controls-supporters/>

⁷⁴ <https://www.atlanticcouncil.org/wp-content/uploads/2020/09/CLOUD-MYTHS-REPORT.pdf>

Figure 11: Global Distribution of Cloud Provider Data Centers



Source: Lily-Zimeng Liu

There is potential for operators to use cloud procurement contracts to identify cloud provider details for detailed risk management plans, information on hardware vendor choices, incident reporting and performance data. This data can inform vendor selection and maintenance.

The security benefits of selecting a good cloud service are explored in a recent blog post⁷⁵.

⁷⁵ <https://www.ncsc.gov.uk/whitepaper/security-benefits-of-a-good-cloud-service>

GSMA Industry and Security Standards Activity Areas

GSMA offers its members considerable security⁷⁶ expertise and services through a range of activity areas.

Fraud & Security Working Groups

The GSMA's Fraud and Security Group⁷⁷ drives the association's management of fraud and security matters related to mobile technology, networks and services, with the objective to maintain or increase the protection of mobile operator technology and infrastructure and customer identity, security and privacy such that the industry's reputation stays strong and mobile operators remain trusted partners in the ecosystem. FASG provides an open, receptive and trusted environment within which fraud and security intelligence and incident details can be shared in a timely and responsible way. Members gain from the significant body of knowledge published on fraud and security matters.

Securing the 5G Era⁷⁸

5G has designed in security controls to address many of the threats faced in today's 4G/3G/2G networks. These controls include new mutual authentication capabilities, enhanced subscriber identity protection, and additional security mechanisms. 5G offers the mobile industry an unprecedented opportunity to uplift network and service security levels. 5G provides preventative measures to limit the impact to known threats, but the adoption of new network technologies introduces potential new threats for the industry to manage. GSMA explores a range of security considerations including Secure By Design, 5G deployment models and 5G Security Activities.

Telecommunication Information Sharing and Analysis Center

The GSMA Telecommunication Information Sharing and Analysis Center⁷⁹ is the central hub of information sharing for the Telecommunication Industry. Driven by the ethos "One organisation's detection is another's prevention", we believe information sharing is essential for the protection of the mobile ecosystem, and the advancement of cybersecurity for the telecommunication sector. Drawing on the collective knowledge of mobile operators, vendors and security professionals, the T-ISAC collects, disseminates information and advice on security incidents within the mobile community – in a trusted and anonymised way.

Coordinated Vulnerability Disclosure Programme

The GSMA Coordinated Vulnerability Disclosure⁸⁰ programme gives security researchers a route to disclose a vulnerability impacting the mobile ecosystem meaning the impact can be mitigated before it enters the public domain. We work with mobile operators, suppliers and standards bodies to develop fixes and mitigating actions to protect customers' security and trust in the mobile communications industry.

Security Accreditation Scheme

The Universal Integrated Circuit Card (UICC) in mobile devices, and its applications and data play a fundamental role in ensuring the security of the network, the subscriber's account and related services and transactions. The GSMA's Security Accreditation Scheme⁸¹ enables mobile operators to assess the security of their UICC and Embedded UICC (eUICC) suppliers, and of their eUICC subscription management service providers.

⁷⁶ <https://www.gsma.com/security/>

⁷⁷ <https://www.gsma.com/aboutus/workinggroups/fraud-security-group>

⁷⁸ <https://www.gsma.com/security/securing-the-5g-era/>

⁷⁹ <https://www.gsma.com/security/t-isac/>

⁸⁰ <https://www.gsma.com/security/gsma-coordinated-vulnerability-disclosure-programme/>

⁸¹ <https://www.gsma.com/security/security-accreditation-scheme/>

Network Equipment Security Assurance Scheme

The Network Equipment Security Assurance Scheme⁸², jointly defined by 3GPP and GSMA, provides an industry-wide security assurance framework to facilitate improvements in security levels across the mobile industry. NESAS defines security requirements and an assessment framework for secure product development and product lifecycle processes, as well as using 3GPP defined security test cases for the security evaluation of network equipment.

NESAS provides a security baseline to evidence that network equipment satisfies a list of security requirements and has been developed in accordance with vendor development and product lifecycle processes that provide security assurance. NESAS is intended to be used alongside other mechanisms to ensure a network is secure, in particular an appropriate set of security policies covering the whole lifecycle of a network. The scheme should be used globally as a common baseline, on top of which individual operators or national IT security agencies may want to define additional security requirements.

⁸² <https://www.gsma.com/security/network-equipment-security-assurance-scheme/>

Physical Security

Physical security of infrastructure remains a vital component of any layered security approach; especially as more open networking may result in a greater range of vendors requiring physical access to operational equipment.

Physical security strategies include considerations such as:

- Deter: to stop or re-direct the attack
- Detect: to verify an attack and respond
- Delay: to slow the progress of the attack
- Mitigate: to minimise the consequences of an attack
- Response: actions to prevent the attack completing and identify learning and improvement areas

Any security strategy needs to be proportionate to the threat so the threat assessment topic discussed earlier is applicable to physical security too. Security controls can then play through into:

- Asset and site design
- Access control, alarms and locks
- Active access delay systems
- Security staff
- Intruder detection and tracking
- Physical defences including security walling and perimeter defence
- Building entry points
- Control rooms
- Unmanned Autonomous Vehicle defence
- Power, lighting and HVAC resilience

Therefore, considerations of these physical security strategies are important considerations when potentially significantly densifying cell sites to deliver increased coverage, higher bandwidth and lower latency services. The increased number of sites and vendors makes the physical security challenge ever more important.

Personnel Security

Personnel security is a key consideration when considering real-world threat actors attack techniques for any network whether it contains open source code or not. In the early stages of an attack, target reconnaissance and targeting are a typical first step.

Targeting techniques can focus on individuals sharing useful information on social media platforms such as LinkedIn. Useful information can include details of equipment where the individual has skills. This might identify specific equipment in use which may have known vulnerabilities that can be exploited if not patched or otherwise protected. Information might be shared that indicates an individual's job role includes Administrator access or skills. This can be used to target an individual for phishing or other cyber-attacks.

Successful compromise of such an individual's account can provide persistent presence and opportunity to directly or indirectly access critical systems or data. Attack modelling or Holistic Penetration test (see earlier section) exercises can be useful in assessing the defences and available information in place for key administrator roles in the operational network. Remedial action can be made to encourage changes in social media postings or other publicly available information. Hence, social media security awareness and regular attack modelling are a key activity for an organisation to help protect its staff.

In the earlier section on Management Plane Security, the concepts of least privilege, separation of duties and privileged access workstations were identified as key additional protective steps to make it much harder for any attack that breaches perimeter attacks (e.g. phishing or malware attacks targeting administrator accounts). Implementing these internal controls makes lateral movement and escalation of privilege much harder to achieve.

Hence, personnel security considerations are part of any layered security defensive approach, especially allied to physical, architectural and detailed controls identified earlier.

Bringing it together

This report has identified a range of those developing controls and described within the contexts of systems, component and infrastructure.

The system security control aspects can be summarised in the system security 'wall' shown below.

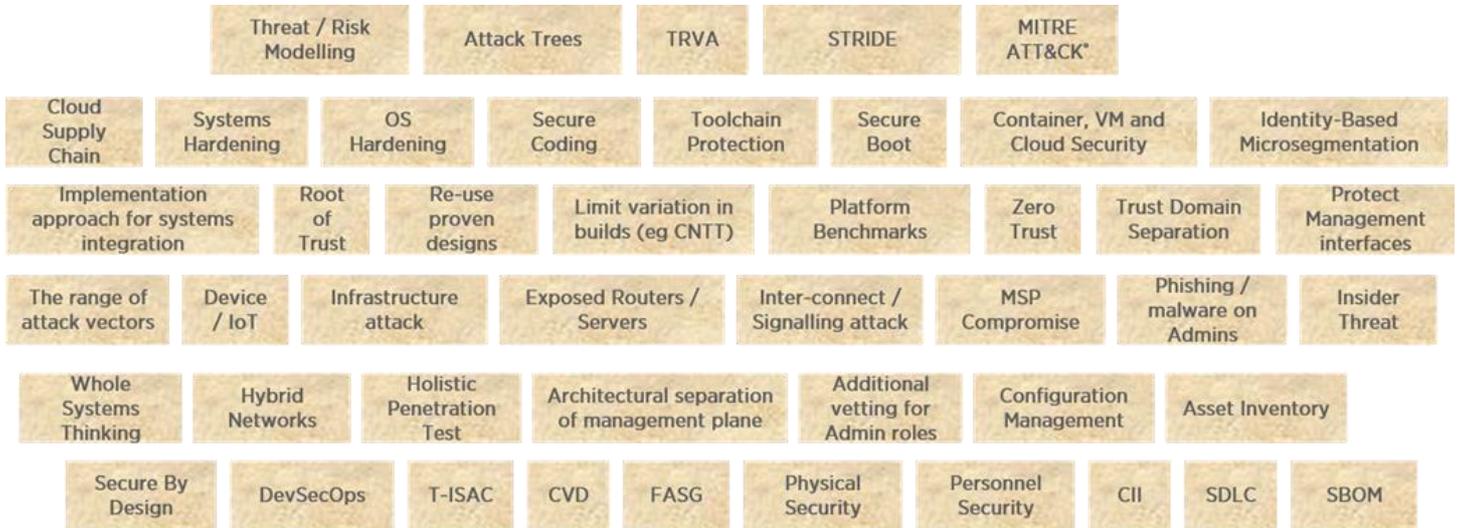


Figure 24 A 'wall' of layered system level controls

Consideration of component level controls can be summarised in the Component Security 'Wall' shown below.



Figure 25 A 'wall' of component level controls

Combining the systems and component level considerations can build a framework for considering the design and operation of open networks. The system and component lifecycles can be combined to illustrate their co-dependence and cyclic nature. The cycle time for each lifecycle will be notably different (i.e. the system lifecycle is likely to be slower) and the number of cycles undertaken in a system lifetime will be different (i.e. there is likely to be many more cycles of the component lifecycle). The application of the different controls will vary depending on where any specific change activity is taking place and at what level of change granularity. The diagram below illustrates a combination of both systems and component lifecycles (excluding a Decommission stage).

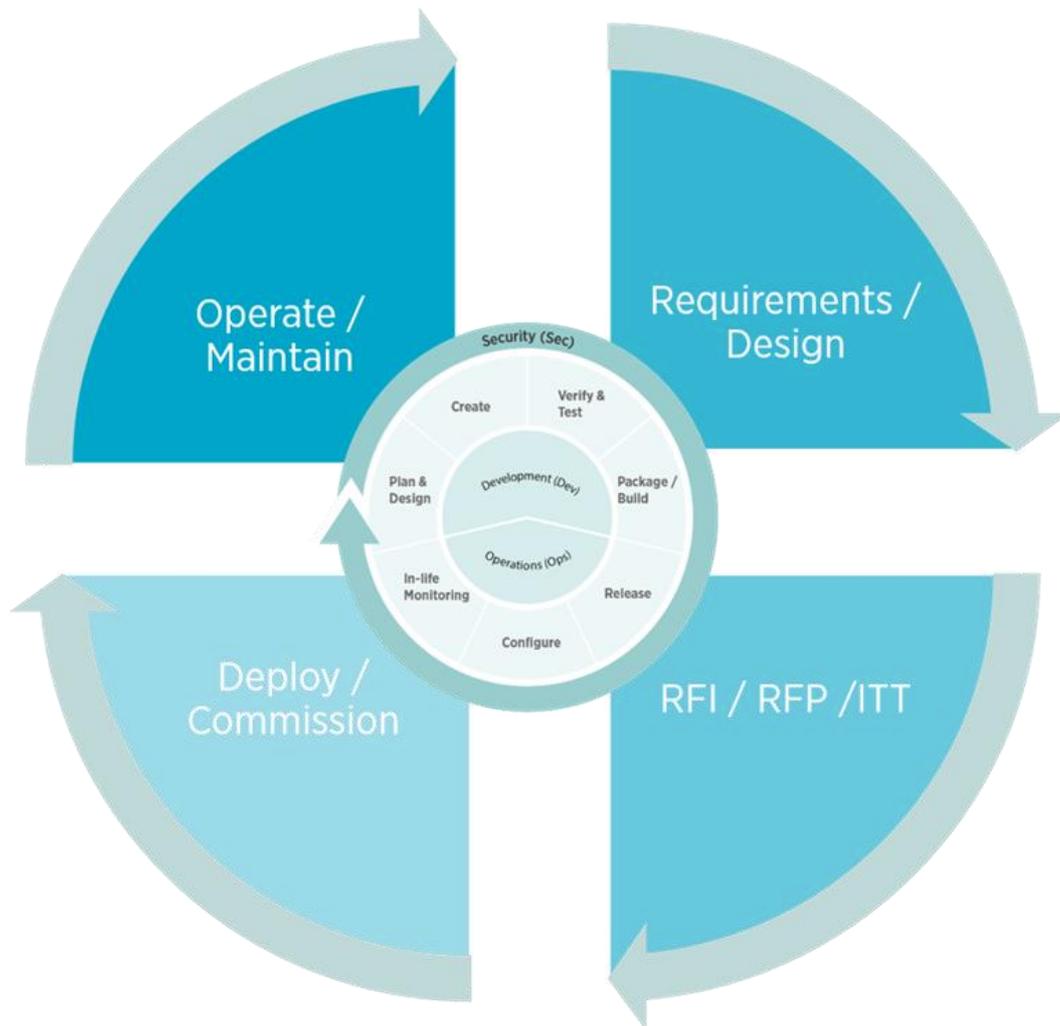


Figure 26 A combination of systems and component lifecycles

The following considers the application of the various security approaches ('bricks in the wall') outlined in the paper. As approaches will vary significantly and to make it more accessible, it is not an attempt to exhaustively list security functions. Rather it is an attempt to illustrate how some of the security controls can be applied through a lifecycle perspective.

Requirements / Design

Actions to implement high levels of security are best achieved through actions early in the lifecycle such that later detailed actions are possible. For example, if the system design does not include a platform that supports a trusted platform module, then that control cannot be implemented. Thus a significant proportion of the controls are identified in the earlier stages of system implementation.

Secure by design is the major theme of this section; the premediated consideration of security within the system as the design stage such that security is able to be delivered through implementation and into service. As such, it can be advantageous to consider layered security controls that build in an additive collective high security approach. Whole systems thinking is an important step here such that existing or legacy equipment vulnerabilities cannot be used to undermine security within newer system component including consideration of the total security arrangement for hybrid networks.

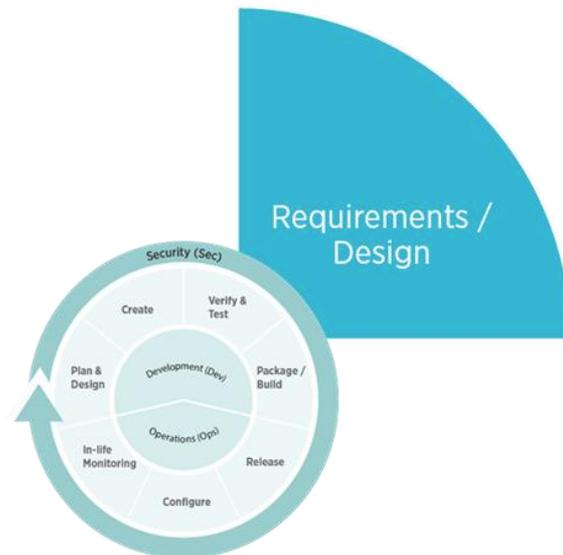


Figure 27 The component lifecycle overlapping with the systems Requirements / Design phase

Infrastructure Considerations (VM, Container, Cloud) can set a strong foundation as can OS Selection (including support for security features). An option is to base the design to re-use proven designs and to consider CNTT builds that may have been deployed successfully in other environments.

Platform benchmarks are a useful basis to consider for infrastructure. Identity-Based Microsegmentation is a feature that may form part of an infrastructure system design. Support for a White Box cryptographic solution may form part of a wider security solution.

Designing the system to deliver a bottom-to-top security with a Root of trust, Zero trust principles and Trust Domain separation can form the basis for later security controls. At this stage, one can use risk / threat assessment to deliver an Impact Assessment of architectural design choices and the overall cost base.

Architectural separation of management plane allows stronger security regimes to be delivered allied with the ability to deliver privileged access management supported by privileged access workstations. Design consideration should be given to include protection for management interfaces and the ability to deliver passive scanning of internet-connected equipment.

Account authentication mechanisms, password management systems and including a multi-factor authentication design will aid operational security.

Consideration of the resulting attack vectors is important and can include physical access, physical Infrastructure, air interface, exposed compute, Interconnect / signalling, MSP Compromise, Phishing / malware on Admins and Insider Threats. This consideration can be used to limit the range of attack vectors and surface of the resulting system

GSMA Permanent Reference Documents (PRDs) can provide a useful reference for security considerations.

Secure Coding

The following topics can be considered directly at the component level or included as design requirements that might be included in a systems-level procurement stage.

Toolchain protection is an important consideration when considering a particular tool or software build. If toolchains are insecure, a significant vulnerability is created and should be considered at this early stage. Similarly, it is important to have the ability to protect software / container images taken from a registry/repository so support for this should be designed in. A Secure Boot arrangement and support for Remote Code attestation are also important.

The OWASP Top 10 Web Application Security Risks identifies important areas to avoid when implementing code. When considering inclusion of a particular software package or distribution with open source origins, it is important that there is a track record in delivering open source code maintenance. Open source software can be judged against the distribution being developed in one of the higher levels of the Core Infrastructure Initiative⁸³ and that a Software Design

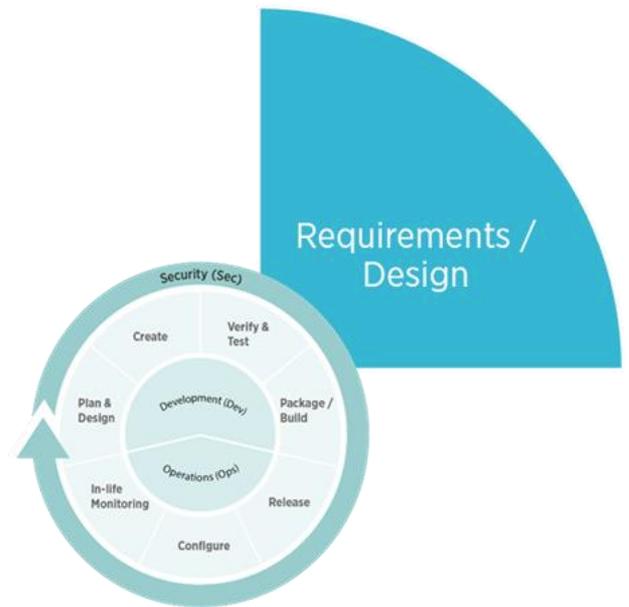


Figure 28 The component lifecycle overlapping with the systems Requirements / Design phase

Life Cycle has been followed.

Support for a White Box cryptographic solution may form part of a wider security solution.

Requirements should include the necessity to use open source naming standards (such as SWID, SPDX, CycloneDX) and the production (& maintenance of a Software Bill of Materials).

Consideration can be given to a particular vendor's certification status against GSMA's Network Equipment Security Assurance Scheme. This can be used as part of a requirement set that builds a holistic security approach.

⁸³ <https://www.coreinfrastructure.org/>

RFI / RFP / ITT

Assess the risk coverage provided by procurement responses and use this to understand the likely cost of residual security risk. This can support the development of a business case. Selection of an implementation approach needs to include the impact on in-house skills and in-life support for security.

SAFECode⁸⁴ generated a useful set of questions that can be used as part of an assessment of the effectiveness of contribution into open source communities. These include:

- *Does the community/supplier provide clear vulnerability/patch reporting methods, to include reporting to commonly used repositories (e.g., CVE ID in the National Vulnerability Database), and provide frequent feedback on submitted vulnerabilities?*
- *Is there a dedicated website for security issues and is there a way to (privately) submit security patches?*
- *Does the supplier perform automated security testing of the components, both periodically and on an ongoing basis?*
- *Do the supplier's automated standards-based assessment tools utilize public vulnerability and security flaw repositories (Common Weakness Enumeration, CVE, Common Attack Pattern Enumeration and Classification, etc.)?*
- *Does the community/supplier routinely disclose vulnerabilities and prepare customers for patch deployment?*
- *Does the community/supplier have a history and reputation for actively patching reported vulnerabilities?*
- *Does the component have a regular maintenance and update cycle?*
- *Does the component have a clearly defined and consistent set of maintainers?*
- *What controls does the supplier have to protect against unapproved changes/updates?*
- *What is the expected lifetime of the component?*
- *What criteria or process will be used to determine when to update the component?*
- *How much documentation is available on the component, and what is the quality of that documentation?*
- *How long has the component existed and when was the last major release?*
- *How widely used is this component both publicly and within your organization?*
- *What is the reputation of the component, author, supplier or community?*

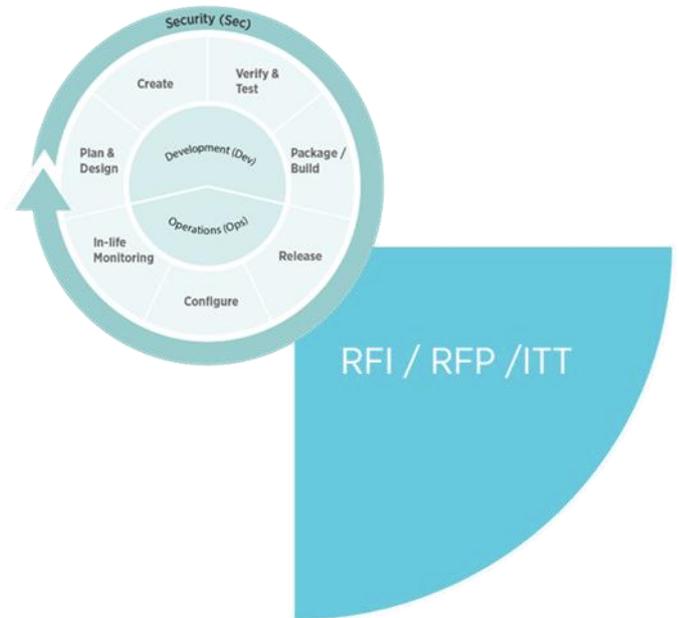


Figure 29 The component lifecycle overlapping with the systems RFI / RFP / ITT phase

⁸⁴ Managing Security Risks Inherent in the Use of Third-party Components at https://safecode.org/wp-content/uploads/2017/05/SAFECode_TPC_Whitepaper.pdf

Deploy / Commission

At this stage, the intention is to ensure the security features designed in earlier have actually been implemented in practice.

Holistic Penetration Testing can be used to gain a wider test aperture on the totality of the system rather than just the new component being installed. Active penetration testing can be used as a precision tool to validate the design implementation. Additionally, a threat assessment can be used to inform approval to go live after vulnerabilities have been fixed, baseline regression testing completed, defect management plan and tracking established.

For components, integration can be considered through additional API testing, validating Plugfest results, undertaking Static Code Analysis to test any open source code. A SCA can be undertaken to audit the SBOM. Dynamic Application Security Testing can be used on the deployed code as well as ensuring binary equivalence of the deployed code. Software image files can be scanned for known vulnerabilities in repositories before they get deployed into production

Specific security use cases developed during the earlier 'shift-left' can now be tested as well as fuzzing tests and error handling tests.

Systems Hardening

Security hardening approaches should be deployed against a range of components including Hardware, OS, VM, Container and Cloud / IaaS.

It is also a great time to ensure the principle of least privilege is implemented and to ensure the removal of unused management protocols, IP addresses and internet services. Unused applications and accounts should be disabled to limit attack opportunities. If relevant, this stage can be an opportunity to validate 'secure by default' settings. More generally checks should be made to ensure settings, credentials and passwords are configured and protected correctly, i.e. avoiding default or simple passwords. Limitation of root access is an important factor as is ensuring the removal of any administrator accounts created during installation and test activity.

The asset inventory can also be updated at this stage to ensure it accurately represents the deployed infrastructure.

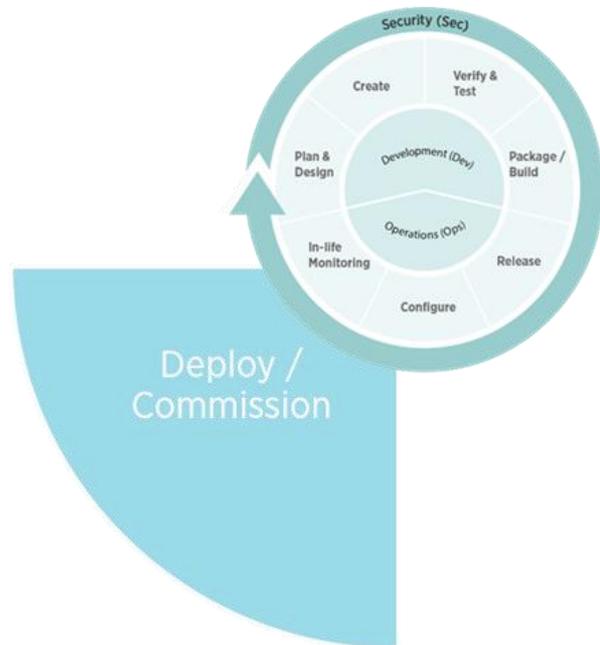


Figure 30 The component lifecycle overlapping with the systems Deploy / Commission phase

Operate / Maintain

This stage is vital to the practical delivery and maintenance of security. Holistic (i.e. not just the new platform or component) Penetration tests can be repeated in-life and repetitively especially following change activities. As part of this activity it is possible to validate security capabilities to detect attack approaches such as password spraying, unapproved privilege escalation and lateral movement.

It is important to maintain an accurate asset inventory (e.g. Compute, IP addresses, Routers) so that they can be maintained and protected. Undertaking activities such as regular passive scanning of infrastructure can assist with maintaining an accurate inventory and that equipment management interfaces are suitably protected.

Similarly, Configuration management is important to maintain a 'known state' of the network set-up.

Up to date patching is a vital aspect, especially where rapid CI/CD cycles may mean faster and more incremental code deployments. Vulnerability management (including use and maintenance of SBOM) is a key activity here whereby CVEs with high scores must be rapidly assessed against the deployed code base (SBOM) and patches implemented within Service Level Agreements to ensure ongoing protection. Risk assessment can also be used to assess the response to a given vulnerability being made known, especially if it is not possible to deploy a bug fix (perhaps due to an ongoing incident or service protection period). Consideration⁸⁵ can be given to:

- MITIGATE1: Patch/Update the Version.
- MITIGATE2: Replace with an Equivalent.
- MITIGATE3: Branch Code Internally.
- MITIGATE4: Contribute to Community/Vendor.
- MITIGATE5: Mitigate Through Code.
- MITIGATE6: Accept Risk.

The DevSecOps approach will mean security consideration is being given to the upgrades and enhancements that will be required in the next iteration of component or system update.

Privileged Access Management is a key security component in protecting the powerful access to mobile network systems. For example, it is important to maintain a limit on 'root' access / Administrator Access (with a formal approval mechanism, limited duration accounts, deletion of old accounts and limiting the number of accounts) as is the implementation of secure Privilege Access Workstations. Once approved, these PAM accounts must have suitable account access authentication and Multi-Factor Authentication. Personnel Security is another important human aspect of securing high privilege access accounts. Additional vetting can be considered for Administrator roles including encouraging limiting the operational details published by Administrators on social media.

Processes and containers in the runtime environment can be scanned, vulnerable processes blocked and prevent vulnerable containers must be prevented from being exposed. Additionally, configuration files can be scanned for any unauthorised changes

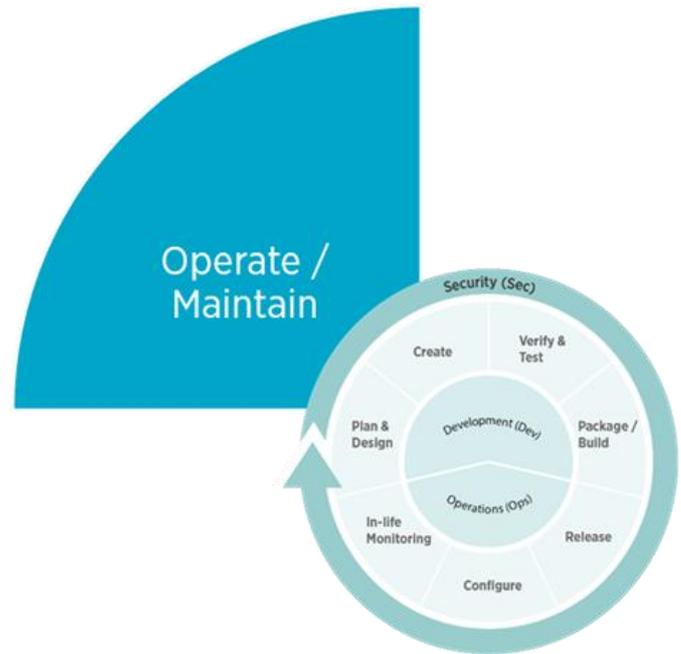


Figure 31 The component lifecycle overlapping with the systems Operate / Maintain phase

⁸⁵ Managing Security Risks Inherent in the Use of Third-party Components at https://safecode.org/wp-content/uploads/2017/05/SAFECode_TPC_Whitepaper.pdf



As networks exist in an inter-connected state, it is in operator interests to contribute to the wider security environment and knowledge base. As such, active participation in national and international security activities are beneficial. Within GSMA, this can be focused on sharing intelligence through T-ISAC, contributing to SAS and GSMA Security projects and Fraud and Security groups.

Conclusions

This whitepaper has drawn together a range of security approaches from operational IT and telecoms network perspectives. The approaches presented in this whitepaper are directly applicable to RAN considerations but are relevant to other network areas such as transport, software defined networking, core network and Internet of Things implementations.

The precise functionality of a given component is often specified in industry or international standards bodies; the focus for this whitepaper has been on *how* systems are built especially in more open disaggregated networks.

Open source software considerations have been considered in a range of deployment scenarios including application software that may contain open source library components and virtualisation layer code that may have significant open source software code origins.

One of the key aspects identified has been that of 'Whole system thinking' whereby consideration is given not solely to the newest network components but also to how they might successfully integrate securely with the existing infrastructure.

Component and system lifecycles have been used to explore the temporal aspects for the application of security necessary to design and operate a secure network.

The development of new capabilities can build on the foundations of previous work and experience. As networks trial and test more disaggregated components that skill and experience base is built stronger. There is more work to create the best mix of security considerations that will blend proven existing security practices with new approaches to match the new technology approaches.

In order to align with new technology approaches for OSS, there are many opportunities for action to create the best set of security considerations that incorporate proven existing security practices combined with new approaches. Specifically:

- Where vendor software includes open source components directly within code or is included in a full stack supply, encourage vendors to update/patch upstream components quickly or enable operators to act directly.
- To incorporate a Software Bill Of Materials to ensure full visibility of the deployed code in use.
- Exploit the strengths of open source transparency through code inspection, Source Code Analysis (particularly to generate and validate an SBOM), dynamic application security testing and encouraging use of coding standards through both vendor-Software Development Life Cycles and Core Infrastructure Initiative
- Where infrastructure virtualisation is delivered through a software package that is open source code-derived, use scanning tools to identify obsolete and vulnerable products and encourage a supply arrangement to enforce the ability to update out of date components within a stack.
- For infrastructure virtualisation, consider proving and re-using deployments with established industry benchmarks and common security-proven builds that have been extensively defined, tested and maintained. The Cloud Infrastructure Telecom Taskforce has undertaken work in this area.
- Incorporate proven security methods that deliver 'Bottom to top' security to preserve the root of trust for the solution as a whole. Current equipment is often supplied from a single vendor, open networking is changing this and may mean there are different vendors involved in each layer
- The O-RAN Alliance Security Group is defining security requirements to align to the specifications and interfaces. GSMA are keen to see and assist the O-RAN Security Group to drive the maturity of security specifications that will build confidence for scale deployments. These are important security considerations that require comprehensive design, feasibility and testing approaches that build maturity through practical experience.
- Consider the total operating environment into which open source code is deployed such that holistic security outcomes are considered across both new and existing infrastructures.
- Utilise a lifecycle approach such that security is designed-in, comprehensively tested in detail and in-context, deployed securely and then operated to maintain this security in-life.

Appendix A - Open Source Licensing

Excerpt from <https://www.compact.nl/articles/the-risks-of-open-source-software-for-corporate-use/>

Table 2 enumerates the properties of the six most common open-source licenses. As discussed earlier, GNU licenses are the most restrictive as they impose to release the associated source code as well as the list of changes, but you're allowed to use the GNU trademark in the name of your project. The LGPLv3 license is a bit less restrictive as you can keep your code closed source if you only use the open-source code as a library for your project. The Mozilla license is similar to the LGPL, except you don't have to state the changes made to the codebase and cannot use the Mozilla trademark. As a result, if you do not intend to commit to the open source community, the best strategy is to target projects licensed under the Apache or MIT license as you don't have an obligation to disclose the source. Note that the MIT license is a bit more permissive than the Apache License 2.0. It is usually considered as a best practice to go for a less restricted license when unsure, as, the risk of license violation is then decreased. Some mitigations to this risk are proposed in the last section of this publication.

Licenses name	Permissions	Conditions	Limitations
GNU Affero General Public License v3.0 (GNU AGPLv3)	Commercial use Distribution Modification Patent use Private use	Disclose source License and copyright notice Network use is distribution Same license State changes	Liability Warranty
GNU General Public License v3.0 (GNU GPLv3)	Commercial use Distribution Modification Patent use Private use	Disclose source License and copyright notice Same license State changes	Liability Warranty
GNU Lesser General Public License v3.0 (GNU LGPLv3)	Commercial use Distribution Modification Patent use Private use	Disclose source License and copyright notice Same license (library) State changes	Liability Warranty
Mozilla Public License 2.0	Commercial use Distribution Modification Patent use Private use	Disclose source License and copyright notice Same license (file)	Liability Trademark use Warranty
Apache License 2.0	Commercial use Distribution Modification Patent use Private use	License and copyright notice State changes	Liability Trademark use Warranty
MIT License	Commercial use Distribution Modification Private use	License and copyright notice	Liability Warranty

<https://www.compact.nl/wordpress/wp-content/uploads/2020/04/C-2020-1-Bouix-t2-groot.png>



GSMA HEAD OFFICE

Floor 2
The Walbrook Building
25 Walbrook
London
EC4N 8AF
United Kingdom
www.gsma.com