Secure design principles

Guides for the design of cyber secure systems

Getting the most from the secure design principles

These principles are intended to help ensure that the networks and technologies which underpin modern life are designed and built securely.

The problem

To be useful, systems very often need to move, store and provide access to sensitive data. Unfortunately, this makes them prime targets for cyber attack. If these systems are successfully compromised, the fallout can be damaging, expensive and embarrassing.

However, the picture need not be a bleak one. Frequently, the very worst outcomes can be avoided if services are designed and operated with security as a core consideration.

With this in mind we have developed a set of principles to guide you in the creation of systems which are resilient to attack, but also easier to manage and update.

System design

Throughout this guidance, we use the term system, by which we mean ‘a collection of digital components that are connected using communication
technologies to perform a business function.' A good example of the sort of system we are describing here is the UK’s online passport application service, but it could refer to many other digitally-enabled business functions.

We will also use the term **cyber-physical system**, by which we mean ‘a system that measures or controls the physical world to achieve a particular goal.’ A good example is a modern car, in which complex logic measures the physical environment in order to control the movement of the vehicle.

The principles have been conceived to be applicable to both digital systems and cyber-physical systems.

---

**Audience**

This guidance is aimed at people who design systems. The principles are most useful in the design and build phases of a project, although they can also be used to review existing systems.

---

**How this guidance is structured**

Applying the principles will require some customisation to suit your particular situation. For example, the exact requirements of an online information service will be different to the remote management of a power station. However, the principles will guide your considerations in either case.

*The Cyber Security Principles* offer the most generally applicable advice. *The Virtualisation Design Principles* apply to the more specific case of systems which rely on virtualisation technologies.

We have divided each set of principles into five categories, loosely aligned with stages at which an attack can be mitigated:
- **Establish the context**
  Determine *all* the elements which compose your system, so your defensive measures will have no blind spots.

- **Making compromise difficult**
  An attacker can only target the parts of a system they can reach. Make your system as difficult to penetrate as possible.

- **Making disruption difficult**
  Design a system that is resilient to denial of service attacks and usage spikes.

- **Making compromise detection easier**
  Design your system so you can spot suspicious activity as it happens and take necessary action.

- **Reducing the impact of compromise**
  If an attacker succeeds in gaining a foothold, they will then move to exploit your system. Make this as difficult as possible.

**PUBLISHED**
21 May 2019

**REVIEWED**
21 May 2019

**VERSION**
1.0

**WRITTEN FOR**

Small & medium sized organisations

Public sector

Cyber security professionals

Large organisations
1. Establish the context before designing a system

Before you can create a secure system design, you need to have a good understanding of the fundamentals and take action to address any identified shortcomings.

1.1 Understand what the system is for, what is needed to operate it, and which risks are acceptable

It is essential to have a clear understanding of the purpose of any system. You need to know which data, connections, people, and other systems will be required for it to operate.

You should determine what impacts you are not willing to accept.

Examples might include:

- unauthorised access to view, modify or destroy data, or the system being unavailable to users for a period of time
- a significant fraud being conducted
- the safety protections of an industrial control system being undermined

Explore examples from other organisations where things have gone wrong, and play out what this would mean in your own context. Feed this into your risk analysis.

To inform your design decisions, you will also need to know which risks are acceptable. Document the risks you are willing to take and ensure that all people involved in designing the system are familiar with them, so they can make well-informed decisions.

1.2 Understand the threat model for your system
Consider employing threat modelling techniques such as attack trees to help you discover the ways in which an attacker could realise their goals. Your design should also consider what level of capability an attacker would need to be successful, and whether your aim is to defend, detect, or recover, along with any useful time bounded goals.

For example, a common level is to be able to defend against publicly known tools and techniques, detect attempted attacks using them, be able to recover within a given time frame from a worst case scenario successful attack such as loss of all data.

Once you understand these items you can map security controls to those attacks to gain confidence that you should be appropriately resilient.

1.3 Understand the role of suppliers in establishing and maintaining system security

The suppliers you choose to build and operate your system play a vital role in helping to keep it secure. It’s important that all parties understand their responsibilities.

Contracts with suppliers should make your security requirements clear, but being over-prescriptive can lead to adversarial behaviour. It’s better to build a shared risk proposition with suppliers, so they are invested in doing the right thing, rather than just fulfilling a contractual obligation.

See also
The NCSC’s Supply chain security guidance is designed to help you establish an effective regime of control and oversight, for your suppliers.

1.4 Understand the system 'end-to-end'
You should understand the critical information and/or communication flows that your system relies on for operation. Take account of every possible point at which data could be stored, manipulated or rendered.

The following areas are often overlooked:

1. **Devices used to access data**
   If data is displayed or processed on a device it should be assumed that the data is present on that device. Any data a user can access could be available to malware on the user’s device.

2. **Third-party services**
   Outsourced support suppliers, hosting providers and the management environments of system integrators are often put out of scope when considering the security of a system. Avoid making this mistake, since an attacker with access to one of these environments could attempt to gain access to your system.

3. **Network-security devices**
   Web-browsing proxies and other network-monitoring devices typically used in corporate environments may decrypt traffic between your system and its users. These devices may have access to large volumes of sensitive data and could be exploited by attackers.

4. **Copies of your data**
   Consider copies of data stored in audit logs and monitoring tools, or copies that have been exported into business intelligence or management information tools.

5. **Communications over insecure networks**
   If your system communicates over channels which are not physically secure, your design will need to incorporate technical controls to provide an appropriate level of confidentiality and integrity.

6. **Appropriate security for every iteration of your system**
   During the design process, you may create separate iterations of your system for different purposes, such as development, testing and production. The impact of compromise associated with these environments is likely to vary at different phases of the system life-cycle and should be carefully considered. This is particularly relevant for complex industrial projects and cyber-physical systems, where a large number of different components are integrated to form a complete system.

1.5 Be clear about how you govern security risks
Good governance implies effective control over your systems and operations security, not blind adherence to pre-determined processes.

Where design decisions require you to balance security, usability and cost, it’s important to talk about any trade-offs in terms of their business impact, rather than relying on technical language.

Consider the cost of not doing something just as much as the cost of doing it. Keep in mind that these costs could include fines under GDPR or NIS legislation, as well as business cost and reputational damage.

Our guidance on governance and risk management may help you decide on suitable governance arrangements for your system.

1.6 Ensure there is no ambiguity about responsibilities

Everyone involved in designing and operating a system should be suitably qualified or experienced, know what their role is, and know what decisions they are empowered to make.

Ensure that the right people are empowered to protect critical systems and accept that this could mean giving relatively junior people the ability to affect business operations. This could extend to deliberately reducing functionality or service levels in response to external events - without reference to senior management.

You should adopt a continual development approach to skills and training. This will ensure that gaps in your capabilities are identified, logged and mitigated. Where appropriate expertise is not available, the associated risk should be escalated and managed as part of your organisation’s risk management system.
WRITTEN FOR
Self employed & sole traders
Large organisations
Public sector
Cyber security professionals
Small & medium sized organisations
2. Make compromise difficult

Designing with security in mind means applying concepts and using techniques which make it harder for attackers to compromise your data or systems.

2.1 External input can't be trusted. Transform, validate, or render it safely

Any data from an external or less trusted source could have been crafted to attack your system.

Well structured data can be validated to ensure it conforms to the expected format. If this isn’t possible, the only way to gain confidence in its trustworthiness is to transform it.

If you cannot transform the data, you’ll need to take care when you render it, ideally doing so in an environment you don’t mind being compromised. If you’re importing software or binaries, validate cryptographic signatures to ensure the software really was built by a vendor you trust.

Transformation, validation and safe rendering

1  Transformation
   We don’t believe it is possible to reliably check for malicious code in complex file formats like PDFs or word-processing documents.
   In these cases, it’s best to transform the file or content into another format. This should have the effect of ‘neutering’ any malicious content, before the file is passed on to its destination.

2  Validation
   Checking that the structure and content of data or files are as expected, to ensure that they will not inject malicious code into the destination system, or have unintended effects.
   This is a technique that can only be relied upon for relatively simple file formats.

3  Safe rendering
   Sometimes, validation and transformation may not be possible, or won’t give you sufficient confidence that the content is safe.
In this case, rendering the content in a disposable environment such as a non-persistent virtual machine or remote desktop may be your best option.

---

**See also**

For more advice on transformation and validation, along with our recommended architectural pattern for data import, see our guidance on [Safely Importing Data](https://www.ncsc.gov.uk/collection/cyber-security-design-principles/making-compromise-difficult).

---

### 2.2 Reduce attack surface

You should only expose the interfaces necessary for the operation of your system. If would-be attackers can’t reach an interface, they can’t attack it.

Remove all default configurations and features that aren’t required, such as user accounts, passwords, scripts and demo capabilities.

Don’t expose software unnecessarily. When building upon common tools or software, disable any components and libraries you don’t need.

Enforce read-only, limited data views, by pushing a needed subset of data out from essential business systems to staging areas, such as a DMZ. Your users can view them there, safely. If an attacker compromises the staging area they should be unable to interact with your business systems.

---

### 2.3 Gain confidence in crucial security controls

Understand which components in your system are providing the most important security controls, and gain confidence that they are performing as expected.

To gain the appropriate levels of confidence, you should ensure that:
• vendors or service providers are trustworthy, and competent at cyber security
• individual products and services are well designed, engineered and operated
• your deployment of any given product or service has been well configured
• your deployment of any given product or service remains well configured, throughout its life

Your approach will depend on what it is you’re trying to gain confidence in.

For suppliers, you could look for formal certifications and audit reports. For a specific service or product, you could review its track record of responding to discovered vulnerabilities. And for specific deployments, penetration testing will give you detailed feedback.

You should focus your efforts on the controls that are most important for the security of your system.

2.4 Protect management and operations environments from targeted attacks

Attackers often target privileged users (administrators, engineers, etc) with spear phishing emails or watering hole attacks.

To avoid an attacker obtaining the privileged accesses which these users require, you should design your system so that users cannot view email, or browse the web, from the same account or device that they use to perform their privileged actions. In this guidance we use the term 'administrator' to refer to any privileged user.

Ideally, you should design your system administration architecture to use a separate management infrastructure. As a minimum, this means using bastion hosts. However, you should be aware that a bastion host approach does not avoid malware on the user’s device being able to control an administrator’s session on a remote desktop.
This risk can only be mitigated by removing the opportunities for malware to gain access to the administrator’s device. This could be done, for example, by providing separate admin devices, or by ensuring that admin devices must access email and the web via remote desktop, or similar technology.

---

2.5 Prefer tried and tested approaches

Building something bespoke, when there are a variety of commodity options you could leverage, is not something you should do lightly. This is particularly important in software development and cryptography.

Popular software frameworks and libraries are often well maintained and tested, with a community of developers actively searching out and fixing vulnerabilities. By building upon these ready-made solutions you gain the advantage of this testing and scrutiny. However, you should still perform your own checks and assessments on any third party software.

When it comes to cryptography, designing new techniques is incredibly difficult to do well. Unless you’re an expert in the field of cryptography, you should never need to do something bespoke. Use existing algorithms and protocols, preferably those in common use by your chosen software stack.

---

2.6 All operations should be individually authorised and accounted for

Sensitive or privileged actions should only be permitted through an access control function which can verify that the user is who they say they are, and that they have permission for the access they are requesting.

The ability to attribute actions to individuals rather than to groups will be important when it comes to establishing accountability. It will also aid incident response.
If you have a compelling requirement to share credentials to enable continual operation of a system from a control room, physical and procedural controls could be used to achieve the same outcome. This might involve things like physical access control and CCTV.

2.7 Design for easy maintenance

A poorly maintained system is a vulnerable one. Make sure you are monitoring for security advisories and patches and have the ability to either mitigate all issues, or quickly determine which issues you must address.

Security vulnerabilities need to be fixed promptly, either through software patches, or by taking other mitigating actions. Frequent small updates are preferred over infrequent large ones as they have lower risk profiles. Increasing the frequency of deployments also creates confidence in your deployment mechanisms, and ensures that teams are well disciplined in rolling back changes.

Design your system so you don’t need to have outages which impact the business in order to apply updates.

2.8 Make it easy for administrators to manage access control

Having a unified view of users and permissions for a system, or systems, can help administrators maintain access control more easily.

Your design should support an identity lifecycle management process (e.g. for joiners and leavers, people changing roles, and ‘break glass’ credentials if needed).

Taking a single sign-on approach rather than duplicating similar identification and access control systems can simplify identity lifecycle management.
2.9 Make it easy for users to do the right thing

Security breaches often occur because users have developed workarounds for system inadequacies. Be sure to consider the potential for this and identify any methods users might resort to when circumventing security features.

Make the most secure approach the easiest one for users.
3. Make disruption difficult

When high-value or critical services rely on technology for delivery, it becomes essential that the technology is always available. In these cases the acceptable percentage of ‘down time’ can be effectively zero.

3.1 Ensure systems are resilient to both attack and failure

In order to cope with failure it is common practice to provide standby systems, alternative routes, and data backups. These perform well against random failure or mistakes, but often less well against malicious attack.

For example, if you have 10 identical load balanced servers and each has a 1 in 10 chance of random failure, the chances of them all failing at once are 1 in 10,000,000,000. However, if they all have the same vulnerability, it’s very little extra work for an attacker to make all 10 fail rather than just one.

A second example is backups. Having a copy of your important data is a good idea. However, if an attacker can delete or corrupt the backups easily then they will only be useful to recover from random failures and mistakes.

Finally, for cyber-physical systems, safety controls are implemented to reduce the risk of a hazardous outcome. This process should carefully consider the possibility that an external threat actor could alter the integrity or availability of the safety controls.

Safety system architectures should be designed to ensure that unacceptable consequences are prohibitively costly for an attacker to achieve. To obtain this outcome, safety controls should be independent in the event of both a system compromise or a mechanical failure.

3.2 Design for scalability
To handle exceptional peaks in demand, or future expansion, you may need to scale your service quickly.

Considering the potential for future demand early on should mean your systems are easier to scale later. They should also be better suited to scaling out under increased demand, or when under attack.

### 3.3 Identify bottlenecks, test for high load and denial of service conditions

Identify any system bottlenecks. For example, low capacity, legacy business technology, or an essential microservice which calls a third party service. Ensure that you have a plan in place to handle these bottlenecks during periods of high load or outage.

Add specific tests for abnormally high load, and for denial of service, to your overall testing strategy. For instance, you could simulate some denial of service attacks by purposefully terminating certain microservices or infrastructure elements in your pre-production environments.

There are also openly available tools, such as Netflix’s Chaos Monkey, which can help you test how your system will perform under high load or when components fail. It’s important to test how you respond to failure conditions, as well as understanding what those failure conditions could be.

See also
NCSC guidance on defending Denial of Service attacks.

### 3.4 Identify where availability depends on a third party and plan for the failure of that third party
Many organisations rely upon third party services, such as telecommunication links, hosting, authentication or system administration services.

Ensure you understand the availability characteristics of these third party provisions and the impact on your operations should they fail, especially at times of critical demand. Have a plan for minimising disruption if such an event occurs.
4. Make compromise detection easier

Even if you take all available precautions, there’s still a chance your system will be compromised by a new or unknown attack. To give yourself the best chance of spotting these attacks, you should be well positioned to detect compromise.

4.1 Collect all relevant security events and logs

Having the right data is essential. This is true whether you want to be well prepared for analysis in event of a breach, or if you want to detect potential and actual compromises in real-time.

Ensure you log enough to perform root cause analysis in event of a failure. Will your logs hold the data you need to work out whether a failure happened as a result of a breach? Both infrastructure and application level logs may be needed.

Ensuring log integrity

As well as collecting logs and capturing relevant events, you should ensure that the integrity of your logs would be maintained in event of a breach.

The attacker should not be able to cover their tracks.

4.2 Design simple communication flows between components

A well thought-out design, with clearly defined and tightly constrained communication between components, can simplify security analysis and enable you to identify when something is amiss. Components attempting to communicate in ways which are not part of your design can be a strong indication of compromise. Configure your monitoring tools to detect these indicators and automatically raise alerts.
4.3 Detect malware command and control communications

Watch for attempts by compromised components to contact their command and control infrastructure. This can be achieved by allow listing external domains, or addresses that are acceptable for data egress. Attempts to reach other domains should be prevented and reviewed.

4.4 Make monitoring independent of the system being monitored

This ensures that if the system being monitored is compromised, the attacker will have no visibility of whether the breach has been detected.

The same principle holds true for cyber-physical control systems (such as industrial control systems), where telemetry and control channels should be kept independent if it is critical to know how a system is behaving, irrespective of an attacker’s actions.

4.5 Make it difficult for attackers to detect security rules through external testing

If user activity causes one of your security rules to trigger, ensure you only give the minimum user feedback necessary.

This makes it more difficult for an attacker to understand your security logic when trying to map out your defences.
4.6 Understand 'normal' and detect the abnormal

Knowing how your systems normally operate means that unexpected behaviour can be recognised.

In addition to designing simple communication flows between components, it can be useful to monitor network load, storage I/O, compute performance, or transaction activity, to help you understand when your system is behaving abnormally.

Whilst this information is useful for optimising system operations, it could also provide potential indications that an attacker is attempting to compromise your systems, or that a breach has occurred. Ensure that anyone monitoring your systems knows what 'normal' should look like.
5. Reduce the impact of compromise

Design to naturally minimise the severity of any compromise.

5.1 Use a zoned or segmented network approach

Segmenting assets on a network provides the following benefits:

- It helps to contain the compromise to the segment that has been breached
- It enables you to better protect the assets that are most sensitive or valuable
- It supports the ability to limit or examine communication flows between segments. This means monitoring rules can be created which are able to assert with high confidence that a breach or misconfiguration has occurred

Decisions on how to segment a network will typically consider the protections different assets require, their need for interaction with other assets, and the extent to which their integrity is trusted.

5.2 Remove unnecessary functionality, especially where unauthorised use would be damaging

If functionality exists for *authorised* users then it can be abused by *unauthorised* users in event of a compromise.

Reduce the presence of unnecessary functionality and you reduce this risk. In doing so you’ll also cut the operational overhead of maintaining software or functionality you don’t need, simplifying your system and making monitoring easier.

Removing unnecessary functionality can take several forms, such as tuning the default configurations of the software you use, or removing debug or test functionality from production systems.
5.3 Beware of creating a ‘management bypass’

A common design flaw is to have weaker security controls and security architecture in management communications, than in the systems being managed. In such scenarios, compromising a single external-facing component can result in privileged access to systems or data via channels which were only intended for administrative use.

5.4 Make it easy to recover following a compromise

Design your system architecture so that, if you detect a compromise, you can quickly rebuild to a known clean state - once you’ve addressed the flaw which led to compromise.

Create a design that will enable you to both recover and maintain the records and data you might need to support an investigation. If you wait until post-compromise, you might find that you have to choose between recovering the system quickly or keeping the data you need for an investigation.

5.5 Design to support 'separation of duties'

Where the impact of attack, misuse or compromise would be significant, consider requiring the most privileged or potentially dangerous functions in the system to demand two or more individuals working together to perform them.

Example

Consider how you might prevent a single administrator (or their account) from exporting a copy of all data, or carrying out high-impact changes.
5.6 Anonymise data when it’s exported to reporting tools

Performance or reporting tools should be supplied with de-sensitised data.

Suppose you have an operations team that wants to create and display a performance dashboard for a financial payment system. To function properly, their dashboard does not need to work on raw transaction data, which may well be sensitive. Any personally identifying information can be stripped from the data before the analytic system processes it. This will reduce the number of places that a high impact breach could occur.

We recommend implementing controls to anonymise data as close to the source as possible. Rather than relying on reporting tools to anonymise data, you should maintain control of the process yourself.

5.7 Don’t allow arbitrary queries against your data

Don’t design functionality or deploy applications which enable arbitrary queries against your data. These applications undermine segmented system designs by providing an easier path to compromise data.

5.8 Avoid unnecessary caches of data

These data stores are likely to be less well protected than the primary data store, but can potentially yield high-value information to an attacker.

If a cache or temporary data store is required, then it should operate a data-fading policy that purges records as soon as possible after access has finished, ensuring that minimal data is stored in the cache and what is stored is protected appropriately.
Virtualisation security design principles

These principles focus on virtualisation technologies which may be used in the cloud, on servers deployed on-premise, or on end user devices.

The principles are a subset of the Cyber security design principles, extending them to help you think through the security considerations when designing systems that use virtualisation.

Virtualisation

Deploying and maintaining discrete systems using physical infrastructure, such as servers and network routers can be expensive, time consuming and inefficient. Virtualisation allows multiple systems to share physical resources, enabling many virtual systems to be created and deployed cheaply and quickly.

Virtualisation is a valuable tool for consolidating resources, supporting legacy systems, increasing flexibility and reducing costs. Although it is often used to build compute resources, it can be applied at many layers of a system, including networking and storage.

This technology is powerful, but it also introduces additional layers of complexity and, potentially, additional risks into your systems. It’s therefore important to ensure that security is considered throughout the design process of any system which relies on virtualisation.

These principles aim to help you design systems that take advantage of virtualisation, without also introducing unmanageable risks.

Terminology
There are many different types of virtualisation including compute, networking, storage, with more novel applications being continuously devised. We’ll focus on infrastructure virtualisation but these principles could be applied to any type of virtualisation.

Components of a virtual system

Throughout this guidance we refer to virtualised systems, virtual instances and virtualisation platforms.

The diagram and definitions below will help you grasp how these objects are related

**Virtual system** – A system built using multiple virtual instances running on the virtualisation platform.

**Virtual instance** – The logical virtual instance which runs on the virtualisation platform. This could be a full fledged virtual machine, virtual firewall or other networking device, or perhaps a virtual file system. Virtual instance are generally logical representations of physical resources, but may also be an additional layer of abstraction, used to bring multiple resources into a single logical view.
**Virtualisation platform** – The underlying provider of virtualisation. The platform allows multiple, segmented virtual instances to be run on it and manages the resources that are available to them. The virtualisation platform can include many components such as the hypervisor, orchestration and management functions.

**Hardware** – The physical hardware that the virtualisation platform runs on. Often hardware has specific features to enable efficient virtualisation so that the hardware can be shared between virtual instances at near native speed.

---

**Audience**

The primary audiences for this guidance are system architects, engineers and security experts designing and building systems using virtualisation.

The principles may also be useful for review and accreditation activities.

---

**Structure of virtualisation security principles**

These principles will guide you through some of the risks associated with virtualisation and provide guidance on mitigating those risks. You will gain the most from these principles if you use them to guide your security choices early in your project’s design process.

We’ve split the principles into five sections, each dealing with a specific aspect of system security:

1. **Establish the context**
   Before you start designing your virtualised system, you should first understand what business operations it will support, the risks it will face and the impact of its compromise.

2. **Make compromise difficult**
   Design with security in mind at each layer of the virtualised system.

3. **Make disruption difficult**
Virtualisation has a number of features which can be used to help avoid disruption and improve availability. However, if a system is not designed correctly, virtualisation can become a single point of failure and reduce resilience.

4. Make compromise detection easier
The design should include the ability to detect attacks and compromises. This will enable you to respond promptly to both attempted and successful compromises, introducing additional security controls where necessary as part of the ongoing lifecycle of your system.

5. Reduce the impact of compromise
Features such as replication, snapshots, and high availability can be used to speed system recovery, giving virtual systems an edge over traditional infrastructure. When designing a system, you should take advantage of these strengths wherever possible.
1. Establish the context

Before you start designing your virtualised system, you should first understand what business operations it will support, the risks it will face and the impact of its compromise.

**The virtualisation security design principles**

**Note**
This principle builds on Establish the context, from the Cyber security design principles

1.1 Understand the risk profiles of all systems sharing a virtualisation platform

While virtualisation aims to provide separation between virtual instances, this protection is not perfect. There have been virtual machine escape vulnerabilities which allow one virtual instance to interact with another or the underlying virtualisation platform. There have also been examples of vulnerabilities can cause a denial of service on the virtualisation platform, thus impacting all the virtualised instances on the platform.

Given this, it may not be appropriate to host virtual instances with widely differing risk profiles and impacts of compromise on the same virtualisation platform. For example, a virtualised system running a cyber-physical system such as an industrial control process alongside a system connected directly to the internet hosting a web server.

If there is a requirement to share a virtualisation platform in this manner, additional controls will be required to manage the additional risks. An example would be disabling the virtual instance to virtualisation platform communication channels to reduce the attack surface, but this will also disable functionality.
1.2 Where virtualisation is not being used as a security barrier, extra security controls aren't required

Once you have understood how you will use virtualisation, and the systems that share the virtualisation platform, you can determine the risk and impact of compromise.

If virtualisation is being used to host systems of the same risk profile and impact of compromise, you will be able to use the full and rich functionality of the virtualisation platform. This is the case when virtualisation is being used to consolidate resources or improve ease of administration.

1.3 Understand the role of suppliers in establishing and maintaining system security

This is especially key when building services on top of virtualisation platforms provided by external parties, such as cloud services.

Virtualisation may be present in multiple components in different forms throughout their infrastructure, from compute through to storage and networking. You may not be aware that these components are virtualised, or that you are sharing resources with multiple users.

One example of a resource sharing scenario is Software Defined Networking, often deployed in data centres. To the end user, it appears they have a private network. In fact, they are sharing it with multiple tenants of the same data centre.

Work with your supplier to understand the technology they use and determine if it’s appropriate for your system. Use the Supply chain security guidance to inform your discussion and help with your assessment.
Cloud services
Cloud services heavily rely on virtualisation to help them deliver their services. The Cloud Security Principles can help you build confidence in their security posture.

1.4 Virtualisation doesn't mitigate vulnerabilities in legacy and obsolete systems

Virtualisation can be useful for hosting legacy and obsolete systems. However, if these systems lack security features or have vulnerabilities, this will not be changed by virtualising them. These weaknesses could still be used by an attacker.

Virtualisation of obsolete systems can help to segregate them from the rest of your infrastructure, but a compromised virtual instance could be used to launch an attack against other virtual instances, or the underlying virtualisation platform.
Small & medium sized organisations
2. Make compromise difficult

Design with security in mind at each layer of the virtualised system.

Note
This principle builds on Making compromise difficult, from the Cyber security design principles

2.1 Protect virtualised systems as you would non-virtualised systems

A virtualisation platform doesn’t inherently protect the virtual instances running on it from compromise. The purpose of the platform is to allow multiple virtual components to run simultaneously and provide separation between them.

You should apply security controls to a virtualised system just as you would non-virtualised systems. This will help prevent compromise of the targeted virtual instance and onward compromise of other virtual instances on the same platform.

2.2 Continually update virtual instances and virtualisation platforms

Virtualising an unpatched system doesn’t reduce the risk of it being compromised. Therefore, updates should be applied to the virtual instance software when they become available.

Virtualisation can be used to streamline your update methodology. For example, a virtual reference system could be run to test updates before being applied to a
production system. Also, snapshots could be taken of a system before applying a patch, providing you with the ability to roll back changes if needed.

The design of your system should ensure that each component can be easily updated, use virtualisation features such as automatic failover to provide the flexibility to enable this.

Carefully consider how the virtualisation platform can be updated and maintained without impacting the virtual instances running on it. Updating the underlying platform shouldn’t impact operations or users.

### 2.3 Protect virtualisation platform management interfaces

An attacker with access to your administration console, or management infrastructure, can control the configuration of your virtualisation platform and all the virtual instances running on it.

Ensure that only a limited number of people, with a specific need, have access to the administration console. Make sure the administration console is sufficiently protected, using controls such as network segregation, strong password policy, multi-factor authentication and role-based access control.

Network accessible administration interfaces should only accept connections from authorised management infrastructure, which is segregated from normal system infrastructure.

### 2.4 Understand communication channels between virtual instances and the virtual platform

Many virtual platforms provide communication channels between virtual instances and to the virtual platform itself.
Rich functionality of the virtual platform is often enabled using these communication channels and so often it’s desirable to use them.

Ensure you understand the implications of using these communication paths and assess the risk versus benefit balance. These interfaces provide a route to the virtualisation platform, which an adversary could potentially exploit to move laterally between instances, or down to the virtualisation platform itself.

When using them configure as per the best practice guidance provided by the virtual platform documentation.
3. Make disruption difficult

Virtualisation has a number of features which can be used to help avoid disruption and improve availability. However, if a system is not designed correctly, virtualisation can become a single point of failure and reduce resilience.

**Note**

This principle builds on [Making disruption difficult](https://www.ncsc.gov.uk/collection/cyber-security-design-principles/virtualisation-security-design-principles/making-disruption-di...), from the Cyber security design principles

---

3.1 Design the virtualisation system to have redundancy

Systems that use virtualisation should be designed with redundancy in mind. A denial of service attack on a virtualisation platform can impact all the virtualised machines running on it. This could be achieved, for example, by compromising a poorly protected virtual machine and using that to launch an attack on the underlying platform.

Design for an acceptable level of disruption, using multiple layers of redundancy, if required. Measures which can help with this include multiple instances of hardware, network paths, storage arrays, splitting the system across different locations, online and offline backups.

---

3.2 Design to be flexible, scalable, and highly available

Many virtualisation platforms have features which enable them to be flexible, scalable, and highly available. Your design should take advantage of these features, helping you to produce stable and resilient systems.
Use of these features should be applied throughout the virtualisation stack. This will also enable ongoing maintenance, such as applying security patches to the underlying virtualisation platform and infrastructure, without impacting the virtualised systems running on them.

### 3.3 Ensure the management infrastructure is highly available

If a virtualisation platform is disrupted, the management infrastructure needs to be available for administrators to fix the issue. The management infrastructure should be separate from the operational infrastructure, so that an attack on operations cannot affect management activities.
4. Make compromise detection easier

The design should include the ability to detect attacks and compromises. This will enable you to respond promptly to both attempted and successful compromises, introducing additional security controls where necessary as part of the ongoing lifecycle of your system.

Note
This principle builds on Making compromise detection easier, from the Cyber security design principles

4.1 Know your virtual infrastructure

Design your system so that it’s easy to audit. You should be able to answer the question, "What is running, where, and which systems can it access?"

Uncontrolled use of virtualisation can quickly get out of hand, leading to a situation where you have little or no oversight of the systems running on your platform.

Your development workflow should include a process for setting up and documenting new virtualised systems. This is particularly important when deploying production systems which are critical to your organisation.

4.2 Use virtualisation features to improve audit and monitoring

In a physical environment, keeping track of equipment and "shadow IT" can be difficult. Using virtualisation can compound this problem by enabling quick and
easy creation of virtual systems, which may not be secured correctly, or simply forgotten about. This is particularly true in rapid release development cycles where the focus is on the next delivery, rather than cleaning up the last iteration.

Virtualisation can actually make auditing and monitoring easier. Since virtualisation platforms are managed from a central point, it’s possible to design the management infrastructure so that it can query virtualised system at each layer of the stack.

Use the flexible nature of virtualisation to insert monitoring capabilities into each layer of the virtualisation platform, reporting back to a central location.
5. Reduce the impact of compromise

Features such as replication, snapshots, and high availability can be used to speed system recovery, giving virtual systems an edge over traditional infrastructure. When designing a system, you should take advantage of these strengths wherever possible.

---

5.1 Design for rapid recovery

Virtualisation provides functionality which can help you recover from compromise quickly.

For example, your virtualisation platform can consist of multiple nodes to make a cluster, virtualised systems can then be replicated to multiple nodes, providing a quick recovery option should a node in your infrastructure become unavailable.

Having a backup of your system in a known good state could also be beneficial when recovering from a compromise. This can be achieved using snapshots or separate reference platforms. Ensure that these backups represent the latest iteration of your system and are continually patched.

---

5.2 Design your system to gracefully degrade service

Design systems which can gracefully degrade functionality, delivering a minimal set of services, in the event of an incident which prevents you from maintaining full functionality.
In a situation where the full feature set of a system can’t be delivered, perhaps due to cyber attack or even natural disaster, use the flexibility of virtualisation to respond quickly to any degradation in service.

For example, virtualisation can provide a means to quickly spin up systems in a good known state or perhaps networking can be quickly reconfigured to direct users to a failover system.
Examples

Case studies demonstrating the practical application of the design principles

**Design principles and Operational Technology**

A fictional case study exploring the application of our secure design principles.

**Security design principles and virtualisation**

Applying the virtualisation design principles to a practical worked example

PUBLISHED
21 May 2019

REVIEWED
21 May 2019

VERSION
1.0

WRITTEN FOR

Small & medium sized organisations
Public sector
Cyber security professionals
Large organisations
Design principles and Operational Technology

A fictional case study exploring the application of our secure design principles.

If you are responsible for the design or maintenance of an Operational Technology (OT) network, this study will help you to navigate the cyber security issues you will encounter as you design your cyber-physical system.

Last year we published our new cyber security design principles. Based on the NCSC’s experience of architectural review and handling incidents across UK Government and Critical National Infrastructure (CNI) systems, these principles are intended to help architects and designers produce secure and resilient systems.

The principles cater for both IT and OT systems but it can be difficult sometimes to see exactly how a principle should be applied in any given case. So in this example, we’re going to walk through the design process for an OT system, guided all the way by the secure system design principles.

Meet 'Admin Corp'

Let’s imagine we’re following a fictional organisation who are responsible for making decisions about the cyber security of a CNI processing plant.

Admin Corp runs a plant that produces Adminox, a highly volatile, refined form of administrative paperwork that is essential to every organisation in the country. It is created from volatile raw products using a continuous chemical process.

As an essential service, Admin Corp must comply with the EU NIS Directive. This means that Admin Corp’s network, information systems and technology needed for the production of Adminox must be protected from cyber attack.

Also, because Admin Corp are regulated for safety by the UK Health and Safety Executive, they must take steps to ensure the continued safety of the Adminox production process.
Admin Corp – system requirements

The process for producing Adminox involves a number of steps, with the final product stored under pressure in a tank. Clearly, no one would benefit from the unconstrained release of this Adminox, with the potential for additional red-tape clogging up local services for years.

Admin Corp’s system therefore has **two critical non-functional requirements**: 

- As a responsible and safety regulated company, they need to keep the local environment *safe* from release of Adminox.
- Maintaining the *availability* of the product for customers in order to continue as a profitable company.

---

Applying the principles

The design principles should be applicable whether there is an existing architecture which requires re-design or a completely green-field site. In Admin Corp’s case there is an existing architecture which they wish to improve.

Let’s walk through each of the NCSC design principles and see how they can be applied to achieve the non-functional requirements:

---

1. Establish the context before designing a system

*“Determine all the elements which compose your system, so your defensive measures will have no blind spots.”*

The most important first step here is to gain a complete, end-to-end understanding of system operation, which parts are critical and to identify the organisation’s approach to undesirable consequences.
Implementation of this principle is likely to require involvement from all parts of the business: information technology teams, cyber security engineers, process operators, process control specialists and functional safety engineers (amongst others!).

**Attack trees**

Admin Corp want to understand the threat model for the Adminox processing system, so will gather relevant experts to build attack trees. These will illustrate the path an attacker would need to take to achieve a specific impact on either safety or availability. Note that an impact may only become noticeable when, for example, a disabled safety system is called upon. These attack trees will be used to inform design decisions later on.

**Network zoning**

As a fundamental part of this design, Admin Corp create a series of zones to group systems from which similar impact would occur if an attacker was to compromise them.

Separate zones are built for Business, Process and Safety functions. The premise being that if an attacker gains access to the Business zone, then the worst impact they could have is disruption to the running of the company, communication with customers etc.

Penetrate the Process zone, where the Process Control System (PCS) lives and the attacker can disrupt operations, causing a product availability issue.

Finally, if the attacker gains access to the Safety zone, then they may be able to adjust safety parameters to cause a destructive physical effect.

In our example, Admin Corp have a fairly simple zone configuration, but for larger systems, network zone design decisions are likely to be influenced by product design, geography and physical protection of the infrastructure (amongst other factors).

The **Purdue model** may act as a helpful reference model for this part of the design process.

**Critical Zone Boundaries**
Having created their attack tree and network zones, Admin Corp then walk through them, identifying the key points in the architecture where it might be possible to detect or prevent an attacker from achieving their goals – focusing in particular on the zone boundaries.

At each of these points, later in the design process, Admin Corp will reduce the attack surface by deploying countermeasures. In this way, even if an attacker reaches that part of the system they are going to be highly constrained and likely detected in their attempts.

As an example, think back to the 2015 attacks against the energy networks in Ukraine. Here, according to open-source reports, the attacker’s actions weren’t visible to the operator until the point at which they were moving the mouse around on the screen of the computer that ran the control system.

Ideally, the attacker’s actions should have been spotted or prevented before this point. Identifying and protecting key network choke-points will help with this.

Supply chain security
Also at this point, although beyond the scope of this blog, it’s important for Admin Corp to consider supply chain security. Specifically, Admin Corp will be considering how to gain assurance that new process logic, configurations, software, patches and other changes to the system do not impact on the key non-functional requirements.

Network design and documentation
Having established some of the fundamental aspects of the system, Admin Corp will then generate, communicate and agree upon a simple diagram of the plant, that depicts an end-to-end understanding of the physical process and logical network design. This will include the Business, Process and Safety zones and the boundaries between them.

Clearly documented on this diagram are the consequences the organisation has decided they are not willing to accept and the mitigations which have been deployed to prevent those consequences occurring.

Also depicted are the roles responsible for maintaining the operation of the system, both from a process perspective and a cyber security perspective.
(include the role that suppliers play). Later on, Admin Corp will use this to ensure that suitably qualified and experienced people are assigned to relevant roles and ensure that there is no ambiguity about responsibilities.

The diagram will enable good decision making by ensuring critical knowledge is widely understood. However, of equal importance is that it will also help develop a shared risk proposition with suppliers who may be involved in the maintenance of the plant or needing to conduct incident response in the event of a cyber attack.

Some organisations may choose to use existing risk models such as Bow-tie to illustrate the mapping of cyber security controls to undesirable impacts and consequences.

A simple network diagram
The Adminox system diagram could look like this:

Having got to this point, Admin Corp now use the NCSC guidance on governance to set-out clearly how security risks to the Adminox production process are...
2. Make compromise difficult

“An attacker can only target the parts of a system they can reach. Make your system as difficult to penetrate as possible.”

Having understood the context and armed with the diagram, Admin Corp will start making compromise difficult for the attacker.

Considering the original requirements, each of the network zones will require incrementally higher levels of confidence that the boundaries separating them from the layer below will stop an attacker.

Admin Corp would like to have greatest "trust" that the Safety zone has not been compromised, followed by the Process zone and finally the Business zone.

The key to making compromise of these boundaries difficult is to implement the following practices:

- **Do not trust external input**
  
  **Browsing the Internet from the control room:** Any input data coming from a lower trust zone could result in the exploitation of the system processing it. This means it’s important to consider carefully the types of data you’re bringing into and processing in the higher trust zone.

  In the Admin Corp example, the operators with computers in the Process zone require access to the Internet and email in order that they can manage the process effectively. However, giving these system operator terminals in the Process zone direct access to content downloaded from the Internet would be an example of a "browse-up" anti-pattern, which could result in compromise of the Process zone.

  To prevent this, Admin Corp will develop a secure browsing system based on a Virtual Desktop Infrastructure (VDI) solution. This system effectively creates a virtualised window up into the Business zone which is created and destroyed as
users require it. If malicious content is sent to the VDI system, it will be contained and eradicated from there before it is possible for the attacker to affect the process. This is an example of a "browse-down" pattern, where riskier activities are isolated using a separate processing context.

Admin Corp also realise that their process and safety engineers have the same requirement for their engineering laptops (to browse the Internet and access email). To mitigate the risk of compromising the Process zone, they extend use of the VDI environment to these users too. Through policy and technical controls, the engineering laptops are only permitted to access the Process and Safety zones (via a dedicated Process zone VPN gateway discussed in more detail later).

Enforce one way flow

*Visibility of the safety system:* For regulatory reasons, Admin Corp want to have visibility of the safety system so they can monitor plant safety in real-time.

Admin Corp’s executive board have decided that – due to the potential consequences – any electronic connectivity between the Safety and Process zones must only move data from the Safety zone from the Safety system (a one-way flow).

Admin Corp will use hardware known as a data diode to enforce one-way flow. This will prevent attackers from conducting online interactions with the safety systems and stop other malware in the Process zone impacting on the Safety zone.

Reduce the attack surface

Considering each of the boundary systems between the different zones of the network in turn, Admin Corp will look at the attack surface presented to an attacker operating in the lower ‘trust’ zone and reduce it by:

- Switching off unnecessary system and network services
- Implementing firewall rules which deny everything except for agreed critical network services
- Prioritise pushing sensor data from the higher trust zone to the lower trust zone and conducting complex operations such as data analytics in the low
trust zone. They will prefer tried and tested approaches to ensure simple and well understood mechanisms are put in place to push data out of the high trust zone

- Avoiding exposure of services in high trust zones to access from the low trust zone. This is important because, exploitation of these cross-border services could lead to the attacker gaining access to the high-trust zone. However, in some cases, cross-border access will be unavoidable. For these cases, Admin Corp develop and test strong authentication mechanism to control access to any services exposed across the boundaries. For some use-cases it may also be appropriate to consider implementing the NCSC’s pattern for safely importing data.

- Hardening all boundary hosts against attack by using modern operating systems and configuring contemporary mitigation measures to make it much harder for lateral movement to be successful.

Gain confidence in crucial security controls
The Admin Corp operations team are unhappy with the idea of a red-team test against their live systems. They worry about the potential this has for impacting operations.

This is a reasonable position to take, so they scope a red-team test very carefully, with the objective of penetrating the non-critical boundary systems within the Process zone (such as a historian or data-transfer system).

If a simulated attacker can successfully reach those systems without being stopped or detected, then it will be necessary to do more work to harden the attack surface.

Additionally, given the potential for safety consequences, Admin Corp create a lab environment in which the core systems and boundary between the Process and Safety zones are replicated. Lab testing provides a way to gain confidence that their data diode is operating effectively. This replicated environment allows effective red-teaming to be conducted safely.

In a further effort to obtain assurance without impacting production, Admin Corp also run a table top red team activity (paper-only) and architectural review with skilled security architects.
3. Make disruption difficult

“Design a system that is resilient to denial of service attacks and usage spikes”

To ensure the resilience of an OT system with high availability requirements, you will need to consider the design from a number of different perspectives. Design decisions will need to take into account physical, environmental, geographical and technical factors.

Overall system resilience can be achieved in many ways. For instance, by having multiple paths to a single destination, deploying redundant systems, or using geographically separated distribution networks.

These design decisions can lead to a good level of resilience - one that will cope with system failures or environmental incidents such as weather events. However, in some cases, design decisions can have an adverse impact on cyber security resilience. For instance, if multiple redundant servers are deployed, running the same version of vulnerable software, this would not be resilient from a cyber security perspective. An attacker would only require a single vulnerability to disrupt production. So, an attempt to reduce the likelihood of system failure could result in poor cyber security resilience.

The designers of the Admin Corp process are therefore mindful of the potential for a single vulnerability to impact their system. Through their work to develop an attack tree, they identify that the Process Control System (PCS) has a number of system components that are deployed in a redundant configuration, intended to maintain availability of the process. For compatibility reasons, they are unable to change the system design to include a different manufacturer’s components. Instead, they work with their supplier to ensure that robust cyber security controls are placed around the components, thereby making disruption difficult.

4. Making compromise detection easier
“Design your system so you can spot suspicious activity as it happens and take necessary action”

Having set out the system context and defined network zones, Admin Corp now needs the ability to detect when an attacker is operating in their OT environment.

Collecting logs
Admin Corp will be able to detect intrusions by collecting all relevant security events and logs. This is particularly important for the systems that sit on the edge of zones and communicate with lower trust systems.

Not all sensors, actuators and field equipment produce security events or logs, so it’s important to focus on the systems in the path of the attacker which do produce events and logs.

Previous attack tree modelling work will help Admin Corp determine where best to focus their efforts, but they also choose to develop even more detailed attack hypotheses, using expert advice find the most valuable data collection points in the architecture.

Admin Corp decide to use the NCSC’s Logging Made Easy for hosts in their Business Zone, while they establish a business case for investment in a more expansive Security Operations Centre.

Detecting malware
A key objective in collecting logs is to detect malware command and control communications.

A malware infection may be intentional or unintentional. Whichever the cause, early detection of the malware command and control activity is likely to reduce the potential impact. This is important because, even if the malware is unable to communicate with its controlling system it could still cause damage. For example, data deletion malware does not need to phone home in order to do damage.

Collection and analysis of domain name requests and attempted connections to Internet systems from OT environments are important steps towards detecting malicious activity.
Keep the attacker in the dark
When monitoring for command and control activity, it’s important to ensure that an attacker does not realise that they have been detected— or they may change their approach and become harder to detect.

This is achieved by ensuring that monitoring is independent of the system being monitored. Or in other words, use optical network taps and security controls to prevent an attacker gaining access to the monitoring environment. This approach will also make it difficult for attackers to detect security rules through external testing and as a consequence reduce the likelihood that the attacker will achieve their goals.

Simple communication
Detection of an attacker’s activities is easier if there are simple communication flows between components.

In Admin Corp’s design, they deploy a single, multi-factor authenticated, VPN gateway to provide access to the OT environment. Having gained access through this gateway, a bastion host constrains user activity to agreed policies. After this, network and host detection rules allow security analysts to understand what is normal and detect abnormal states or behaviours.

This gateway allows Admin Corp to provision and control access for remote engineers, integrators and suppliers, all of whom are using systems that Admin Corp trusts to the same degree as the Process Zone.

Access to the Process Control System (PCS)
Admin Corp’s Process Control System (PCS) is the brains of their production process. It is a vendor-provided solution that takes inputs from sensors across the plant and reflects that state on Human Machine Interfaces (HMIs). It then uses pre-configured logic to control the process and allows the operators to interact via the HMIs in order to respond to alarms, carry out maintenance and ensure the continued operation of the plant.

Admin Corp also uses a historian to record data from the process, which can be used report on process efficiency, enable maintenance and conduct other engineering operations.
In Admin Corp’s case there are a number of use-cases for access to the PCS

1. **Read-only PCS data for business analysts and regulatory compliance officers** (desktops connected to the Business zone network).

2. **Fully functional PCS screens for control room operators** (desktops connected to the Process zone network).

3. **Direct control of valves and actuators affecting the process of generating the Adminox product** (in the Process zone).

It may be tempting to consider the Human Machine Interface (HMI) component of the PCS an adequate boundary between different zones of the system. Indeed, in some cases, this may be the pattern endorsed by the manufacturer.

Given the requirement for read-only screens in the Business zone, it might be tempting to make holes in the firewall to allow hosts in the Business zone to connect to the PCS. By doing this though, the number of opportunities to detect the attacker before they can impact the availability of the PCS is reduced.

If Admin Corp were to expose the PCS through the firewall to the Business zone network, the attacker may need just a single exploit against the PCS to achieve their goals. This is something that, as a NIS regulated operator, Admin Corp will wish to take steps to prevent, in order to meet **Cyber Assessment Framework objective B4**.

**Detecting attacks against the PCS**

Instead of making holes in the firewall, the Adminox system designers and risk owners have agreed to create a mechanism by which read-only access to the PCS data is provided to the business network using known and trusted technologies. By deploying historian capabilities into the network, Admin Corp can collect the process data and make it available for analysis.

However, the primary historian will be deployed into the PCS environment, so for Admin Corp to maintain separation between the zones, they install a second replica of the historian in the Business zone. This replica system is sent data using securely configured database replication services from the PCS historian instance and then monitored carefully for compromise.
5. Reducing the impact of compromise

“If an attacker succeeds in gaining a foothold, they will then move to exploit your system. Make this as difficult as possible”

By using a zoned network approach and implementing effective monitoring, Admin Corp has made it much harder for an attacker who has gained access to the Business zone to traverse into the Process and Safety Zones.

However, they also need to take steps to reduce the impact of compromise in the case that an attacker is successful in compromising any one of the zones.

Only the essentials
The first step towards limiting the implications of compromise is to remove all unnecessary functionality from key boundary systems in the attacker’s path. This will reduce the likelihood that an attacker can further their activity in the network.

Administration
The design principles tell us to “beware of creating a ‘management bypass.” With this in mind, Admin Corp will also take a careful look at how they manage the systems which control authentication for each of the zones.

The NCSC regularly sees operators choosing to use the same Active Directory for both the business and process networks, or rely on network or systems management functions that are carried out from the business network.

The result of this is that, typically, an attacker simply has to compromise Active Directory or management systems in the business network in order to have full control of the process network. This is an example of the "Management bypass" anti-pattern.

As Admin Corp have decided that the Process zone is a higher trust zone than the Business zone, they decide to have an entirely separate Active Directory within the Process zone, which has no trust relationships in place with the Business zone.

They also decide that all network and system administration within the Process zone must be conducted from a hardened systems enclave within the Process
zone itself.

Allowing for a smooth recovery
Acknowledging that there is likely to be a compromise at some point, it is important to make it easy to recover following a compromise.

For Admin Corp, this means maintaining a set of offline "gold images" for systems which would allow administrators to quickly recover to a known-good state, in the case of a destructive attack.

It’s important to note that it’s not only an attacker with destructive intent who might do this, but far more commonly, ransomware, which could impact a system in this way.

For Admin Corp, the ladder logic and configuration of the process controllers is essential and so processes will be developed to regularly backup programs and configurations, storing them offline too.

Cost analysis
Having conducted their attack tree analysis and to reduce overall costs, Admin Corp choose to take a surgical approach, by backing up only the systems that would have an impact on operations.

Separation of duties
For some systems, it may also be appropriate to ensure that the design supports ‘separation of duties’.

In Admin Corp’s case, the process of making changes to any systems within the Safety Zone are subject to strict separation of duties, not just to meet any regulatory requirements for safety, but also to reduce the impact of a cyber attack.

Proposed changes to the network design, or configuration, will be peer-reviewed and authorised by the Admin Corp Change Board, which is composed of individuals with functional safety, operations and security expertise. Final change execution will be implemented by a separate team of suitably experienced engineers.
Protecting documentation
Finally, the NCSC knows from it’s experience of red-teaming CNI systems that an attacker who has access to up-to-date design documentation, P&ID and schematics is likely to achieve their objectives faster than one who doesn’t.

So, Admin Corp must carefully consider how to protect key design documents. This is particularly important for those documents that would be critical to an attacker achieving the consequences which Admin Corp are particularly intent on preventing, such as documentation of the safety system. It is possible to do this by avoiding unnecessary caches of documents and limiting access to certain key design documents on a need to know basis.

Next steps
Having achieved these steps, Admin Corp believe they have appropriate risk mitigation in place as part of their system design to defend against both deliberate acts of cyber intrusion as well as incidental malware infections that might have an impact on the production process.

Their next steps are to ensure they have iterative improvement processes in place to initiate improvement actions from any events which threaten to impact Admin Corp’s availability and safety requirements.

As an operator of Critical National Infrastructure, Admin Corp ensure they continue to manage the Adminox process safely and effectively by joining the NCSC’s CISP platform and maintaining a relationship with the NCSC Private Sector CNI Engagement Team.
VERSION
1.0

WRITTEN FOR
Public sector
Cyber security professionals
Large organisations
Security design principles and virtualisation

Applying the virtualisation design principles to a practical worked example

Let’s walk through an example to see how these virtualisation principles can be used when coming up with a (very) high-level design for a system using virtualisation.

One of the original requirements for the virtualisation principles came from the industrial process control community, who asked the question, “How can I use virtualisation safely in an industrial control environment?”

So, we’ll come up with a design for a fictitious manufacturing company, who have an industrial control system (ICS). They’re looking to consolidate infrastructure using virtualisation, without impacting on integrity or availability.

However, the top priority is that no aspect of system safety is compromised. This includes personnel in the workplace and any potential harm to the general public.

Establish the context

The first thing we need to do is establish the context of the system. This process is all about understand what business operations the system will support, the risks it will face and the impact of its compromise.

In this case, we’re dealing with a system used for industrial control. It currently has a traditional architecture, based on the Purdue model, with physically separate infrastructure between each level.

Figure 1. Traditional architecture
Consolidation and risk

The aspiration is to consolidate as much of the ICS as possible. However, based on risk analysis and threat modelling, it has been decided that consolidation of the enterprise IT with the ICS would be too risky.

Consolidation of operations together with the control systems is suitable because the risk profiles of all systems sharing the virtualisation platform will be a similar level.

There are availability concerns about putting all components of the system on to one virtualisation platform, so the design needs to ensure it doesn’t become a single point of failure.
There is also a legacy system used to control machinery that is no longer supported and has known vulnerabilities. Unfortunately, this is tightly integrated into the system and prohibitively costly to replace. The technical architects understand that virtualisation doesn’t mitigate vulnerabilities in legacy systems. To manage the risk, they plan on using micro-segmentation, a technique for dividing up the workload of a system into segments, based on its security needs, and applying necessary controls on a per-segment basis.

Consolidating with virtualisation

The system architects take the business context and requirements and start to develop an architecture which consolidates the layers of the ICS using virtualisation. They come up with the following high-level concept diagram.

Make compromise difficult
Each layer of the virtualised system should be designed with security in mind. This will help to make compromise more difficult.

In our example, a legacy component needs to be used, but access to it must be tightly controlled. To achieve this, the design makes use of virtual networking and micro-segmentation. This also helps to ensure that virtual instances don’t impact on one another, by helping to limit the lateral movement an attacker could make if they compromise the legacy component.

In our example we’ve used the virtual firewalls to enforce a network traffic flow from the legacy controller virtual instance to a physical network segment, used only for the legacy equipment.

Securing legacy components using micro-segmentation
Make disruption difficult

It’s important for the manufacturing industry to maximise up-time. Virtualisation has a number of features which can help with this. However, care must be taken with these features as they can have a negative effect if used incorrectly.

Our system design will incorporate features such as high-availability clustering and automated failover. These features have the added benefit of allowing the operations team to continually update virtual instances and virtualisation platforms while making disruption difficult.
Making compromise detection easier

Monitoring across all the layers of your architecture can be improved using virtualisation. For example, using virtual networking to enable flexible network-level monitoring, sending system logs to collectors via dedicated virtual networks, enabling logs to be sent from each compartmentalised segment to a central log collection and analysis platform.
Performance monitoring of your system’s virtual instances is easier too, as the virtualisation platform oversees their resource allocation. All of this helps to make compromise detection easier.

---

**Minimise the impact of compromise**

No matter how well you design a system you cannot be certain it will never be compromised. So, it’s a good idea to anticipate and be prepared for system compromise and failures.

You should put in place a response plan which makes it easy to recover and ultimately reduce the impact of compromise. Virtualisation can help with this too. By using features already deployed for high availability such as clustering...
and failover, but also using snapshots and backup which you can use to rapidly restore a system using orchestration.

The design process

I hope going through this simple design process has shown how useful the Cyber security design principles are for building security into your systems.

The base principles bring consistency to the design, and review, of your systems. The Virtualisation security design principles extend this concept to address the specific set of problems and strengths faced by virtualised systems.

PUBLISHED
7 August 2019

REVIEWED
11 May 2020

VERSION
1.0

WRITTEN FOR
Cyber security professionals
Large organisations
Public sector
Security architecture anti-patterns

Six design patterns to avoid when designing computer systems.

Introduction

At the NCSC, our technical experts provide consultancy to help SMEs and larger organisations build secure networks and systems.

This security paper describes some common patterns we often see in system designs that you should avoid. We’ll unpick the thinking behind them, explain why the patterns are bad, and most importantly, propose better alternatives.

This paper is aimed at network designers, technical architects and security architects with responsibility for designing systems within large organisations. Technical staff within smaller organisations may also find the content useful.

Download this security paper (PDF)

Terminology

A few quick points on terminology before we start.

Anti-patterns

The term 'anti-pattern' is now used to refer to any repeated (but ineffective) solution to a common problem, it is credited to Andrew Koenig who coined it in response to the seminal book 'Design Patterns: Elements of Reusable Object-Oriented Software'.

Trust

Computer systems rarely exist in isolation. That is, they connect to networks and other systems. You might trust some of these other networks and systems more than others, and the owners of those might not trust yours at all. We use the terms:
- **less trusted** (or **low side**) to refer to the system in which we have less confidence in its integrity
- **more trusted** (or **high side**) to refer to the system in which we have more confidence in its integrity

Information technology vs operational technology

When thinking about trust and integrity, we consider **administration** of information technology to have broadly similar requirements to the **operation** of operational technology. Our examples below focus on the more typical information technology examples, but we think many of the concepts can be used in operational technology environments too.

---

**Anti-pattern 1: ‘Browse-up’ for administration**

When administration of a system is performed from a device which is less trusted than the system being administered. Unfortunately it is all too common to see ‘browse-up’ approaches to administering systems, which proves that common practice isn’t always good practice. In such scenarios, an end user device used by an administrator can be one of the easiest paths into the target system, even if access is via a ‘**bastion host**’ or ‘**jump box**’.

In computer systems where integrity is important (whether in digital services which handle personal data or payments, through to industrial control systems), if you don’t have confidence in **devices** that have been used to administer or operate a system, you can’t have confidence in the integrity of that **system**.

There’s a common misconception that a bastion host or jump box is a good way of injecting trust into the situation, to somehow get confidence in the actions an administrator is taking from a device you don’t trust. Unfortunately, that’s not possible.

**Bastion hosts are useful for helping monitor and analyse the actions that administrators are performing, and they can help you avoid exposing more than**
one protocol outside of your system for administration purposes. But they won’t help you be confident that the user on the device is the person you intended to allow access to. Behind the scenes, the credentials used to authenticate to the jump box could have been compromised (a reasonable assumption, given the device is less trusted). Even if administrators are authenticating their sessions with two factors, there is still the potential for malware to perform session hijacking on remote desktop or shell connections in the same way that online banking sessions are hijacked. Having gained access, the attacker can perform additional actions on behalf of the administrator. The system is under their control.

How to identify this anti-pattern

Here are three ways you can identify browse-up administration:

1. By looking for administration activities performed via a remote desktop (or remote shell) from a device which is part of a less trusted system.

2. By looking for outsourcing or remote support connections where a third party uses a remote desktop or shell to reach into a network. If you’ve got confidence in the integrity of the device used by the third party, then this isn’t a browse-up problem, but if you have less confidence in their system than in yours, then it is.

3. Finally, any device which browses the web or reads external email is untrusted. So if you find an administrator using a remote desktop or shell to perform administration from the same processing context that they browse the web (or read their external email) from, then that’s browsing-up to o.
A better approach: ‘browse-down’

You should always use devices that you have confidence in the integrity of for administration of production systems. Those devices need to be kept hygienic (that is, they should not natively browse the web or open external email, as those are dangerous things for an administration device to do).

If, for convenience, you want to do those things from the same device, then we recommend that you ‘browse-down’ to do so. In a ‘browse-down’ model, the riskier activities are performed in a separate processing context. The strength of separation can be tailored to your needs, but the goal is to ensure that if an activity in the less trusted environment led to a compromise, then the attacker would not have any administrative access to the more trusted environment.

There are many ways in which you can build a browse-down approach. You could use a virtual machine on the administrative device to perform any activities on less trusted systems. Or you could browse-down to a remote machine over a remote desktop or shell protocol. The idea is that if the dirty (less trusted) environment gets compromised, then it’s not ‘underneath’ the clean environment in the processing stack, and the malware operator would have their work cut out to get access to your clean environment.

Further reading

- Microsoft Guidance: Privileged Access Management
- NCSC Guidance: Systems administration architectures

Anti-pattern 2: Management bypass

When layered defences in a network data plane can be short-cut via the management plane.

It’s good practice to separate management communications from the normal data or user communications on a network. In some system architectures, this would be known as separating the data plane from the management plane. However, whilst it is common to separate these types of communications with
network controls, it is a common mistake to only apply the defence-in-depth concept to the data plane. If the management plane offers an easier route to the ‘crown jewels’ of a computer system than the data plane, then this a management bypass.

How to identify this anti-pattern

Look for any management interfaces from components within different layers of a system, all connected to a single switch used for management, without the corresponding layers.

A better approach: layered defences in management planes

The solution is simple build similar layered defences into management planes to those you have in data planes. Good practices include:

- manage from a higher trusted device, browsing down to lower trust layers
- separate bastion hosts to manage systems in each trust boundary
- different credentials for different layers to help prevent lateral movement
- restrict how systems on the data plane communicate with management plane infrastructure and vice-versa
Further reading

- NCSC Blog: Protect your management interfaces (contains other tips on administrative access to systems)

---

Anti-pattern 3: Back-to-back firewalls

When the same controls are implemented by two firewalls in series, sometimes from different manufacturers.

There seems to be a widely believed myth that the security benefit of ‘doubling up’ on firewalls to implement the same set of controls is a worthwhile thing to do. Some also believe that it is preferable for the two firewalls to come from different manufacturers, their thinking being that a vulnerability in one is unlikely to be present in the other. In our experience this almost always adds additional cost, complexity, and maintenance overheads for little or no benefit.

Let’s explore why we see little benefit in back-to-back firewalls in almost all cases. Take the example of an OSI layer 3/4 firewall. It has a simple job to do; control which network communications can pass through the device, and which ones can’t. Putting two layer 3/4 firewalls in series is analogous to draining boiled potatoes with two colanders rather than one – it just creates more washing up.

But what if there was a vulnerability that can be exploited in a single firewall? Well, yes, that’s possible. There are vulnerabilities in most things after all. But firewalls don’t tend to have vulnerabilities that can be exploited to yield code execution from processing the header of a TCP/IP packet. They tend to have vulnerabilities in their management interfaces, so you shouldn’t expose their management interfaces to untrusted networks.

Even if there were vulnerabilities discovered in the data plane interfaces of a firewall, applying patches swiftly after their release would mean that any attack would need to exploit a zero-day vulnerability, rather than a well-known vulnerability. Furthermore, defence-in-depth design would mean that it should take more than a firewall breach to compromise sensitive data or the integrity of
a critical system, and needing two zero-days to be exploited puts the attacker’s capability level well beyond the threat model for most systems.

Having two firewalls would also double your admin overhead, and if you require two different vendors then you need to retain expertise in both, which adds still more cost and complexity. Plus, you have more infrastructure to maintain, and most of us find it hard enough to keep up with patching one set of network infrastructure.

However, there is one exception where we’ve found two firewalls to be useful; for supporting a contractual interface between two different parties. We cover this exception at the end of this section.

How to identify this anti-pattern

Look for two firewalls in series in a network architecture diagram.

A better approach: do it once, and do it well

One well-maintained, well-configured firewall or network appliance is better than two poorly maintained ones. We also recommend the following good practices:

- avoid exposing the management interfaces of network appliances to untrusted networks, and properly manage the credentials used with them
- have a simple policy configuration to reduce the chance of mistakes being introduced
- use configuration management tools to ensure you know what the configuration should be, so you can tell when it isn’t correct (a tell-tale sign of compromise or internal change procedures not being followed)

The one exception

There is one example of using two firewalls back-to-back that makes more sense; to act as a contract enforcement point between two entities that are
connecting to each other. If both parties agree on which subnets in their respective networks can communicate using which protocols, then both can ensure this is enforced by applying the agreed controls on a firewall they each manage.

Further reading

- NCSC Blog: Protect your management interfaces

---

**Anti-pattern 4: Building an ‘on-prem’ solution in the cloud**

When you build - in the public cloud - the solution you would have built in your own data centres.

Organisations taking their first step into the public cloud often make the mistake of building the same thing they would have built within their own premises, but on top of Infrastructure-as-a-Service foundations in the public cloud. The problem with this approach is that you will retain most of the same issues you had within your on-prem infrastructure. In particular, you retain significant maintenance overheads for patching operating systems and software packages, and you probably don’t benefit from the auto-scaling features that you were hoping you’d gain in the cloud.

**How to identify this anti-pattern**

Look for:

- database engines, file stores, load balancers and security appliances installed on compute instances
- separate development (and test, reference, production etc.) environments left running 24/7
- virtual appliances used without considering whether cloud-native controls would be suitable
A better approach: use higher order functions

Unless you're quicker at testing and deploying operating system patches than your public cloud provider is, you are probably better off letting them focus on doing that. Compare their track record of patching operating systems against your own, and judge for yourself.

Similarly, when it comes to patching database engines (or other storage services), their higher abstraction Platform-as-a-Service offerings are likely to be maintained to a level that many large enterprises will be envious of. Using higher level services like these means:

- unnecessary infrastructure management overhead is reduced
- you can focus on the things that are unique to your organisation
- your system is easier to keep patched to address known security issues

Further reading

- NCSC Blog: Debunking cloud security myths

---

**Anti-pattern 5: Uncontrolled and unobserved third party access**

https://www.ncsc.gov.uk/whitepaper/security-architecture-anti-patterns
When a third party has unfettered remote access for administrative or operational purposes, without any constraints or monitoring in place. Many organisations outsource support for some or all of their systems to a third party. This isn’t necessarily a bad thing, unless done without understanding and managing the risks involved. If you outsource administration or operational functions, you’re dependent on another organisation to keep your system secure. The staff, the processes and the technology of the third party all need to be considered.

Leaving the staff and processes to one side for the moment, if a third party is administering your system, they will require access, often remotely. It’s common to allow third parties to have access through a bastion host, either over the internet from allow list locations, or over a private network. However, there are often not enough controls in place to limit the operations that can be performed via the bastion host. If this is the case, and a bastion host (or the device used by the third party) is compromised, then an attacker could gain significant access to connected systems.

Let’s take an example. Suppose you have purchased some niche technology that comes with a specialist support contract where the vendor needs remote access to support the device. In this case, the support organisation only needs access to the component they are supporting, and not to any other parts of your system. If you provided a bastion host that gave access to an internal network (and relied on their processes to only access the component they supported, rather than technically enforcing that process), then a breach of the supplier’s system (or of your bastion) host would be much more damaging than it could have been.

By locking these accesses down and efficiently auditing the connection, the risk of third party compromise can be greatly reduced.

How to identify this anti-pattern

It’s often possible to identify these relationships with third parties by looking for ‘umbilical cords’ out of network diagrams.
A better approach: a good contract, constrained access and a thorough audit trail

A good approach includes the following:

1. Choose third parties carefully with a sensible contract that sets out the controls relating to the people, processes and technology you need to have confidence in.

2. Constrained access following the principle of least privilege; only allow remote and authenticated users to have logical access to the systems they need to reach.

3. Ensure you have the audit trail you need to support incident response and support effective protective monitoring. When it comes to incident response, will you be able to confidently know which commands were executed by which user from the third-party supplier?

We also recommend the following good practices when designing a remote access solution for third parties:

- ask your supplier how they prevent attackers moving laterally between their other clients and your systems
- ensure that remote support staff use multi-factor authentication and ensure they do not share credentials – this will help you account for individual actions in event of a breach
- provide separate isolated third party access systems for different third parties
• consider using a just-in-time administration approach, only enabling remote administrative access in relation to a support ticket that is being actively worked on

Further reading

• NCSC Guidance: Assessing supply chain security

Anti-pattern 6: The un-patchable system

When a system cannot be patched due to it needing to remain operational 24/7. Some systems need to run 24/7. A lack of foresight could mean a system can’t have security patches applied without scheduling a large window of downtime. Depending on the technologies and the complexity, it may require a window of hours (or days) to apply a patch, which could be unpalatable length of operational downtime. As time goes by, the option to defer applying security patches could mean you’re left with huge number to apply during a maintenance window. Applying so many patches has now become too much of a risk, so you’re trapped in a vicious circle with a system that’s virtually un-patchable.

How to identify this anti-pattern

Look for a lack of redundancy within system architectures. Systems which require all components to be operational at all times do not lend themselves to phased upgrades, where the system could remain operational whilst undergoing maintenance.

The lack of a representative development or reference system (or ability to quickly create one) can signify a related problem. If the system owners have no confidence that the development or reference system is similar to the

https://www.ncsc.gov.uk/whitepaper/security-architecture-anti-patterns
production system, then this can contribute to a fear of affecting stability by patching.

A better approach: design for 'easy' maintenance, little and often

One of the NCSC’s design principles is to design for easy maintenance. In some systems, this could mean ensuring you can patch a system in phases, without needing to disrupt operations. Whist this would likely require higher infrastructure cost, some of the overall lifetime costs could be lower when factoring in:

- fewer, shorter outages
- reduced risk of compromise (which could incur a costly incident response)

Further reading

- NCSC Guidance: Vulnerability Management
- NCSC Blog: Time to KRACK the security patches out again

PUBLISHED
22 May 2019

VERSION
1.0

WRITTEN FOR
Large organisations
Cyber security professionals
Public sector