To Patch or Not to Patch?

A Decision Process for Applying Patches to Software in the Safety Critical Domain

August 2019

Author Thomas Turner

Supervisor – Richard Hawkins

This is a report on a project submitted for the degree of MSc Safety Critical Systems Engineering in the Department of Computer Science at the University of York.

Abstract

Safety critical systems have been growing in complexity and many are now connected to networks which is eroding the traditional air gap argument, this combined with the increased use of commercial off the shelf software is leading to a situation where they are vulnerable to attack. Many systems with vulnerabilities have patches available which can start to mitigate these attacks but applying patches comes with its own risks. This project investigates how a decision can be made whether to apply patches or not and how to evaluate the risks associated. The work concludes that a patching decision can be made by using a risk based process and proposes a Goal Structuring Notation pattern that can be used to justify the patching decision.

Acknowledgements

I would like to thank the following people for their help with this work.

- Jane Fenn, BAE Systems mentor
- Steve Porter, supportive manager

Ethical Statement

The work presented in this report have been conducted in line with the ethical requirements from the University of York [1]. There a number of Subject Matter Experts have been asked their views on the proposed process and some of these have been asked to trial the process proving feedback in the form of questionnaires. The participants were asked for consent to use their views and experiences and all data has been anonymised. All information has been obtained legitimately and referenced appropriately. No harm to any personnel, equipment or organisations has resulted as a conduct as part of this project.

This project proposes a process for assessing the need to apply a patch to safety critical and safety related systems. It has been evaluated by a limited number of sources from limited domains. The process is misused may result in patching decisions being made incorrectly potentially leading to safety or security issues. The process represents the author's best efforts to define a process to aid in making a patching decision; however it must be used with caution to ensure it is applicable in the specific instance.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

As safety critical and safety related systems become more widely connected, and more Commercial Off The Shelf (COTS) software is being used, safety critical and safety-related systems are becoming vulnerable to more cyber-attacks, this is supported by Johnson [1] who concludes in his paper 'Why We Cannot (Yet) Ensure the Cyber-Security of Safety-Critical Systems'. Stakeholders are starting to argue that if the system is not secure against cyber-attacks, the system cannot be argued as being safe [2].

The security industry currently works on a process dubbed 'Penetrate and Patch'. In Penetrate and patch once a vulnerability is identified, a patch is produced by the vendor and then released to allow the operators of affected systems to update their systems. While many people consider Penetrate and Patch a poor method of security [3] [4] [5], it is likely that it is here to stay. Within the traditional IT domain, software is patched regularly, for example the phrase 'Patch Tuesday' has been coined for the regular updates to Microsoft Products [6] and the advice and practice is generally to apply all patches immediately.

Traditionally within the safety critical domain software is written once and then only updated as part of midlife updates or is often never updated. This is likely to do with the development lifecycle imposed by many safety processes that adds large amounts of development assurance through design traceability, code reviews and various methods of verification and validation. Due to this development process it becomes unfeasible to make lots of updates to the software within a safety critical system.

The unwillingness to apply patches to safety systems is leading to a patching gap and so vulnerabilities that have been known about by the industry for years and are patched on traditional IT systems may still be open on safety systems [7]. This patching gap is becoming more of an issue as the safety critical systems are being connected to networks, thus eroding one of the traditional 'air gap' arguments that has been used to justify why the system does not need patching.

This disparity of cultures can leave safety related system open to cyber-attacks and the assertion from Bloomfield et al. 'If it's not secure, it's not safe' [2] and the risk of a cyber-attack is undermining the safety case for a system.

While we have seen some attacks on safety critical systems with attacks such as STUXNET [8] which was directly targeted against a safety critical system and WannaCry [9] which affected safety critical infrastructure such as the NHS, we have not seen as many as security researchers may lead you to believe are possible. It is the authors opinion that this is due to the lack of motivation of the attackers. At the present time there is little motivation for attackers to launch attacks on safety critical infrastructure, but this could change at any moment. We need an approach to argue that safety critical systems are secure so that we can argue that they are safe.

Safety and security both use a risk based approach, analysing the systems for issues that can lead to an unwanted event. Safety can use a quantitative approach, applying numbers to the likelihood of an event happening and the severity of the outcome. Security, in contrast, generally uses a qualitative assessment as it is very hard to put numbers on the likelihood of a threat being realised. These two assessment types are very hard to compare with much real meaning which makes the comparison between safety and security risks hard, this leads to one type of risk potentially being artificially inflated over the other.

1.1. What is a Patch?

In the context of this report, a patch is any update to a piece of software which is released to fix a vulnerability. It can be seen as a special type of update that is required to be made to the software to remove a known vulnerability which could lead to compromise of the safety critical system. The difference between a patch and a normal change is the speed at which it needs to be performed which might lead to a reduction in the development assurance activities that are carried out before the patch is applied.

1.2. Scope

This project will look at the decision process around patching safety-related and safety-critical systems across multiple domains. As there is a move to using COTS and more mainstream operating systems within the safety critical domain, consideration will be given to non-safety critical software. The process will start from the point that a vulnerability or patch is discovered up until the point that normal operation continues.

2. Literature Survey

Within the literature survey carried out for this report [10], it was identified that there is no clear guidance given on how to decide whether or not to apply a patch to a given system. Most of the guidance is to either patch or not to patch without consideration of the residual risk from either the patched software or the vulnerability, respectively. The

Department of Homeland Security (DHS) do begin to define a process, however this does not go all the way to answering the question. A more in-depth critical evaluation of the DHS process [11] is presented in section 4.4 of this document.

3. Problem Analysis

As discussed in the introduction there is little guidance for making a patch or don't patch decision for safety critical or safety related systems, most of the advice is patch or don't patch as a rule. Both patching and not patching a system changes the risks associated with the system. Patching the system can alter the behaviour of the system, potentially introducing new failure modes or creating new paths to hazardous conditions, whereas not patching a system leaves it open to attack. If it is accepted that if a system is not secure it cannot be safe, doing nothing about known vulnerabilities in the system leaves us with an unsafe system. If we accept this, it is important that as vulnerabilities are discovered, either by security researchers or by our own testing, we must assess the risks and make a decision on applying or withholding patches to the system.

The literature survey in [10] and section 2 identify that although the DHS report is the closest to providing guidance on patching or not patching, it does not fully answer the question as it provides a patch now or patch later decision.

3.1. Project Goals

The goals of this project are to;

G1. Allow a good decision to be made about applying patches to systems in a safety critical or safety related context
G2. Allow patching decisions to be justified

3.1.1. Goal 1 Success Criteria

There is currently little guidance provided to engineers about making a decision to patch or not to patch when a vulnerability is discovered, or a new patch is released. Most of the guidance is binary, either always patch or never patch. A good decision needs to be a decision made on the balance of risk between patching and not patching vulnerabilities. A good solution to the problem will allow an engineer to assess the risk of both patching and not patching and allow them to compare the level of risk for both options. To confirm that the solution allows engineers to make good decisions the process should be run a number of times within different domains and different scope of change. A good decision is one that is accompanied by a compelling justification.

The decision needs to be able to be made by an engineer with limited safety and security knowledge. This is important as the decision is likely to be made by system operators which may not have detailed knowledge or experience of safety and security processes. A

good solution should be either self-explanatory or simple to follow. To confirm that the solution allows engineers with limited safety and security knowledge to make the decisions when running the process, the process should be run by people with different levels of knowledge and experience. The outputs from those with a good level of safety knowledge and experience, a good level of security knowledge and also engineers with limited safety and security knowledge, should be similar. If the output is vastly different, it may indicate that the process is calling on large amounts of domain knowledge to help make the decision which could mean that the process cannot be easily followed by engineers with limited knowledge on safety or security.

One of the issues identified in the literature review was the overloading of the terminology between safety and security. The solution will need to be explicit in the meaning of terms that are overloaded between safety and security. To demonstrate that this has been met when giving the process to be run by people with different knowledge levels, they should not require any additional explanations of the terms.

Summary of Success Criteria for Goal 1

GS1.1. Guidance to allow a patching decision to be made
GS1.2. The guidance should be usable by non-security and safety specialists
GS1.3. Terminology in the guidance should be unambiguous

3.1.2. Goal 2 Success Criteria

The patching decision for safety critical or safety related systems must be able to be justified to the stakeholders of the system. This is important as the patching decision potentially undermines the safety case that has already been presented for the system.

In DEF STAN 00-056 [12], the Ministry of Defence requires a Safety Assessment, or safety justification that "consists of a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment." The justification for the patching decision should meet this requirement. While other industries do have different definitions of what a safety justification is, they all generally define an argument that the system is safe, supported with evidence that can be defended.

The justification should use the output of following the guidance generated as part of goal 1 to allow the decision to be supported. If additional evidence is required to be produced to help support the justification it would indicate that the process used for goal 1 is not adequate as if the decision cannot be justified with the output of the process, the decision should not have been made. To confirm that the justification can use the output from the process used to make the decision, the process should be followed to create a decision and then a justification generated for the decision only using the outputs from the

process. The justification should then be reviewed by both a safety and security expert to ensure that they both can understand the decision made.

The justification of the decision should be able to be followed by a non-safety specialist, leaving them with the confidence that the right decision has been made. To confirm that the justification is adequate, once the decision process has been run, the justification should be given to a non-specialist to review ensuring that they understand the decision made.

Summary of Success Criteria for Goal 2

GS2.1. Provide a justification for a patching decision
GS2.2. The justification should be able to be followed by a non-specialist
GS2.3. The evidence required for the justification should come from the output of goal 1

4. Development Process

This section looks at the potential solutions to the problem:

4.1. Patch as a rule

Patch as a rule states that all patches should be applied immediately as soon as they become available from the supplier.

4.1.1. Advantages

This method ensures that vulnerabilities are dealt with as soon as possible reducing the potential vulnerabilities that can be exploited by an attacker.

4.1.2. Disadvantages

Patching early may lead to a reduced level of testing of the patch potentially leading to unintended operation or interactions.

With many embedded systems it can be hard to perform the update because the system is inaccessible and downtime of the system is very costly or may have safety implications, such as with medical devices. The patch as a rule may lead to a large increase in cost of the patching as there may be a high number of patches to deploy. Some of the patches that will be deployed onto a system with a patch as a rule may not be required as the vulnerability may not effect the specific implementation.

With some connected systems, patches can be set to be automatically applied, such as with the later versions of the Microsoft Windows operating system. While this automatic patching does help to ensure that all patches are deployed in a timely manner, it introduces a number of issues: availability, remote update vulnerability. When the system

performs the automated patch process, a system restart maybe required, as with many of the Windows updates. This means that the system will be unavailable during the reboot process.

If a patch is able to be applied remotely, it is possible that an attacker may be able to undermine the update process and push their own malicious update to the system. This can be seen in the "ShadowHammer" attack on ASUS [13] where malware was delivered to up to a million users via the ASUS Live Update.

While the original supplier is supporting the product and releasing patches, it maybe possible to apply released patches, trusting that the patch is not malicious and will not adversely affect the system. As safety systems have long expected lives, 30 years plus is not unusual, an original supplier may no longer support the the product, for example Windows generally has 5 years of mainstream support and an additional 5 years for extended support [14] or they may have gone out of business, this leaves the system without patches from the supplier. Some third parties do provide patches for systems that no longer have support from their suppliers [15]. While using these third party suppliers may help to mitigate the risk of having vulnerabilities in the software, it does pose the new risk of malicious updates leading to the question, do you trust the supplier?

4.2. Don't Patch as a rule

The option of not patching is how many safety-related systems are operating; leaving systems at the original know 'good' state. If vulnerabilities are identified they are left in the system, possibly adding them to a slower update cycle along with functional updates to the system.

4.2.1. Advantages

If the system is left at the known good state, the operation of the system is known and will have been tested throughly. This reduces the risk of incorrect operation due to incompatibilities of a patch along with other patching issues such as down time to apply the patch and bugs introduced by the patch.

There is no cost to not applying patches unless the system is successfully attacked. In contrast applying patches to a system can lead to the cost of developing and testing the patch, the cost of applying the patch and the potential downtime of the system while applying the patch (reboots, etc.).

4.2.2. Disadvantages

By not applying patches the system is potentially left open to an attacker to exploit a vulnerability. This can lead to any number of negative outcomes such as: unplanned downtime, overriding of safety systems, denial of service, breach of security goals, it is

also possible that even if the safety of the system is not directly affected, timings or other potentially critical functions may be affected. One of the signs of being infected with malware is a slow running computer [16] as the malware is using the systems resources, this means that the system may not respond in the required time frame to meet the safety goals.

One of the arguments in favour of the don't patch as a rule approach of dealing with patches is that many safety systems are isolated, air gapped, from other networks meaning that the likelihood of the vulnerability being exploited is deemed low. This argument is being broken down, as the literature survey [10] discusses, many systems are now being more widely connected to allow monitoring or for other business reasons and in some cases systems maybe connected without the knowledge of the system operators.

4.3. Don't patch but apply mitigations

This is an approach taken by some safety critical systems, where although there is no drive to apply patches, there is a understanding that some mitigations may be required to continue operating safely. Mitigations, or workarounds, vary from changing processes used as part of operating or servicing the system, to applying additional layers of protection to form defence in depth or making other physical modifications to the system such as disconnecting from the Internet to reduce the likelihood that the vulnerability can be exploited.

4.3.1. Advantages

By applying mitigations rather than applying a patch, it is possible to reduce the likelihood that a vulnerability can be exploited. The advantage over applying a mitigation rather than applying a patch is that a mitigation does not have to change the system helping to reduce potential for unwanted side effects of the patch.

4.3.2. Disadvantages

As mitigations are added to the system, there is a possibility that there is a reduction in the functionality of the system or an increased workload to the users of the system. If there are changes to the workload of the users the overall safety of the system may change, this is supported by the HSE report [17] where it states: 'It is rare for a report to state that an accident resulted from deficiencies in manpower numbers. However training is often mentioned and heavy workloads on operators during process upsets is referred to frequently, especially in connection with "alarm floods".'

4.4. Decision Process

A decision process for allowing a decision to patch or not patch a system to be made on a case by case basis can help to realise the benefits of both patching and not patching. At

the point that a vulnerability or patch is identified a risk-based approach is used to decide whether to apply the patch or not.

Identify patch -> Assess risk -> Is risk of patch greater than risk of vulnerability?

Yes -> Apply appropriate mitigations -> continue operation

No -> Apply patch -> continue operation

As identified in the literature survey [10], the Department of Homeland Security (DHS) have produced a decision process to allow operators to decide on when to apply a patch. Their decision process does not allow for a mitigation to be applied but does provide a starting point. This process is shown below:

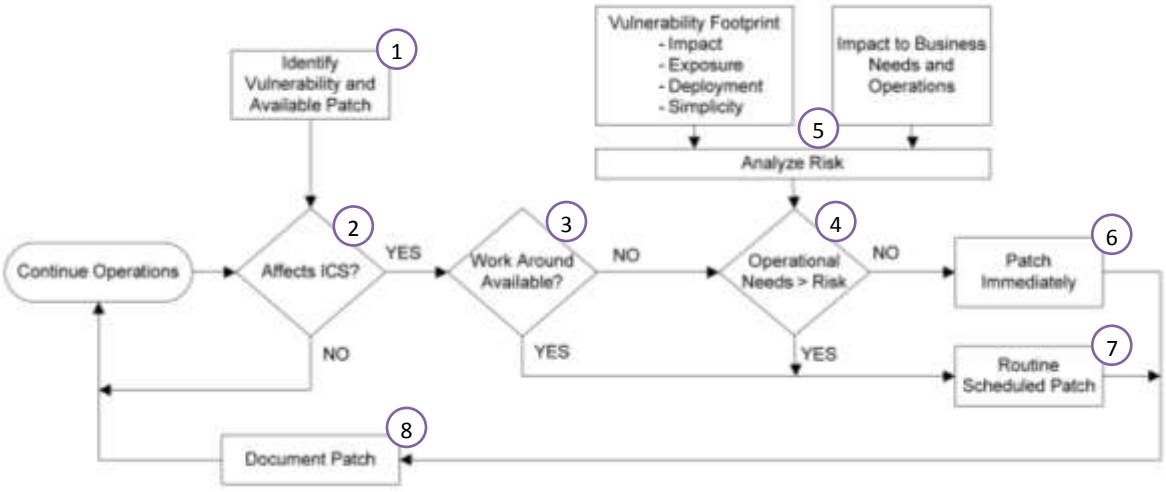


Figure 1 DHS Patch Urgency Decision Tree [11]

Whilst this process does allow for a decision to be made regarding a patch, it is focused on patch now or patch later. There are a number of issues that the author has identified with this process;

1. This step is the starting point for the DHS process. The step requires a vulnerability and a patch to be identified. There may be occasions where there is no patch available, however the vulnerability is still present and requires some attention. The author agrees that the process needs to start with the identification of the vulnerabilities and patches. Although some guidance should be offered to help users of the process identify them. There is a question over what methods should be used to identify vulnerabilities and find available patches. This is important as if the method of identifying patches and vulnerabilities is not robust enough, there will always be an unknown risk of the known vulnerabilities and patches that have not been identified by the team managing the patching. This is important as as identified in the literature survey, once a vulnerability is discovered or patch released the

likelihood of the vulnerability being exploited goes up as attackers can reverse engineer the patch to exploit the underlying vulnerability.

2. The DHS process focuses on Industrial Control Systems (ICS) whereas the process developed to be needs to be inclusive of safety-critical and safety-related systems. This step of the process asks if the System of Interest (SOI) is affected, it maybe more appropriate to ask how is the SOI affected as there may be no or negligible safety affect. There also may be a combination of vulnerabilities that could combine to have a more serious affect.
3. Step 3 asks if a workaround is available, it is important to consider here how the work around affects the system, it is possible that the work around could introduce more risk than the vulnerability itself.
4. The steps around 4, (Vulnerability Footprint, Impact to the business needs and operations and analyse risk) forms the risk assessment for the vulnerability. The following is a generic risk component model for security from the University of Oxford [18];

Figure 2 Generic Security Risk Component Model

The risk component model appears to be compatible with the DHS process where the Vulnerability footprint is the right-hand side of this model and the business impact is the left. This shows that this is risk assessment step. This is a complex step as it requires an assessment to be made over the safety and security impact of both applying the patch and not applying the patch. There are many conflicting requirements here and so some additional guidance may be required to help generate the assessment. This step also needs to consider the amount of testing and confidence that is held in the patch, i.e. official large vendor vs open source patch and full regression testing vs functionality testing of the patch.

5. Step 5 asks if the operational need outweighs the risk of the vulnerability. This trade off must be considered, however it is also important to consider the longer-term impact. While in the short term the risk may be considered to be acceptable in the context of the operational need, over time the time at risk may increase to the point where the risk is unacceptable. Also the likelihood of the risk being realised may increase, as discussed in the literature survey [10], once a vulnerability or a patch is

released malicious attackers will start to attempt to write code to automatically exploit the vulnerability. Given this, it is important that if it is decided that the operational needs of the system outweigh the risk that this is reassessed to ensure that the balance is not tipped.

6. This step implies that if there is the risk outweighs the operational need then the patch should be applied immediately. As discussed in the first step, it may be that there is no official patch available and so other methods of reducing the risk may need to be applied.
7. This step implies that there is always a need to apply the patch if the system is affected by the vulnerability, there are a number of things to consider here; How often is the routine patch cycle, safety critical systems often are only ever updated once or twice in the system life as development and testing can be expensive as well as the downtime to the system. It is important to consider what is the effect to the system, whilst the system may be effected it is possible to argue that the likelihood of someone malicious gaining physical access to the system is so small and there are easier ways to attack the system that it is acceptable with the vulnerability so there may not be a need to apply the patch.
8. This step requires the patch to be documented into the configuration management. Whilst this step is very important, it is also important to document a no patch patching decision which is not included in the DHS process. Without documenting the no patch decision where the risk is being accepted as it has been assessed to not affect the system in the current configuration or use with a change in configuration or use it may be hard to find and relocate all unpatched vulnerabilities.

4.4.1. Advantages

The decision process guides those who need to make a decision to achieve the benefits from the choice made, patch, no action or mitigation.

4.4.2. Disadvantages

If you apply the decision process there could be a delay in applying the patch, if decided to do so. This delay can potentially lead to a longer than necessary time at risk. When either applying all patches or no patches it is easy to identify the configuration of the system, whereas when applying patches selectively, the configuration of the system must be documented carefully otherwise it is very easy to become unaware of the exact configuration, this could result in patches being applied when they are not needed or patches not being applied as it is not thought it is relevant to the system configuration.

4.5. Solution Comparison

A number of solutions have been identified to the patching problem, the first two, patch as a rule and don't patch as a rule, summarise the polar opposites, in general IT systems

are patched because they are connected and because they are connected are constantly subjected to attack as new vulnerabilities emerge. Embedded systems knowledge of vulnerabilities and there potential for exploit is only just emerging which is challenging the don't patch as a rule approach. Both of these options have serious cons that lead the author to conclude that they are not suitable to be used within a safety critical or safety repeated system. Applying mitigations, or workarounds, to allow for a patch to not to be implemented but reducing the risk does allow for the risks to be reduced but also can mean that there is an increased work load for operators as they have to use the new work arounds, the effectiveness of the mitigations may also fade with time as computing power increases allowing for complex attacks to become trivial or as the system is modified and potentially connected to other systems. A decision process allows a decision to be made using the advantages of all the options whilst minimising the disadvantages.

This report will update the DHS decision process, mitigating some of the issues identified with it and providing guidance to allow the process to be applied in both IT and OT systems.

4.6. Patching Justification

Safety cases are generally used across the industry to support the claim that a system is acceptably safe to use in the given context. T. Kelly in Arguing Safety – A systematic Approach to Managing Safety Cases [19] presents Goal Structuring Notation (GSN) as a method for helping present safety arguments more clearly than text arguments. A text-based safety case is possible, but the English language can be ambiguous sometime and large number of cross references in a body of text can be very hard to follow. Whilst there are other graphical methods of representing a safety case, for example, Claim Argument Evidence, GSN is one of the most widely recognised. For these reasons, a GSN justification for the patching decision will be used.

The GSN standard [20] has extensions for a GSN patterns. A GSN pattern allows for a generic GSN argument structure to be instantiated when required. A GSN pattern can be produced for arguing that the patching decision is appropriate that then can be instantiated for each vulnerability that is assessed. The output of the decision process should generate evidence to support the claim that an appropriate mitigation for the vulnerability has been applied to the system.

5. Implementation

The following process is a development of the DHS patch urgency decision tree. It has been updated to focus more generally on safety critical and safety-related systems rather than specifically on ICS and to make some of the hidden steps more explicit. It also allows for other mitigations to be implemented, or a decision that no action should be taken to be made. It address all of the issues discussed within section 4.4.

For each step of the process there is a description describing the step with any guidance, there is also an input and an output section listing what artefacts are required or are generated as part of the step. The final section of each step of the process is a running example that demonstrates the process being used. The example system is for an organisation developing safety critical and safety related systems. They have IBM DOORS running (a requirements tool allowing for capture, trace, analysis and change management) which is open on the Internet to their customers to access. Customers can login to DOORS to view and update their requirements, depending on the lifecycle phase. The system is described in detail in Appendix B – Example System Description.

The following glossary of terms is used to define the specific meaning of words and terms throughout this process.

Term	Definition
Attacker	Someone who is trying to use the system in an unauthorised way
Bugs	A vulnerability or issue with the code that causes functionality issues with the system
Cost	A price paid to do something, this could be financial, time, exposure to risk, etc.
Failure	Something that has gone wrong
Justification	The argument that the patching decision is correct for the vulnerability and the system
Mitigation	Changing the system or the system use (work around) to remove or lessen the effect of the vulnerability
Patch	A software update to a system to fix a vulnerability. This could be a small update that does not require a full reinstall, or a update that requires full reinstallation of the software
Patching Decision	The decision that has been made (No action, patch, apply mitigation) for the specific vulnerability
Qualitative Assessment	An assessment using words rather than numbers to evaluate the results
Quantitative	An assessment using numbers to evaluate the results
Risk	Any risk to the system, safety or security
Root Cause	The initiating cause of the vulnerability
Safety Risk	A situation that is exposing someone or something to danger, harm or loss. It can be defined as probability of situation occurring * severity of the potential harm.
Security Risk	Similar to safety risk, however, consideration must be given to loss of confidentiality of data
Vulnerability	An issue within the system that could be exploited by a malicious actor
Workaround	See mitigation

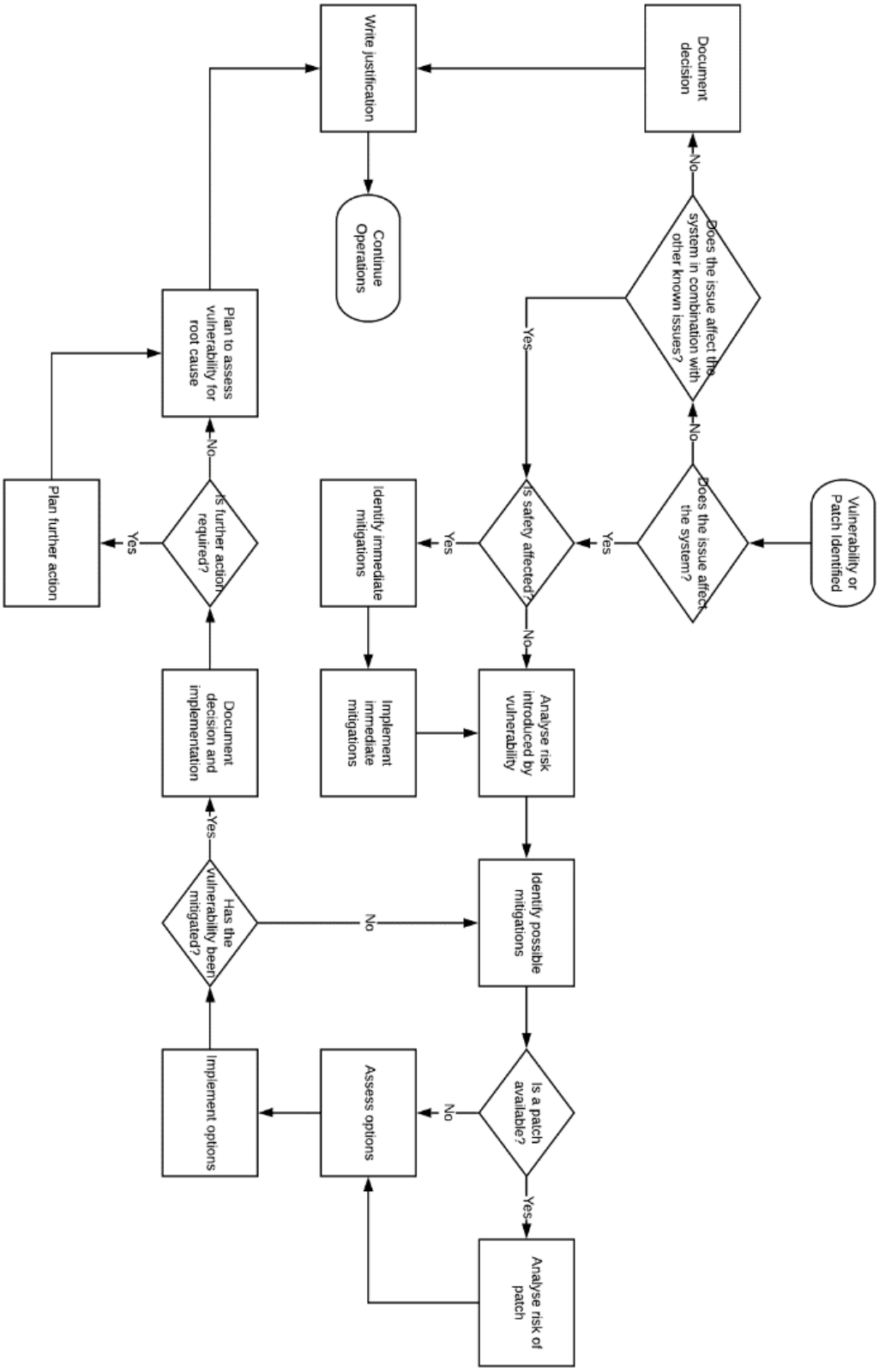


Figure 3 - Patch Decision Process

The following sections provide details on what is expected at each stage of the process along with an example which will be used throughout the document.

5.1. Vulnerability or Patch Identified

The process may start with a vulnerability being identified, there are several ways that this could be identified, including: Standard testing, vulnerability database scanning [21], alerts from suppliers, alerts from communities (for example the CISP Share website [22]) or alerts from the public (i.e. bug bounty). It is important that information from all of the resources are collected and analysed. Like a vulnerability being identified, there are multiple ways for a patch to be identified. Patches may be released from the supplier, or from third parties if the original supplier is no longer supporting the product. As discussed in the literature survey [10], once a patch has been released, hackers will reverse engineer that patch to create an exploit, so it is important to respond quickly to a patch being announced.

If the process is started with a patch, it is important to get information about the vulnerabilities that the patch addresses. This should be able to be obtained from the patch supplier. One of the issues with patches is that they often contain fixes for multiple vulnerabilities or add new features to the software, this increases the size of the patch potentially increasing the risk of the patch itself and new features can sometimes interfere with the system itself. To deal with this, vulnerabilities should be assessed separately to calculate the risk of the vulnerability and the potential risk of the other mitigations. When assessing the mitigations in step 5.12, consideration about the patch fixing multiple vulnerabilities should be added to the assessment.

The process for ensuring that vulnerabilities are identified and analysed should be developed and agreed when the system is first put into service as the this must form part of the argument for the operational system. If the method for identifying patches and vulnerabilities is poor the argument about the safety of the system is undermined.

Inputs

The inputs to this step are the vulnerability databases, supplier, community and public alerts and any other method of identifying patches and vulnerabilities.

Outputs

The output of this step is a description of the vulnerability. Depending on the source of the alert, there may be some sample exploit code to help feed into the testing.

Example

The following example shows a real vulnerability that has an effect on the security of the system. This has been taken from the CVE database [21].

CVE Number	CVSS	Type	Software
CVE-2018-1912	5.4	Cross Site Scripting	IBM Doors

Table 1 Example vulnerability report

This vulnerability will be used throughout the example sections of the process to demonstrate the process.

5.2. Does the Vulnerability Effect the System?

It is possible that the reported vulnerability has no effect on the System Of Interest (SOI). An example of a vulnerability that may not affect the system is if Microsoft released a patch for vulnerability in a service, such as the fax service, that has been disabled on the SOI the vulnerability would not affect the SOI. It is important to have detailed configuration management on the SOIs to enable this step to be carried out.

The effect to the system could be to safety or security, even if the vulnerability only affects the security of the platform. It is still important to go through the patch decision process as the vulnerability is posing a security risk, but the patch itself may pose a safety risk.

If the process has been started by a patch being released, it is possible that there are no vulnerabilities being fixed by the patch that are relevant to the configuration of the SOI. Patches can be released to improve product performance or to add new features. It is important that these patches are only considered if they have a desired function or if there is a requirement to install them to install another patch that is required for safety or security reasons.

Inputs

Description of the vulnerability being assessed and a detailed description of the SOI including software configuration.

Output

A decision whether the vulnerability affects the SOI. If the vulnerability does affect the SOI, it is useful to detail how the vulnerability may affect the system.

Example

CVE Number	System Affected	Effect
CVE-2018-1912	Customer Requirements Web interface	Customers can add JavaScript to the DOORS to disclose database credentials. This could allow a malicious actor to login as another user with higher levels of access.

Table 2 Example vulnerability affects

There is unlikely to be any direct safety impact to the requirements database in DOORS, although if the DOORS database is being used to hold safety requirements, there is a possibility that safety of the systems under development could be affected.

5.3. Does the vulnerability being assessed when in combination with another vulnerability effect the system?

While there may be no immediate issue with the vulnerability being assessed, it is possible that in combination with other known unpatched vulnerabilities that the vulnerability could be exploited. An example of this could be, if, like in the previous step the vulnerability is with a disabled service, but there is another unpatched vulnerability that allows that service to be remotely activated. While in isolation both of those vulnerabilities do not pose a significant risk to the system, in combination they present increased opportunity for exploitation, this is known as an attack chain.

Inputs

Vulnerability details and details of other unpatched vulnerabilities on the SOI.

Outputs

Decision if the vulnerability still does not affect or now in combination does affect the SOI. If in combination the vulnerabilities affect the SOI, both vulnerabilities should be taken through the decision process together. This allows for a decision to be made on patch both, one, or other mitigation to be applied based on the residual risk. There may be more evidence for one patch as it may have been in-service for longer and so has more in-service evidence to show it works correctly.

5.4. Document the no Effect Decision

It is important to document the decision that there is no effect on the system as this implies that the patch will not be applied. If in the future a combination of vulnerabilities can make this exploitable, it is important that information on the vulnerability can be found in order to be re-assessed. There is also the possibility that a change to the use or configuration of the system may make previously assessed vulnerabilities as having no effect, suddenly becoming exploitable. The information about the vulnerability created in this step can then be fed into the change committee; an example of this is a previously disabled service with a vulnerability being enabled as part of an upgrade to the system, or a previously unconnected system being connected to the Internet.

It may be appropriate that this database of no effect decisions is reviewed on a regular basis to;

1. Ensure that the decisions are being made correctly.
2. To ensure that no changes to the configuration or use of the SOI has undermined the no effect decision.
3. To ensure that no development in computing ability has undermined the no effect decision.
   An example of this might be vulnerabilities within cryptographic algorithms. Here, when first assessed a significant amount of time to compromise the SOI could become trivial in the future.

Inputs

Description of vulnerability, configuration details of the SOI to allow comparisons to be made later. Minutes of any discussions leading to the no effect decision.

Outputs

Report or database entry for the vulnerability.

Notes

The output of this step may be considered sensitive as it will list all the unpatched vulnerabilities in the SOI. While the team who assessed the vulnerabilities may have come to a no effect decision, an attacker may be able to use the vulnerability in a way that the assessment team did not imagine. For this reason, the output of this step should be treated as sensitive data with appropriate controls around it.

5.5. Is Safety Affected

While the system is affected by the vulnerability, safety may or may not be affected. If safety is affected more immediate action may need to be taken. A vulnerability should be assessed if exploitation contributes to a safety hazard.

A technique that could help identify if exploitation of the vulnerability contributes to a safety hazard, is a Failure Modes and Effect Analysis (FMEA). The vulnerability can be considered to be a failure mode, the analysis should then decide what the effect of that failure is. It is likely that the 'failure' would not have been considered as part of any original safety assessments as historically safety assessments do not consider the malicious intent, only benign failure. This twisting to consider malicious intent may require a security expert to assist in this decision as they may have additional knowledge about how the vulnerability could be exploited.

While the vulnerability may not contribute directly to any of the hazards, it is also worth considering side effects of the vulnerability being exploited. For example, there may be additional load on the system which can result in in timing issues, availability issues, even effect environmental operating conditions, e.g. extra processor load causing heat build-up.

Whilst this process has been designed to be carried out by non-safety specialist this step may require input from a safety specialist to be able to make the judgment about if safety is affected.

Even if safety is considered not to be affected by the vulnerability, the rest of the decision process should be followed. The process will help to decide about applying the patch to ensure that security of the rest of the system is maintained and allow for a trade-off analysis to be performed about the down time of the system whist applying the patch. It

should be noted that even if the vulnerability itself does not affect the safety of the system, the patch itself could undermine the safety of the system.

Inputs

Description of vulnerability, safety hazard log.

Outputs

Safety analysis for the vulnerability and a decision whether the patch affects safety or not.

Example

The hazard log of the system shows a hazard H1, corruption of the safety requirements as a hazard, with the description of vulnerability and the hazard, it is possible to see that a malicious user could maliciously update and approve the safety requirements of the system thus potentially undermine the safety of the system.

Component	Failure	Subsystem Effect	System Effect	Comments
Safety System Design	Authentication around the DOORS database failed	(1) Anyone can access and update the DOORS database, inc. safety requirements	(1) The design of the safety systems is undermined by a malicious attack	Effect 1 can lead to significant safety impact on multiple programmes

Table 3 Example safety assessment

It is clear that, whilst there is no direct safety impact it is possible for the safety to be undermined so safety can be affected.

5.6. Identify Immediate Mitigations

If the vulnerability affects the safe operation of the SOI, it is important that immediate action is taken. This action may include drastic action such as shutting down the system, grounding the fleet etc. or it may be more appropriate to implement other mitigations such as a temporary air gap between the safety system and the system with the vulnerability.

It is essential to consider when adding any workarounds, the effect of these on the system and those who are operating it. While it may be acceptable to increase the workload for the operators for a short period of time, if this increase in workload is sustained there is an increased likelihood of a mistake possibly resulting in an accident.

It is also possible to accept the risk of the vulnerability at this stage until further analysis has been completed. This decision may be made if the need for operational effectiveness out weights the need to mitigate a minor risk introduced by the vulnerability.

Once mitigations have been identified, they need to be quickly assessed to ensure that they are appropriate to implement. This assessment should be approved by safety and security experts to ensure that the vulnerability will be mitigated, but also there is not a significant negative impact to the safety of the system.

Inputs

Description of the vulnerability, list of hazards that the vulnerability contributes to.

Outputs

Details of the rapid risk assessment used to decide about immediate mitigations for the system.

Example

Description	Advantages	Disadvantages / Safety Impact
Lock the Database for modification	Stops modification to add the exploit code	May be overridden by a user with elevated privileges May result in incorrect requirements being implemented
Only use configured documents	Ensures that only authorised requirements are used	Still allows for unauthorised modifications to be made May result in incorrect requirements being implemented
Stop any development	Ensures that no unauthorised modifications are made	Production stopped
No Action	Development can continue as normal	There is a possibility that the safety requirements could be changed without anyone's knowledge,

Table 4 Example immediate mitigations

5.7. Introduce Immediate Mitigations

Once the immediate mitigations have been identified, they must be approved and then be implemented. It is important that these mitigations are reviewed regularly whilst the rest of the process is being executed to ensure that they are still working correctly and are not increasing the risk to the system

Inputs

Description of the vulnerability, list of hazards that the vulnerability contributes to. The list of possible mitigations for the vulnerability.

Outputs

Mitigation for the immediate safety risk of the vulnerability. Plan for regular review of the mitigation.

Example

As the vulnerability has a potential effect on safety an immediate mitigation may be required. Although in this case the risk to safety is probably small and so the risk could be accepted some of the identified mitigations are simple enough to put into place. The most appropriate mitigation is to lock the requirements database for any modification and to only used configured documents for development until the vulnerability is fixed. This decision may need to be reviewed if the patching process is slow or there are changes within the organisation that require changes to be made to the DOORS database urgently.

5.8. Analyse Risk Introduced by vulnerability

This step involves a more detailed assessment of the vulnerability and the applicability of it to the SOI. It is important to consider both the safety and security risks introduced by the vulnerability.

To analyse the security impact of the vulnerability a security risk assessment needs to be carried out. The STRIDE assessment method developed by Microsoft, adapted by BAE Systems (Appendix A – STRIDE) may provide a useful assessment method to assess the threat to the system. The STRIDE assessment should look at what an attacker could do by exploiting the vulnerability.

S – Spoofing	Can an attacker now send data to the SOI that can force the system to perform an unwanted action? e.g. disabling error or bounds checking
T – Tampering	Can an attacker now tamper with the SOI to cause it to behave in an unwanted way? e.g. changing the system code
R – Non-Repudiation	Can the attacker now perform actions that cannot be linked back to them? e.g. deleting logs, or logging in as other users
I – Information Disclosure	Can the attacker now extract protected information from the system that they do not have access to? e.g. database dumps
D – Denial of Service	Can the attacker now block legitimate access to the system? e.g. switching off services
E – Elevation of Privilege	Can the attacker now change the level of access they have? e.g. logging in as another user or changing their level of access

Table 5 Description of STRIDE in the context of a Vulnerability

The STRIDE assessment then starts to fill in the risk component model as seen in Figure 2. Once the threats have been identified the impact can be worked out to the business. This can be combined with threat intelligence which can help to inform about the capability and motivation of the attacker which will be very dependent on the system that has the vulnerability. The frequency will depend on how the vulnerability can be exploited, if

through the Internet then this will be high, otherwise if physical assess is required and the system is locked away this will be much lower.

The safety risk will have been analysed in step 5.5, this may need to be performed in more detail as the assessment in step 5.5 may have been performed quickly to provide a quick decision.

Inputs

System safety requirements, system security requirements, description of the vulnerability.

Outputs

Assessment of both security and safety risk for the vulnerability

Example

Security Assessment

S – Spoofing	N/A
T – Tampering	N/A
R – Non-Repudiation	The attacker can expose other's credentials which may allow them to login to others accounts thus allowing them to carry out actions that are not attributed to them.
I – Information Disclosure	The attacker can login as others thus exposing information from other customers products
D – Denial of Service	The attacker can login as another customer and change the password thus denying the customer of the service provided.
E – Elevation of Privilege	The attacker can login using any of the credentials used during their attack, exploiting the privileges of those users.

Table 6 Example STRIDE assessment

The frequency of the attack can be considered to be high as the system is connected to the Internet for the customers to access their requirements.

The most likely attacker would be competitors of our customers attempting to get information about their competitors or to sabotage their development projects.

Safety Assessment

As per step 5.5.

5.9. Identify Possible Mitigations

This step aims to identify possible medium to long term mitigations for the vulnerability. These mitigations could be mitigations similar to those generated in section 5.6 for vulnerabilities that affect safety, but mitigations that take longer to implement can also be considered here such as developing an update to the vulnerable software or re-configuration of the system.

The aim of this step is to generate a list of mitigations, not to assess if they are feasible (due to cost, time, etc.) or how effective they are. The benefit of generating a large list of mitigations without bounding the list is that it is possible to argue that the chosen mitigation is the most appropriate. There may also be a possibility of implementing a number of mitigations to help make a layered security approach, also known as defence in depth.

While this step does not include patches that are available for the software that are already written, it should include the possibility of writing a patch in house or contracting a third party to write a patch.

Inputs

Risk assessment for the vulnerability, system design.

Outputs

List of mitigations

Example

- Do nothing
- Limit database access to only customer IP addresses
- Write custom patch

5.10. Is a Patch Available?

A patch may have been made available by the original supplier or a third-party supplier. If patches from multiple sources have been made available, these should be considered separately as they may have different risks, for example one patch may address multiple vulnerabilities thus potentially increasing the amount of code changed and potentially increasing the chance of new vulnerabilities being introduced into the system, whereas another patch may only deal with this one vulnerability and leaves the rest of the code base unaltered.

There are likely to more options for patches for open source applications as many people may be developing for them. The selection of the source of the patch implies a level of trust in the patch, it may be necessary to look at the provenance of the source of the patch to understand which one to choose.

Inputs

Vulnerability, patch alerts

Outputs

List of patches available for the vulnerability and the details of the patches.

Example

To fix this issue, the system must be updated to IBM DOORS 6.0.2 iFix019. There are no other patches available.

5.11. Analyse Risk of Patch

Depending on the source of the patch, the size of the patch, the rigor of the patch development process, the level of testing used to verify the patch, the patch will pose a level of risk. Both the safety and the security risk should be considered here as some patches can introduce new vulnerabilities into the system. In 2018 Microsoft release a patch for at the time unexploited vulnerabilities named Meltdown and Spector, however the patch introduced a new vulnerability which was far easier to exploit [23].

Factors to consider when considering the risk of the patch include;

- Number of vulnerabilities covered by the patch
- Other changes introduced by the patch
- How reputable is the patch source
- How much testing has been carried out
- Has the patch been implemented by other users
- Does the patch have any negative reports by other users

Inputs

List of patches available with details of each patch.

Outputs

Risk assessment for each patch

Example

The iFix019 contains fixes for 25 vulnerabilities. The source of the supplier is very reputable (IBM). There are no reports of the fix causing issues on other systems. While there is a risk to the system by installing the patch it can be minimised with appropriate testing.

5.12. Assess Options

This step brings together all of the information gathered from the previous steps and allows a risk-based decision to be made by comparing the various mitigations. The risk assessment needs to be a qualitative assessment. It is very hard to produce a quantitative risk assessment because the application of real numbers representing value or probability to security risk is difficult and then comparing them in a sensible way to safety risk adds an extra layer of complexity. Every potential mitigation will have advantages and disadvantages as well as the potential to introduce unknown risks. It is important to consider all three of these as well as any costs to implementing the mitigation, these costs could include;

- Time to develop the final mitigation, i.e. amount of time still at risk. This time at risk could be partly mitigated by implementing another interim workaround in lieu of the final solution
- Financial cost to develop the mitigation
- System downtime whist installing the mitigation
- Restricted functionality by using the mitigation

The risk assessment created as part of this step will form a large part of the patching justification.

Inputs

List of mitigations and patches available for the system.

Outputs

Risk assessment of the various mitigations with one (or more) mitigations being selected for implementation.

Example

Mitigation	**Advantages**	**Disadvantages**	**Known Unknowns**	**Effectiveness**
No action	*No new vulnerabilities introduced *No downtime	*Leaves a known vulnerability exposed in the system	*None	Low
IBM Patch	*Vulnerability is fixed, along with 24 other vulnerabilities	*Downtime during update	*It is not known how much testing has been performed by IBM *It is not known if any other functionality improvements have been made (undocumented) *Code change percentages unknown	High
Limit Access (IP based)	*Does not require change to the code	*Requires an extra step for authenticating customers *IP address can be cloned *Still allows authenticated customers to breach other customers'	None	Medium

		requirements		
Write custom patch	*Fixes vulnerability	*System downtime required *Will take some time to write	*May introduce new vulnerabilities * Financial cost to develop is unknown	High

Table 7 Example mitigation assessment

Both the IBM patch and the custom patch come out as high effectiveness but as the custom patch will cost more to develop and it is not known if it will resolve the issue, applying the IBM patch is more appropriate.

5.13. Implement Option

Once the action has been decided on, it needs to be implemented. There may be a number of patch decision cycles being run at once if a number of vulnerabilities have been identified, if this is the case, it may be appropriate to wait until there are a number of patches or mitigations to be applied rather than taking the system offline a number of times in rapid succession. This will need to be considered on a case by case basis as the risk of having the system exposed may be too high to wait.

As part of the implementation, relevant backups should be taken to ensure that if there is an issue with the mitigation that is put into place that it is possible to perform a roll back to a known state.

Inputs

Patch or description of the mitigation

Outputs

Updated system, or new process documentation detailing the mitigation.

5.14. Has the vulnerability been mitigated?

Once the patch or mitigation has been applied, it is important to test the system to ensure that the vulnerability has been mitigated. This needs to be done carefully especially if the vulnerability can affect the safety of the system. There may be example exploit code detailed along with the vulnerability or a custom test can be written. If the vulnerability does undermine the safety of the system, it may be necessary to run the test on a test system.

Another consideration here is any unexpected side effect of the patch or the mitigation. If possible, a full suite of regression testing should be run on the patched system. Some of the considerations of the side effects of the mitigation include;

- New bugs (issues) with the system
- New vulnerabilities introduced by the mitigation
- Changes to the timings of the system

Inputs

Exploit code, regression test scripts, penetration testing.

Outputs

Test documentation. Statement that the system is working.

5.15. Document Decision and Implementation

Once the implementation has been proved to be successful, it is important to ensure that all relevant documents have been updated to show the new configuration of the system. If this is not updated it may be hard to identify if future vulnerability announcements affect the system in its actual configuration.

Inputs

Patching decision and the specifics of the implementation.

Outputs

Updated documentation for the system.

5.16. Assess Vulnerability for Root Cause

It is possible that the vulnerability is an indicator of a more serious issue with either the safety/security assessment or the development of the system. It may be appropriate to review or plan a review of the vulnerabilities that have been identified to try to improve the safety/security assessment and or development process.

An example output of this step may be to consider moving away from COTS systems, developing bespoke systems that may have less vulnerabilities or may not be such a lucrative target for an attacker, if the system is bespoke they must be attacking the specific system rather than exploiting a known vulnerability across many systems as in the Wannacry attack [24].

Inputs

- Description of the vulnerabilities identified on the system to date

Outputs

This report does not investigate the methods used to assess for root cause issues. As such the output of this step would be a plan to investigate the root causes

Example

The vulnerability described in the scenario is due to the use of IBM DOORS. Whilst it is possible to rewrite the DOORS application, it is more likely that new vulnerabilities will be introduced. As IBM DOORS is a large application security researchers and IBM will continue to try to find issues with it and then once disclosed IBM will release a patch for them. There is no need to modify the system other than patching to remove the root cause.

5.17. Is Further Action Required

This step allows a decision to be made whether further action is required to fully mitigate the risk introduced by the vulnerability or the patch. A decision should be made based on the output of the previous steps where an assessment of the risks of the implemented solution has been carried out.

Inputs

- Description of the implemented solution
- Risk assessment for implemented solution
- List of other solutions
- Safety and Security goals for the system

Outputs

Decision if there needs to be further action to close the vulnerability assessment.

5.18. Plan Further Action

If it is decided that further action is required to;

- Fully mitigate the vulnerability or the risks introduced by patching
- Provide additional confidence that the system still behaves as required
- Deal with any root causes identified

This should be planned to ensure that it is completed.

There are many reasons why a permanent solution may not be implemented as part of the initial solution such as time or money constraints, further updates may require large amounts of system downtime, lack of expertise etc. This plan further action step allows for an improvement cycle to be implemented with the first improvement cycle being triggered by the patch or vulnerability being identified (5.1) and subsequent cycles being planned as part of this step.

Inputs

- Description of vulnerability / patch
- Risk Assessment
- List of proposed solutions

Outputs

The output of this step may be a plan to implement a specific known solution, or to perform specific testing, however it may not always be possible to define the further action here and an improvement cycle may be needed so the output would be a plan to investigate and develop more a permanent solution.

Example

As the root cause of the vulnerability is not being dealt with and the patch applied dealt with the vulnerability fully. The only additional action that may be required is additional regression testing of the system. In this case, it is unlikely that any additional action will be required as part of the process.

5.19. Write Patching Justification

It is important to provide a justification of the actions carried out as part of the process to demonstrate that the systems primary safety case has not been undermined by the patching decision and any update made to the system.

As discussed in section 4.6 GSN has been used to write a justification for the patching decision. The top goal is that 'An appropriate mitigation has been applied to the system'. The purpose of this goal is to justify that whatever the mitigation that has been applied (patch, no action or other mitigation) is, it is appropriate to the system. The strategy for arguing this is that a risk based process has been followed to make the patching decision.

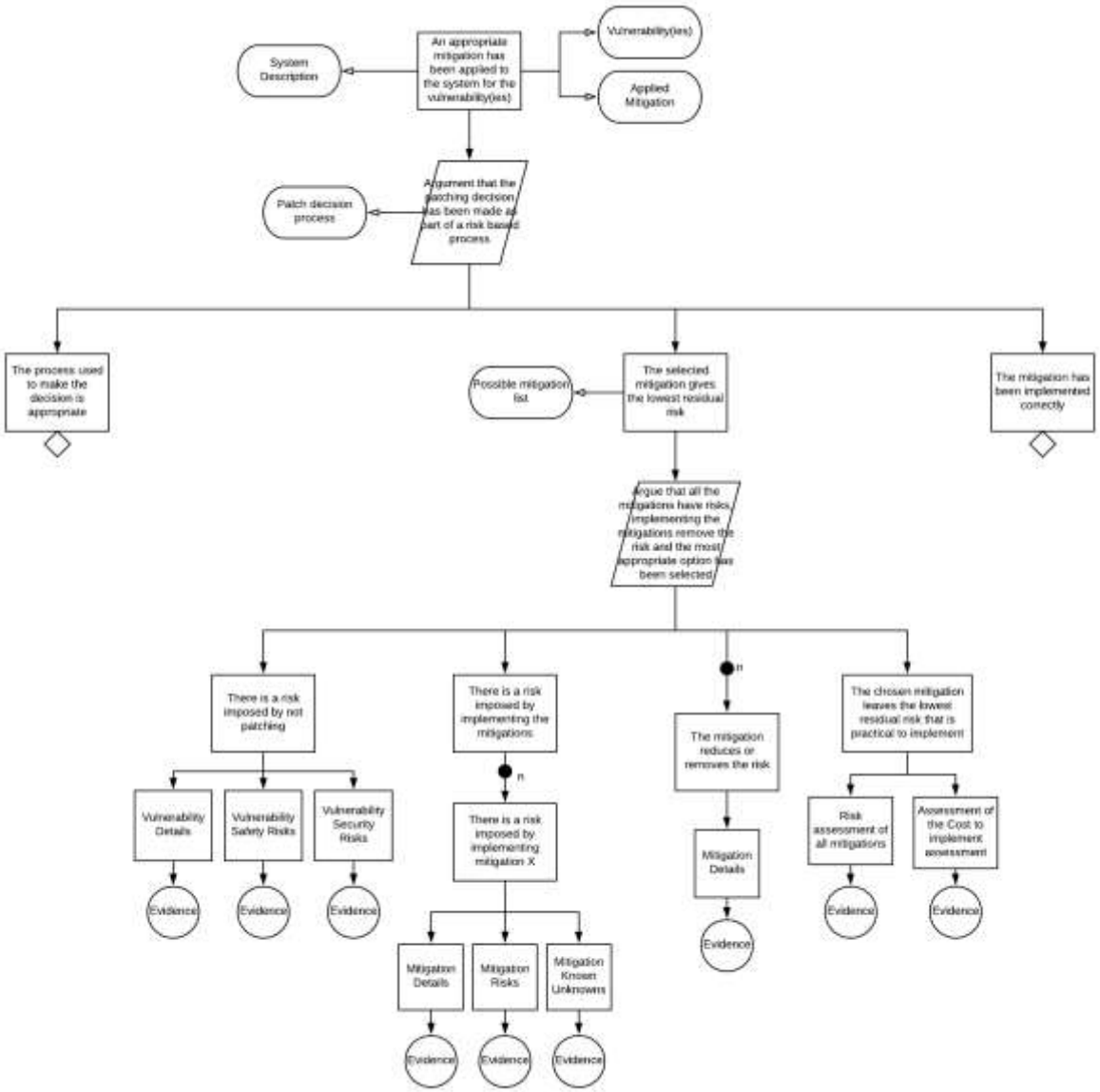


Figure 4 - GSN Argument for the Patching Decision

The argument is based around that a risk-based decision has been made comprising of 3 legs;

1. The process used to make the decision is appropriate
2. The process was used correctly, and an appropriate decision was made
3. The decision was implemented correctly

The first leg is left undeveloped to allow for any decision process to be used whilst making this justification, not just the one described in this document. This leg of the argument will need to be developed for the specific decision process being used with evidence showing why it is appropriate.

The argument that the mitigation has been implemented correctly is also left undeveloped as individual organisations will have their own requirements based on the processes they used.

The undeveloped legs help to provide confidence to the argument, the first providing confidence that the process used is appropriate. If it is possible to find issues with the process, this may allow the patching decision to be undermined. The correctly implemented leg of the justification provides confidence that the patch or mitigation has been applied correctly and that there have been no unwanted side effects observed in testing. The argument for this will need to appeal to the any document showing the implementation of the patch or mitigation and any testing that has been done on the SOI once that patch or mitigation has been applied.

The focus of the argument is that the decision made was appropriate. The argument around this focuses on the fact there is a risk of leaving the vulnerability in the system, the mitigations will introduce risk but reduce the risk of the vulnerability and finally that the most appropriate decision was made.

There is risk in not patching can be solved by showing that there is either a safety or security risk of the vulnerability. This evidence is generated from step 5.2,5.3, 5.5 and 5.8. Step 5.8 generates most of the evidence for this goal.

The risk of the mitigations demonstrates that there is a reason not just to apply the patch to the system. The evidence for this is generated from step 5.11 in the case of a patch and 5.12 for other mitigations.

Demonstrating that the mitigation would reduce the risk of the vulnerability shows that it is worth considering the mitigation and starts to gather support for the chosen mitigation. The evidence for this is generated from step 5.12.

Finally selecting the most appropriate mitigation relies on both the risk of the vulnerability being reduced, knowing the risks introduced by vulnerability as well as any costs, including new risks introduced by the mitigation.

Inputs

- Description of issue
- Assessment of the issue
- List of action carried out to the system
- Risk Assessments of potential and actual solutions
- Patching Justification GSN pattern

Outputs

Patching justification

Example

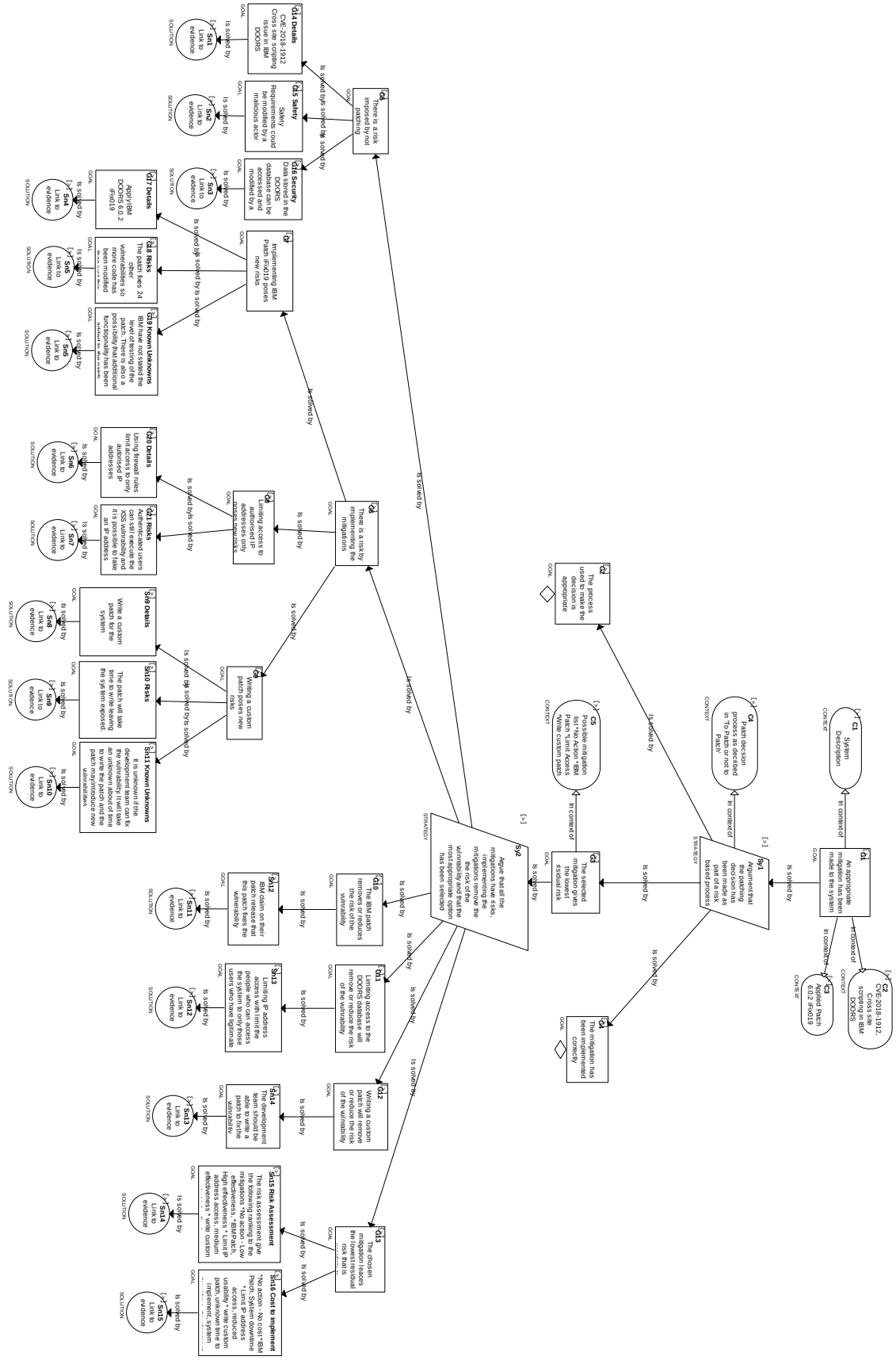


Figure 5 Justification for Patching Decision, expanded in Appendix C

5.20. Continue Operations

Once the patch or other mitigations have been put into place, operations should be able to be continued. It is advisable if mitigations or workarounds were put into place that these should be planned to be revisited to ensure that they continue to provide an effective mitigation to the vulnerability.

6. Evaluation

The output of the implementation phase will be evaluated against the two original goals;

G1. Allow a good decision to be made about applying patches to systems in a safety critical or safety related context
G2. Allow patching decisions to be justified

These goals were split into successes criterial in section 3.1, the evaluation will refer back to these success criteria.

6.1. Evaluation Method

To evaluate the goals fully, the process should be trialled on different vulnerabilities of different type and severity within multiple systems. Trailing the process on different vulnerabilities, works towards demonstrating that the process is capable of handling multiple types of vulnerabilities. It is possible that the process may not be suitable for assessing vulnerabilities where there is no patch or where the vulnerability does not have a major safety affect. By using multiple systems within the analysis, it is possible to show that the process is portable. For a full evaluation, the systems that the process is trailed on should be from different domains such as healthcare, defence, aviation and automotive. Each domain has its own processes, regulators and they can use different terminology so the process may not translate well to other domains.

All the people who exercise the process should be asked to fill in a questionnaire, this questionnaire should ascertain which domain that they work in, along with any specialisms (safety or security) that they have. The rest of the questionnaire should focus on the experience of running the process on their system. The specific details about the vulnerability and the system will demonstrate that the process can be applied to different vulnerabilities and the outcomes should demonstrate that the process allows for a decision to be made rather than being forced down a patch or no patch decision, this will be demonstrated by a range of outcomes of the process. The collated data from the questionnaires that filled in will provide feedback on how effective the process is against the originally stated goals.

6.2. Project Evaluation

Due to the constraints of completing this project within specific timescales it is not possible to run the evaluation as discussed in section 6.1. The process can only be tested within the defence domain and only for a small number of examples.

While this limitation does not affect the validity of the specific results, it does limit the applicability of the results to the domains where the assessment has been carried out. Other domains will need to trial the process separately to ensure that the process is valid for their domain before using the process as this will form part of the justification for patching decision made.

One of the concerns with security assessments is the sensitivity of the information and information about known vulnerabilities in a system is especially sensitive. This sensitivity is because a list of vulnerabilities on a system can become a how to hack the system guide. Due to this sensitivity the details about the vulnerabilities within the systems, especially within the defence community it is not possible to collect specific information on the vulnerabilities and the systems trialled on. Again this does not severely undermine the overall results of the trial, it just limits the ease that the process can be justified as appropriate as there will not be the level of detail in the trails to show that the process is applicable.

6.3. Questionnaire

6.3.1. Questionnaire Development

Whilst the limitations discussed in section 6.2 limit the need for the questionnaire a questionnaire will still be used to collate the data from those people who trial the process. The questionnaire needs to address the 2 main goals. The questionnaire also needs to ascertain the domain and specialisms that the person trialling the process has as this will inform the domains that the process is applicable to and whether the process can be used by non-specialists.

For every person trailing the patch decision process they will be asked to fill in the following form;

Q1. Describe the type of system that the process is being trialled on.
Q2. Describe the vulnerability that the process is being trialled on.
Q3. What was the outcome of the process?
Q4. What domain is the system that is being used for the trial form (aerospace, medical etc.)?
Q5. Do you have any safety or security expertise, if so which one?
Q6. Do you have a patching process in place already?
Q7. If yes what is the existing process?

Q8. If no are patches applied or not?
Q9. Do you feel the process lead you to a good decision, if not, why not?
Q10. Was the process easy to understand, if not what was more difficult?
Q11. Where in the process would more guidance be useful?
Q12. Did you require additional help to apply the process, if yes what help and what for?
Q13. Was any additional evidence required to generate the justification?
Q14. Is the justification compelling?
Q15. Have you shown your accreditor the patching justification?
Q16. Did the accreditor accept the justification, if not what was their response?

The table below provide a mapping between the questionnaire questions and the success criteria defined section 3.1.

Questionnaire Question	Success Criteria	Rationalisation
Q1.		This provides background information about the types of systems that the process has been trialled on, helping to gauge the applicability to different types of systems
Q2.		This provides background information about the types of vulnerabilities that the process has been trialled on, helping to gauge the applicability to different types of vulnerabilities
Q3.		This will allow any trends to be identified, does the process lead you down a specific patching decision etc.
Q4.	GS1.2 GS2.2	The domain that the person works in may indicate what if any specialism that they have
	GS1.3	If all a variety of domains can understand the terminology used there it is more likely that the terminology is unambiguous
Q5.	GS1.2 GS2.2	Self-declaration if the person performing the evaluation has any safety of security specialism
Q6.		This indicates if they are comparing it to something or if it is new to them
Q7.		This indicates if they are comparing it to something or if it is new to them
Q8.		This indicates if they are comparing it to something or if it is new to them
Q9.	GS1.1	If the process did not lead them to a good decision in their opinion, then there may be an issue with the process
Q10.	GS1.2	If the process was not easy to follow, it may be that the process is not useable by non-safety or security specialists. If the answer is no here, reference back to Q1 and Q2 will need to be made to understand if there is a specific issue.
Q11.	GS1.3	If more guidance is needed it may be because there is some ambiguity with the terminology used.
Q12.	GS1.3	If more guidance is needed it may be because there is some

		ambiguity with the terminology used.
Q13.	GS2.3	If more evidence was required to complete the justification than following the guidance produced, it would indicate that this goal has not been met
Q14.	GS2.1 GS2.2	If the justification is not compelling, then the justification would require more work and would potentially not be able to be followed by non-specialists
Q15.	GS2.1	If the justification has been shown to an accreditor, then there is additional evidence that the justification is acceptable.
Q16.	GS2.1	If the justification has been shown to an accreditor, then there is additional evidence that the justification is acceptable.

Table 8 Questionnaire Traceability to Success Criteria

6.4. Summary of the results

There are 4 results that have been submitted of the process being trialled;

R1. Medical Domain, Windows Vulnerability on a computer controlling an x-ray machine
Appendix D – Questionnaire Results, Medical Domain 1

R2. Defence Domain, Application Vulnerability on a computer used to write testing software
Appendix E - Questionnaire Results, Defence Domain 1

R3. Defence Domain, Windows Vulnerability on a non-networked computer used for health and usage monitoring of aircraft
Appendix F - Questionnaire Results, Defence Domain 2

R4. Defence Domain, Bug in Aircraft Code which could cause a mission computer shutdown.
Appendix G - Questionnaire Results, Defence Domain 3

As discussed in section 6.2 the range of results is limited. Most of them are from the defence domain and most are relating to Windows Vulnerabilities. The author believes that most of the issues are for Windows is partly due to sensitive nature of vulnerabilities on systems, but also because Windows vulnerabilities are made public through the databases such as the NVD [21]. The application software with the vulnerability in R2 is commercial software with entries on the NVD, whereas the application software in R4 is not commercial off the shelf software, this again may be why there was more willingness to apply some level of mitigation as the vulnerability is public in R2 whereas R4 the vulnerability is only known about by the relevant stakeholders.

	R1	**R2**	**R3**	**R4**
Domain	Medical	Aerospace / Defence	Aerospace / Defence	Aerospace / Defence
Vulnerability type	Windows	Application Software (COTS)	Windows	Application Software

Patching Decision	Patch	Apply Mitigation	Don't Patch	Don't Patch
Expertise	None	None	None	Safety
Good Decision	Yes	Yes	Yes	Yes
Different to norm	No	Yes	No	No
Hard areas of the process	Analysing the risks	Understanding how the vulnerabilities could be exploited	Including existing mitigations	None
More Guidance required	Step 5.5 and 5.12	Understanding how the vulnerabilities could be exploited	Including existing mitigations	Identifying mitigations
Justification Compelling	Yes	Yes	Yes.	Yes. Apart from undeveloped legs
Additional evidence required	No	No	Details about the existing mitigations were added	No
Accreditor	No	No	No	No

Table 9 Summary of evaluations

6.5. Goal 1 Evaluation

The main focus of goal 1 was to allow a good patching decision to be made. The process defined in section 5 allows for a patching decision to be made, goal 2 looks at the justification of the patching decision made, so evaluating the patching decision part of the process is left out of this section.

The people who trialled the process were able to make a patching decision for their system and vulnerability that they assessed. This shows that the process does allow for a patching decision to be made, fulfilling GS1.1. It should be noted that the only non IT system (Appendix G - Questionnaire Results, Defence Domain 3) that was evaluated against still made the decision to not apply any mitigation. There may be several reasons that this patching decision was made;

- **The risk of the vulnerability was sufficiently low**
  Without the full details of the system and the vulnerability it is very hard to make a conclusion about this. This is one of the biggest problems within the security field as organisations do not want to share details of vulnerabilities, especially if they remain unpatched.

- **The risk of the patch did not warrant the cost of developing and then testing and applying the patch to the fleet of aircraft**
  The cost of applying a patch to the Windows operating system is relatively low, this is because there are millions of active installations (200 million in 2016 [25]) so the cost of the update split between all the devices is low, also as most are connected to the Internet it is possible to remotely deliver and install the patches cutting down on the cost of this. The cost of applying a patch to unconnected systems as many safety critical systems are, or systems that do not have a remote patching capability may be much higher.
  This would then be an example of the business knowingly accepting a risk rather than the business unknowingly accepting the risk.
- **Risks of the vulnerability are being underestimated**
  As the process is a risk-based process and relies on the individual being able to conduct a risk assessment, it is possible that the risks of the vulnerability are being underestimated. If there have never been any cyber incidents, it is very easy to fall into the mind-set that there will never be any cyber incidents. It is very hard to accurately assess the likelihood that an adversary will attack your system and even harder to assess the likelihood that they will be successful. With the limited information available in this scenario and this being the only non IT based example it is impossible to draw full conclusions but additional work may be required to help remove the underestimation bias. An additional leg of the justification may also need to be added;
  'The people used to implement the process are appropriate'
  This leg would allow an argument to be made that people who have performed the risk assessment are able to assess the safety and security risk adequately.
- **There is a poor safety or security culture within the organisation**
  It is possible that there is a poor safety or security culture within the organisation that has led to laissez-faire attitude within the organisation to fixing either safety or security issues. This shows how important the organisations culture is when using the process define in section 5, it is possible that the justification may also need to appeal to the safety and security culture of the organisation.
- **The process does not work for this type of system**
  It is possible that the process does not work for a non IT based system. The only way to determine this is to perform more evaluations on such systems to gather more data. If the patching decision is still always not to apply any mitigation

Further evaluation will help to identify if the reason for this patching decision is an issue with the process or not.

The second sub goals to goal 1 (GS1.2) was the process should be able to be used by non-safety or security specialists. The process uses generally unambiguous language although

the evaluation does show that there is still some ambiguous language used, one of the biggest issues is the definition of risk, especially when being evaluated by either a safety or security specialist as safety risk is often very well defined and quantitative risk assessments can be performed, whereas security risk is very hard to put definitive numbers on so all the risk assessment tends to be qualitative. Further work needs to be applied to help the 2 forms of risk assessment to be compared. In the trial, those with no security background found it hard to evaluate the risk of someone exploiting the vulnerability. This is possibly as they underestimate the capability and motivation of attackers but the STUXNET virus shows that a determined malicious actor can infect a secured air gaped system [8]. There appeared to be no issues with the consideration of the safety risk as part of the process, this may be because safety is far more understood than security, or the safety aspects of the process have been defined in more detail. It is the author's opinion that it is more likely that the safety risks are better understood than security risks. To counteract this more guidance should be added to the security parts of the process if the process is further developed. The process itself suggests a number of areas where specialist knowledge may be needed to help to perform the risk assessment. Whilst these are suggested, the process can be followed without the specialist support but the specialist support does help to aid the confidence in the justification.

The final sub goal for goal 1 (GS1.3) was that the terminology used should be unambiguous. Whilst no one has complained about ambiguous terminology in the evaluation, due to the size of the trial, it is not really possible to draw a conclusion from this.

Goal 1 Success Criteria

GS1.1. Guidance to allow a patching decision to be made

The process proposed in section 5 provides guidance on how to make a patching decision for safety critical and safety related systems. All 4 people who trialled the process were able to make a patching decision.

GS1.2. The guidance should be usable by non-security and safety specialists

The process was trialled by 4 people, 3 of which did not have any specific system safety training and all 4 did not have any security training. They were all able to use the process without additional input. There are identified part of the process that suggest additional support from safety and security specialists, but this is to add credibility to the final patching decision.

GS1.3. Terminology in the guidance should be unambiguous

All those who evaluated the process understood the terminology used, however it is important to understand that this was a very small sample group and so only limited conclusions can be drawn from this.

6.6. Goal 2 Evaluation

The focus of goal 2 is to create a justification for a patching decision. The justification created is not specific to any process of coming to a patching decision, however the patching decision process identified in this report does fulfil all the evidence for the justification.

The GSN pattern in Figure 4 shows the GSN pattern suggested for arguing that the decision make is appropriate. As this decision process will be instantiated for every vulnerability, unless the vulnerabilities are being grouped together due to a patch solving many vulnerabilities, there is likely to be a large amount of duplication with the goal 'The process used to make the decision is appropriate'. The GSN Standard [20] has a modular extension, use of this extension within the GSN patter would allow for the argument over the process to be made the once and then referred to as an away goal reducing the repetition within the argument.

Another suggestion for the argument is that the top goal should be 'Known vulnerabilities have been dealt with appropriately' this would then allow for the goal 'The process used to make the decision is appropriate' to be made at the top level and each vulnerability argued about separately. The benefit of this approach is that all the justifications over the vulnerabilities are in one location together along with the evidence for all the vulnerabilities, this would assist with step 5.3 of the process. An issue with this approach is that every time a new vulnerability dealt with; the safety justification will need to be re-opened. Whilst the only change should be to add a new vulnerability to the argument structure, depending on the requirements of the customer or the accreditor the whole argument may need to be re-reviewed. Modular extension could help here again with each vulnerability being argued as a separate away goal, thus removing the need to re-review each vulnerability every time the argument is updated.

All those who evaluated the process were able to use the GSN pattern to create the justification. Unfortunately none of those who evaluated the process were able to share their justifications with an accreditor; this leaves the justification without any real external assessment. External review of the justification should be conducted to ensure the justification is compelling and cannot be easily undermined. While the justification does work for the no action patching decision, R3 did find that additional mitigations were required to fully justify the no action decision. This should be added to the GSN pattern to reinforce the justification making it more compelling.

Goal 2 Success Criteria

GS2.1. Provide a justification for a patching decision

The GSN pattern justification provided in Figure 4 allows for a justification to be made for a patching decision.

GS2.2. The justification should be able to be followed by a non-specialist

All of those who evaluated the process and justification were able to understand the justification. Only R4 had any safety expertise so the justification is suitable for non-specialists.

GS2.3. The evidence required for the justification should come from the output of goal 1

Only R3 required additional evidence to make the justification compelling and this was to support a no action patching decision. The additional evidence that R3 required had not been fully generated from the process and this was identified in their response to the questionnaire. For the other possible patching decisions (patch or mitigation) all the evidence was generated from the process from goal 1.

7. Conclusions

This report investigates the issues with making a patching decision on safety critical or safety related decisions. The report concludes that whilst there is no one correct decision for all systems, it is important to consider both patch and no action as well as alternative mitigations to protect against the vulnerability. A decision-making process is suggested that uses a risk based approach to decide on the most appropriate patching decision for the specific vulnerability. This process was evaluated in a limited sample size in two domains, medical and aerospace/defence, the outcome of this was a patching decision was made. One of the patching decisions was different to the organisation norm showing that the process does allow for an informed decision to be made. The evaluation does show there is a lack of understanding about how a vulnerability may be exploited by a malicious actor, this potentially shows either a lack of appreciation or it may also show a poor security culture within the organisation.

A GSN pattern, shown in Figure 4, is also presented as a method to justify a patching decision. The pattern does allow for a justification to be made and all the evidence can be generated from the patching decision process identified in the report for a patch and mitigation decision, there may be information missing when the decision is no action.

7.1. Future Work

As discussed in the evaluation the work has only been trialled in one instance in the medical domain and a small number of systems in the defence domain. To allow the applicability of this work to be validated across different types of systems and different domains more trials need to be run to allow a wider conclusion to be drawn. This

additional work will also help to inform the process leg of the justification generated during the process, section 5.19.

A process which is becoming more popular within the safety critical domain especially where security is concerned is System-Theoretic Accident Model and Processes (STAMP). STAMP is a theory created by N. Leveson and discussed in detail in Engineering a Safer World [26]. Systems Theoretic Process Analysis (STPA) is the process for assessing safety, it uses system theory and produces a control model of the system which is then assessed for control issues. This is very different to the way that many existing safety analysis techniques assess safety. As part of the decision process suggested within this document, a HAZOP analysis is suggested as a way to identify if the vulnerability affects safety. As more domain start to adopt STPA it may be more appropriate for them to assess the vulnerability by using STPA. Future work could be done to assess the compatibility of the decision process and STPA.

One of the potential issues with applying patches to some systems is the cost of developing and then applying the patch to systems. This needs to be considered at the design time of the system to make the most saving. Additional investigation on how this can be done both safely and securely needs to be performed so that during the design time of these systems appropriate measures can be put into place. This will help reduce the cost of applying the patches and so increasing the likelihood of applying a patch when performing a cost benefit analysis.

8. References

[1] C. Johnson, “Why We Cannot (Yet) Ensure the Cyber-Security of Safety-Critical Systems,” Safety-Critical Systems Club, 2016.

[2] R. Bloomfield, K. Netkachova and R. Stroud, “Security-Informed Safety: If It’s Not Secure, It’s Not Safe,” Springer, Berlin, 2013.

[3] G. McGraw, “Testing for Security During Development: Why We Should Scrap Penetrate-and-Patc,” *IEEE Aerospace and Electronic Systems Magazine,* pp. 13-15, 1998.

[4] R. R. Schell, “Information security: science, pseudoscience, and flying pigs,” in *Seventeenth Annual Computer Security Applications Conference*, 2001.

[5] J. Leyden, “‘Penetrate and patch’ e-business security is grim,” The Register, 20 February 2002. [Online]. Available: https://www.theregister.co.uk/2002/02/20/penetrate_and_patch_ebusiness_securit

y/. [Accessed 31 August 2018].

[6] M. Rouse, “Patch Tuesday,” Tech Target, July 2017. [Online]. Available: https://searchsecurity.techtarget.com/definition/Patch-Tuesday. [Accessed 27 August 2018].

[7] R. E. Johnson, “Survey of SCADA security challenges and potential attack vectors,” in *International Conference for Internet Technology and Secured Transactions*, London, 2010.

[8] N. Falliere, L. O. Murchu and E. Chien, “W32.Stuxnet Dossier,” Symantec, 2011.

[9] National Audit Office, “Investigation: WannaCry cyber attack and the NHS,” National Audit Office, London, 2017.

[10] T. Turner, “To Patch or Not to Patch,” University of York, York, 2018.

[11] S. Tom, D. Christiansen and D. Berrett, “Recommended Practice for Patch Management of Control Systems,” December 2008. [Online]. Available: https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/RP_Patch_Management_S508C.pdf. [Accessed 27 August 2018].

[12] Ministry of Defence, “Defence Standard 00-056 Part 2 Issue 5,” Ministry of Defence, 2017.

[13] K. O'Flaherty, “Hackers Used Malicious Update To Target 1 Million Asus Devices,” Forbes, 2019 03 2019. [Online]. Available: https://www.forbes.com/sites/kateoflahertyuk/2019/03/25/hackers-used-a-backdoor-to-infect-asus-users-find-out-whos-affected/#185ca1d1132d. [Accessed 15 06 2019].

[14] Microsoft, “Microsoft Lifecycle Policy FAQ,” Microsoft, 01 04 2019. [Online]. Available: https://support.microsoft.com/en-us/help/17140/lifecycle-faq-general-policy-questions. [Accessed 02 09 2019].

[15] K. Mackie, “Third-Party Windows XP Security Update Service Announced,” MCPMag, 21 08 2013. [Online]. Available: https://mcpmag.com/articles/2013/08/21/xp-security-update-service.aspx. [Accessed 02 09 209].

[16] W. Zamora, “How to tell if you’re infected with malware,” Malware Bytes, 24 09 2018. [Online]. Available: https://blog.malwarebytes.com/101/2016/05/how-to-tell-if-

] youre-infected-with-malware/. [Accessed 15 06 2019].

[17] Entec UK Ltd, “Assessing the safety of staffing arrangements for process operations in the chemical and allied industries,” HSE, Norwich, 2001.

[18] University of Oxford, *University of Oxford Risk and Security Analysis Course Notes MSc in Software Engineering/MSc in Software & Systems Security,* Oxford: University of Oxford, 2017.

[19] T. Kelly, “Arguing Safety - A Systematic Approach to Managing Safety Cases,” University of York, York, 1998.

[20] “GSN COMMUNITY STANDARD VERSION 2,” 2018. [Online]. Available: https://scsc.uk/r141B:1?t=1. [Accessed 30 08 2019].

[21] NIST, “National Vulnrability Database,” NIST, [Online]. Available: https://nvd.nist.gov/. [Accessed 26 07 2019].

[22] NCSC, “CISP Share,” NSCS, [Online]. Available: https://share.cisp.org.uk/my. [Accessed 26 07 2019].

[23] S. Lyngaas, “Microsoft's Meltdown patches introduced a whole new vulnerability,” CyberScoop, 28 03 2018. [Online]. Available: https://www.cyberscoop.com/microsoft-meltdown-patches-windows-7-memory-management/. [Accessed 13 05 2019].

[24] W. Smart, “Lessons learned review of the WannaCry Ransomware Cyber Attack,” Department of Health and Social Care, London, 2018.

[25] W. Williams, “Microsoft says that Windows 10 is now on 200 million devices and is its fastest growing OS ever,” betanews, 04 01 2016. [Online]. Available: https://betanews.com/2016/01/04/microsoft-windows-10-200-million-devices/. [Accessed 01 09 2019].

[26] N. G. Leveson, Engineering a Safer World, Massachusetts: MIT, 2011.

[27] Microsoft, “Microsoft Security Bulletin MS17-010 - Critical,” 14 March 2017. [Online]. Available: https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2017/ms17-010. [Accessed 19 July 2018].

[28] NCSC, “10 Steps: Secure Configuration,” NCSC, 08 August 2016. [Online]. Available: Develop and implement policies to update and patch systems: Implement policies to ensure that security patches are applied in an appropriate time frame, such a 14 days

for critical patches. Automated patch management and software update tools might be hel. [Accessed 30 July 2018].

[29] D. Shinder, “Patch or Not? Weighing the Risks of Immediate Updating,” TechGenix, 10 January 2015. [Online]. Available: http://techgenix.com/patch-or-not-weighing-risks-immediate-updating/. [Accessed 30 July 2018].

[30] A. Stephenson, D. Buttle and J. McDermid, “Change Management Strategies for Safety-Critical Software”.

[31] M. R. Endsley, “Situation Awareness Research and Design in Complex Systems,” [Online]. Available: https://www.abdn.ac.uk/iprc/documents/SA%20in%20Safety%20Critical%20Systems.pdf. [Accessed 31 July 2018].

[32] E. Byres, “The Air Gap: SCADA's Enduring Secuirty Myth,” *communications of the acm,* vol. 56, no. 8, pp. 29-31, 2013.

[33] P. Nohe, “What is an Air Gapped Computer?,” HashedOut, 13 March 2018. [Online]. Available: https://www.thesslstore.com/blog/air-gapped-computer/. [Accessed 2018 August 01].

[34] L. G. Paul, “Industrial Cybersecurity: Return of the Air Gap?,” Automation World, 17 January 2017. [Online]. Available: https://www.automationworld.com/article/technologies/security/industrial-cybersecurity-return-air-gap#Right. [Accessed 01 August 2018].

[35] D. Brumley, D. Song and J. Zheng, “Automatic Patch-Based Exploit Generation is Possible: Techniques and Implications,” Carnegie Mellon University, 2008.

[36] P. Baird, *The Relationship between Safety and Security,* FDA, 2017.

[37] Ø. Amundrud, T. Aven and R. Flage, “How the definition of security risk can be made compatible with safety definitions,” *Journal of Risk and Reliability,* vol. 231, no. 3, pp. 286-294, 2017.

[38] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, S. Mangard, P. Kocher, D. Genkin, Y. Yarom and M. Hamburg, “Meltdown,” 2018.

[39] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz and Y. Yarom, “Spectre Attacks: Exploiting

Speculative Execution," 2018.

[40] J. Crowe, "A Clear Guide to Meltdown and Spectre Patches," Barkly, Jan 2018. [Online]. Available: https://blog.barkly.com/meltdown-spectre-patches-list-windows-update-help#windows-updates. [Accessed 06 August 2018].

[41] L. Hautala, "Microsoft disables Spectre patch after bugs reported," C Net, 29 January 2018. [Online]. Available: https://webcache.googleusercontent.com/search?q=cache:HG9-u-WULzkJ:https://www.cnet.com/news/intel-microsoft-disables-spectre-chip-patch-bugs-reported-meltdown/+&cd=5&hl=en&ct=clnk&gl=uk. [Accessed 06 August 2018].

[42] Microsoft, "Update to disable mitigation against Spectre, Variant 2," Microsoft, January 2018. [Online]. Available: https://support.microsoft.com/en-us/help/4078130/update-to-disable-mitigation-against-spectre-variant-2. [Accessed 06 August 2018].

[43] Microsoft, "January 3, 2018—KB4056893 (OS Build 10240.17738)," Microsoft, 03 January 2018. [Online]. Available: https://support.microsoft.com/en-us/help/4056893/windows-10-update-kb4056893. [Accessed 06 August 2018].

[44] G. Cluely, "Thousands of compromised websites spreading malware via fake updates," Tripwire, 28 April 2018. [Online]. Available: https://www.tripwire.com/state-of-security/security-data-protection/cyber-security/thousands-compromised-websites-spreading-malware-via-fake-updates/. [Accessed 06 August 2018].

[45] SSH.COM, "Man-In-The-Middle Attack," SSH.COM, 02 October 2017. [Online]. Available: https://www.ssh.com/attack/man-in-the-middle. [Accessed 06 August 2018].

[46] InfoSecurity, "Target Hackers May Have Gotten In Through the Air Conditioner," InfoSecurity, 06 February 2014. [Online]. Available: https://www.infosecurity-magazine.com/news/target-hackers-may-have-gotten-in-through-the-air/. [Accessed 25 August 2018].

[47] RSSB, "Taking Safe Decisions 2.1," RSSB, 2014.

[48] J. Johansson and R. Grimes , "The Great Debate: Security by Obscurity," Microsoft, Jue 2008. [Online]. Available: https://technet.microsoft.com/en-us/library/2008.06.obscurity.aspx. [Accessed 25 August 2018].

[49] G. Bearfield, “Taking Safe Decisions in the GB Railway Industry,” in *Achieving Systems Safety: Proceedings of the Twentieth Safety-Critical Systems Symposium*, 2012.

[50] J. Viega and G. McGraw, Building Secure Software, Indianapolis: Pearson Education, 2002.

[51] W. A. Arbaugh, W. Fithen and J. McHugh, “Windows of Vulnerability: A Case Study,” *Computer,* vol. 33, pp. 52-59, 2000.

[52] Health and Safety Executive, “ALARP "at a glance",” Health and Safety Executive, [Online]. Available: http://www.hse.gov.uk/risk/theory/alarpglance.htm. [Accessed 25 August 2018].

[53] C. Borowski, “Patch Management Software,” Software Advice, [Online]. Available: https://www.softwareadvice.com/uk/patch-management/#buyers-guide. [Accessed 26 August 2018].

[54] The Government of the Hong Kong Special Administrative Region, “PATCH MANAGEMENT,” February 2008. [Online]. Available: https://www.infosec.gov.hk/english/technical/files/patch.pdf. [Accessed 26 August 2018].

[55] The National Institute for Occupational Safety and Health (NIOSH), “Hierarchy of Controls,” Centers for Disease Control and Preventrion, 11 May 2018. [Online]. Available: https://www.cdc.gov/niosh/topics/hierarchy/default.html. [Accessed 2018 August 27].

[56] A. Reed, “Manual workarounds: There’s no such thing as a “few seconds”,” Adrian Reed's Blog, 21 March 2014. [Online]. Available: http://www.adrianreed.co.uk/2014/03/21/manual-workarounds-theres-no-such-thing-as-a-few-seconds/. [Accessed 27 August 2018].

[57] W. E. Wong, J. R. Horgan, S. London and H. Agrawal, “A Study of Effective Regression Testing in Practice,” in *8th IEEE International Symposium on Software Reliability Engineering*, Albuquerque, 1997.

Appendix A – STRIDE

STRIDE is a threat modelling technique developed by Microsoft. STRIDE is a mnemonic for security threats;

- S – Spoofing
- T – Tampering
- R – Repudiation
- I – Information Disclosure
- D – Denial of Service
- E – Elevation of Privilege

Whilst Microsoft have definitions for each of these, they are very IT specific, the following definitions provide a more general definition which is easier to apply for Operational Technology. These definitions were adapted by BAE Systems.

- Spoofing – An external person or system influencing the system of interest (SOI) through an interface. The end result is that the SOI performs as intended to a false set of inputs.
- Tampering – The SOI is physically modified or the configuration is modified which results in the SOI not behaving as intended
- Repudiation / non-repudiation – The ability for user (system or person) to deny that they performed an action using the SOI.
- Information Disclosure – The SOI providing information to a person or system who is not authorised to have the information
- Denial of service – The SOI functions being blocked resulting in valid requests being dropped (unless this is the correct behaviour)
- Elevation of privilege – A user (system or person) accessing functions that they are not authorised to. This can include access to the system when there should be no access.

Appendix B – Example System Description

The example system is for a system development organisation developing safety critical and safety related systems. To allow their customers to keep up to date with the development of their systems they have a IBM DOORS database which contains all of the requirements, including safety requirements, for the systems under development. The DOORS database is used to link requirements to the design. Customers can access the database over the internet by using the DOORS interface and logging in with their username and password and they can only access their own requirement database. Customers can update their own requirements during the early phases of the development but they are locked for view only when development starts. If a customer then wishes to update a requirement they need to request a change using the DOORS interface.

8.1. Configuration

The system uses DOORS version 6.0.2 with no other patches applied.

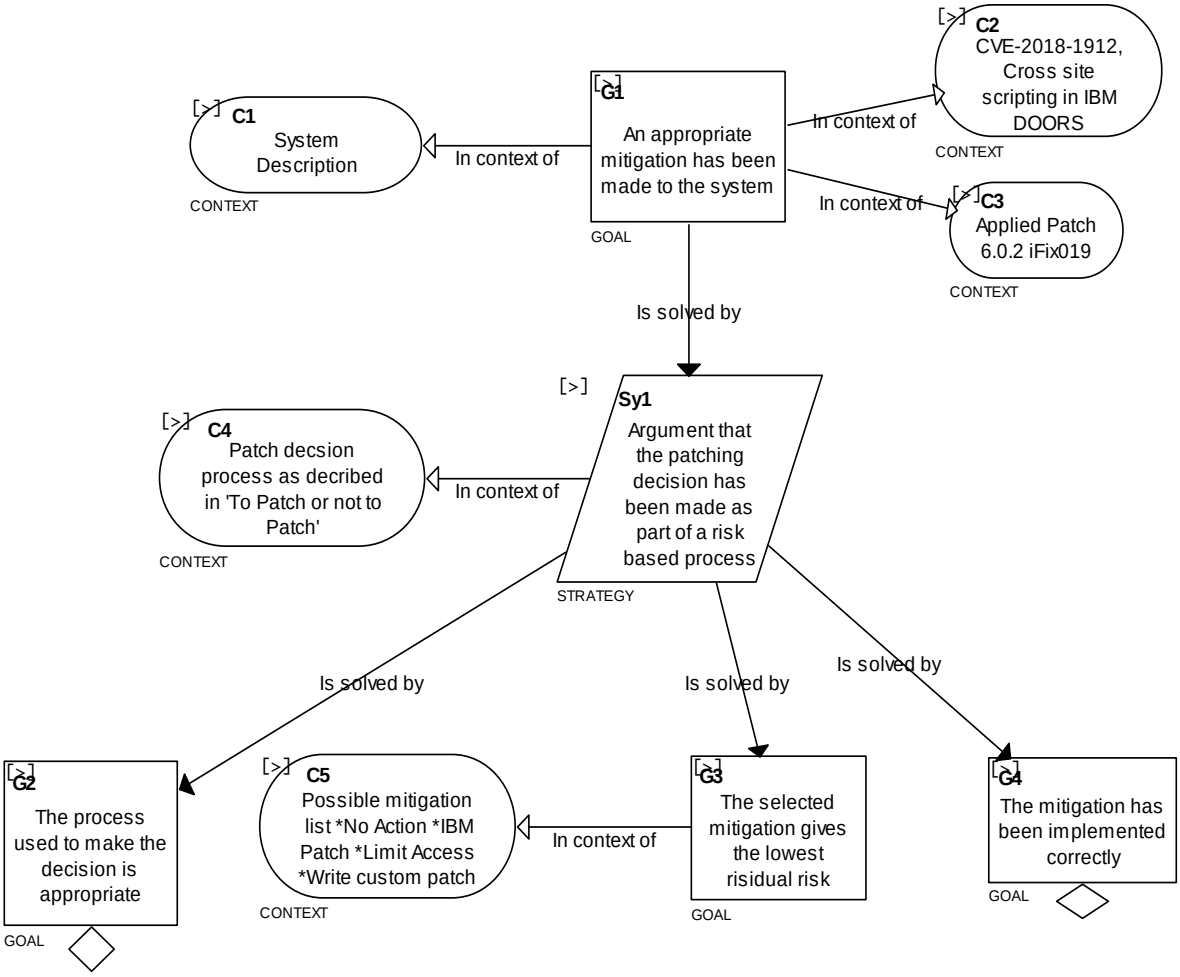
8.2. Safety Hazard log

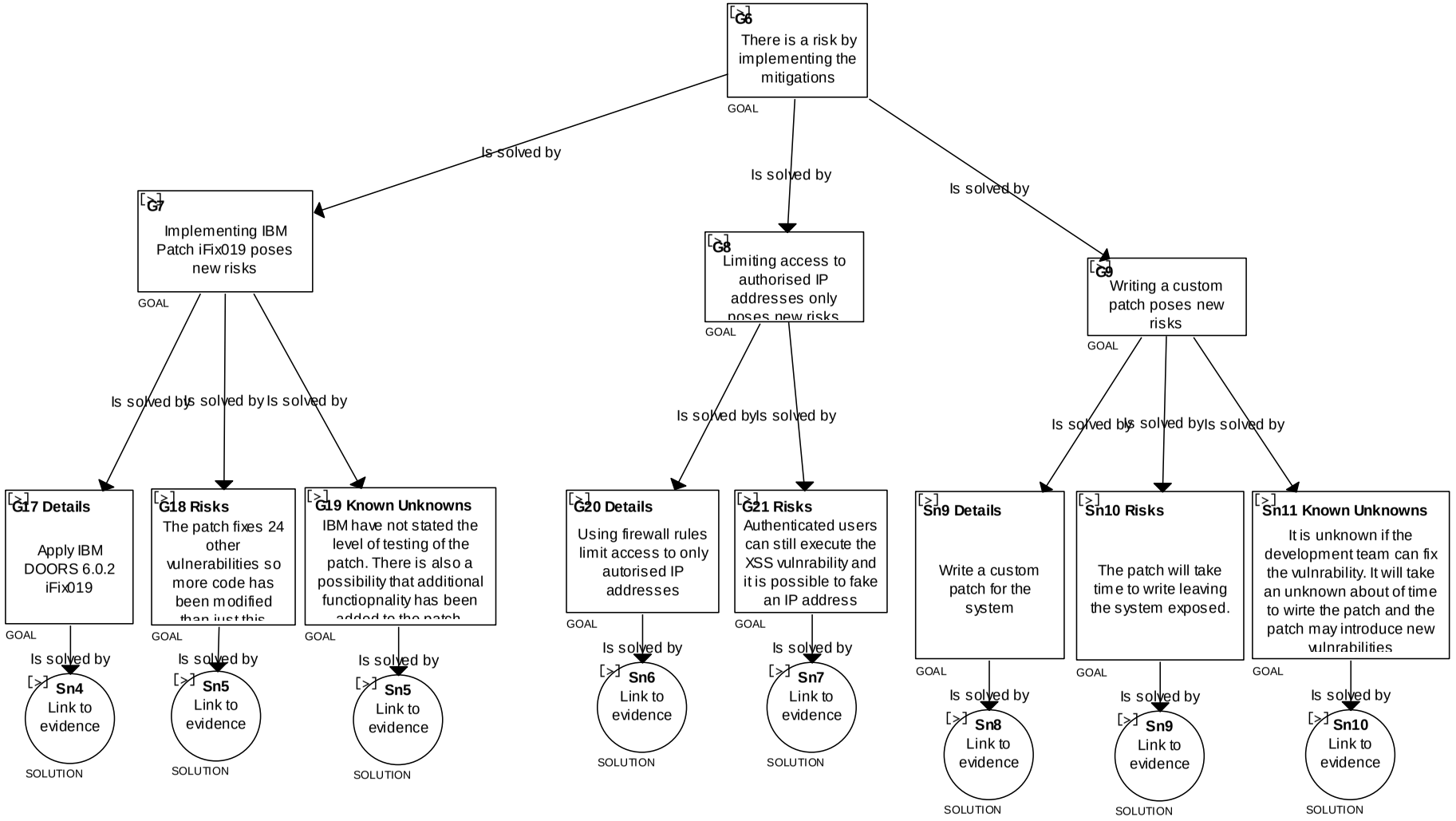
H1.	Corruption of the safety requirements

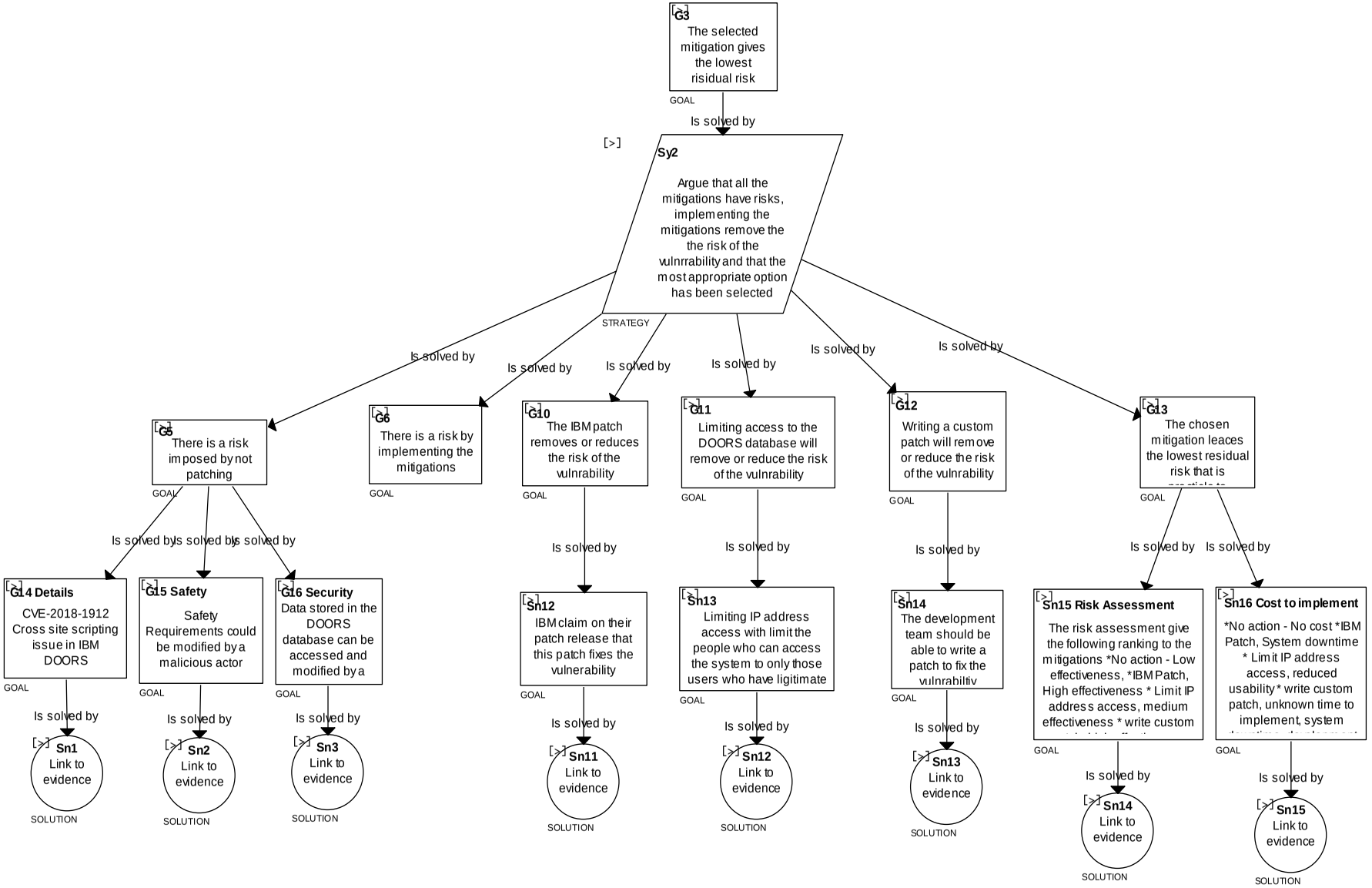
8.3. Security Goals

SG1.	Maintain confidentiality of customers data
SG2.	Block unauthorised traffic on the internal network

Appendix C – Example System Patching Decision Justification




G6
There is a risk by implementing the mitigations
GOAL
Is solved by
Is solved by
Is solved by
G7
Implementing IBM Patch iFix019 poses new risks
GOAL
G8
Limiting access to authorised IP addresses only poses new risks
GOAL
G9
Writing a custom patch poses new risks
GOAL
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
G17 Details
Apply IBM DOORS 6.0.2 iFix019
GOAL
G18 Risks
The patch fixes 24 other vulnerabilities so more code has been modified
GOAL
G19 Known Unknowns
IBM have not stated the level of testing of the patch. There is also a possibility that additional functiopnality has been
GOAL
G20 Details
Using firewall rules limit access to only autorised IP addresses
GOAL
G21 Risks
Authenticated users can still execute the XSS vulnrability and it is possible to fake an IP address
GOAL
Sn9 Details
Write a custom patch for the system
GOAL
Sn10 Risks
The patch will take time to write leaving the system exposed.
GOAL
Sn11 Known Unknowns
It is unknown if the development team can fix the vulnrability. It will take an unknown about of time to wirte the patch and the patch may introduce new vulnrabilities
GOAL
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Sn4
Link to evidence
SOLUTION
Sn5
Link to evidence
SOLUTION
Sn5
Link to evidence
SOLUTION
Sn6
Link to evidence
SOLUTION
Sn7
Link to evidence
SOLUTION
Sn8
Link to evidence
SOLUTION
Sn9
Link to evidence
SOLUTION
Sn10
Link to evidence
SOLUTION


G3
The selected mitigation gives the lowest risidual risk
GOAL
Is solved by
Sy2
Argue that all the mitigations have risks, implementing the mitigations remove the the risk of the vulnrrability and that the most appropriate option has been selected
STRATEGY
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
Is solved by
G5
There is a risk imposed by not patching
GOAL
G6
There is a risk by implementing the mitigations
GOAL
G10
The IBM patch removes or reduces the risk of the vulnrability
GOAL
G11
Limiting access to the DOORS database will remove or reduce the risk of the vulnrability
GOAL
G12
Writing a custom patch will remove or reduce the risk of the vulnrability
GOAL
G13
The chosen mitigation leaces the lowest residual risk that is
GOAL
Is solved by
Is solved by
Is solved by
G14 Details
CVE-2018-1912 Cross site scripting issue in IBM DOORS
GOAL
G15 Safety
Safety Requirements could be modified by a malicious actor
GOAL
G16 Security
Data stored in the DOORS database can be accessed and modified by a
GOAL
Is solved by
Is solved by
Is solved by
Sn1
Link to evidence
SOLUTION
Sn2
Link to evidence
SOLUTION
Sn3
Link to evidence
SOLUTION
Is solved by
Is solved by
Sn12
IBM claim on their patch release that this patch fixes the vulnerability
GOAL
Sn13
Limiting IP address access with limit the people who can access the system to only those users who have ligitimate
GOAL
Sn14
The development team should be able to write a patch to fix the vulnrabiltiy
GOAL
Sn15 Risk Assessment
The risk assessment give the following ranking to the mitigations *No action - Low effectiveness, *IBM Patch, High effectiveness * Limit IP address access, medium effectiveness * write custom
GOAL
Sn16 Cost to implement
*No action - No cost *IBM Patch, System downtime * Limit IP address access, reduced usability * write custom patch, unknown time to implement, system
GOAL
Is solved by
Is solved by
Is solved by
Is solved by
Sn11
Link to evidence
SOLUTION
Sn12
Link to evidence
SOLUTION
Sn13
Link to evidence
SOLUTION
Is solved by
Is solved by
Sn14
Link to evidence
SOLUTION
Sn15
Link to evidence
SOLUTION

Appendix D – Questionnaire Results, Medical Domain 1

Q1. Describe the type of system that the process is being trialled on.

Windows computer controlling an radiographic machine in a dental setting

Q2. Describe the vulnerability that the process is being trialled on.

Windows services vulnerability

Q3. What was the outcome of the process?

Patch

Q4. What domain is the system that is being used for the trial form (aerospace, medical etc.)?

Medical

Q5. Do you have any safety or security expertise, if so which one?

IR(ME)R qualified to operate the radiographic machine, however no security expertise or system safety qualifications.

Q6. Do you have a patching process in place already?

On this system no patches to Windows are installed, however, the radiographic software provider update their software as and when issues are identified.

Q7. If yes what is the existing process?

See question 6

Q8. If no are patches applied or not?

See question 6

Q9. Do you feel the process lead you to a good decision, if not, why not?

Yes

Q10. Was the process easy to understand, if not what was more difficult?

The process was easy to understand. Correctly analysing the risks was more difficult, although if the process was followed again this would become easier.

Q11. Where in the process would more guidance be useful?

Step 5.8 Analysing the risk introduced by the vulnerability, it was very easy to miss the potential malicious exploitation of the vulnerability

Step 5.12, assessing the options, it is easier to do nothing and ignore the risk introduced by the vulnerability

Q12. Did you require additional help to apply the process, if yes what help and what for?

No

Q13. Was any additional evidence required to generate the justification?

No

Q14. Is the justification compelling?

Yes

Q15. Have you shown your accreditor the patching justification?

No

Q16. Did the accreditor accept the justification, if not what was their response?

N/A

Appendix E - Questionnaire Results, Defence Domain 1

Q1. Describe the type of system that the process is being trialled on.

Testing system, used to run tests on mission computer software running on Windows

Q2. Describe the vulnerability that the process is being trialled on.

A vulnerability within LabVIEW software that allows a malformed Virtual Instrument (VI) file to cause a null write which could result in a code execution.

Q3. What was the outcome of the process?

Mitigation. New rule of not loading VI's onto the machine if the source is not known and trusted

Q4. What domain is the system that is being used for the trial form (aerospace, medical etc.)?

Aerospace / Defence

Q5. Do you have any safety or security expertise, if so which one?

None

Q6. Do you have a patching process in place already?

No

Q7. If yes what is the existing process?

N/A

Q8. If no are patches applied or not?

No

Q9. Do you feel the process lead you to a good decision, if not, why not?

Yes

Q10. Was the process easy to understand, if not what was more difficult?

Understanding how the vulnerability could be exploited

Q11. Where in the process would more guidance be useful?

Understanding how the vulnerability could be exploited

Q12. Did you require additional help to apply the process, if yes what help and what for?

No

Q13. Was any additional evidence required to generate the justification?

No

Q14. Is the justification compelling?

Yes

Q15. Have you shown your accreditor the patching justification?

No

Q16. Did the accreditor accept the justification, if not what was their response?

N/A

Appendix F - Questionnaire Results, Defence Domain 2

Q1. Describe the type of system that the process is being trialled on.

Stand-alone Windows computer used to process health and usage monitoring from an aircraft. The data is transferred using custom hardware from the aircraft to the computer.

Q2. Describe the vulnerability that the process is being trialled on.

Major windows vulnerability that has ransomware that will target it (Wannacry)

Q3. What was the outcome of the process?

Don't patch

Q4. What domain is the system that is being used for the trial form (aerospace, medical etc.)?

Aerospace / defence

Q5. Do you have any safety or security expertise, if so which one?

None

Q6. Do you have a patching process in place already?

No

Q7. If yes what is the existing process?

N/A

Q8. If no are patches applied or not?

No patches are applied

Q9. Do you feel the process lead you to a good decision, if not, why not?

Yes

Q10. Was the process easy to understand, if not what was more difficult?

The process was easy to follow, but it was hard to understanding how to model the existing mitigations with reduced the likelihood of the vulnerability being exploited.

Q11. Where in the process would more guidance be useful?

Modelling existing mitigations

Q12. Did you require additional help to apply the process, if yes what help and what for?

No

Q13. Was any additional evidence required to generate the justification?

Existing mitigations needed to be added.

Q14. Is the justification compelling?

Mostly although as the argument for not patching relies on the existing mitigations this had to be added into the argument

Q15. Have you shown your accreditor the patching justification?

No

Q16. Did the accreditor accept the justification, if not what was their response?

N/A

Appendix G - Questionnaire Results, Defence Domain 3

Q1. Describe the type of system that the process is being trialled on.

Training aircraft mission system

Q2. Describe the vulnerability that the process is being trialled on.

Bug in the code which could allow an attacker to send a malformed command to the aircraft through a wireless interface which would cause the mission computer to shutdown mid flight

Q3. What was the outcome of the process?

Fix the bug at next update

Q4. What domain is the system that is being used for the trial form (aerospace, medical etc.)?

Aerospace, defence

Q5. Do you have any safety or security expertise, if so which one?

Safety

Q6. Do you have a patching process in place already?

No

Q7. If yes what is the existing process?

N/A

Q8. If no are patches applied or not?

Patches are not applied

Q9. Do you feel the process lead you to a good decision, if not, why not?

Yes

Q10. Was the process easy to understand, if not what was more difficult?

Yes

Q11. Where in the process would more guidance be useful?

Identifying mitigations, could an example list of mitigations be included?

Q12. Did you require additional help to apply the process, if yes what help and what for?

No

Q13. Was any additional evidence required to generate the justification?

No

Q14. Is the justification compelling?

The process leg of the argument is not completed and as the patching decision was to wait until the next update cycle most of the pattern could not be filled in

Q15. Have you shown your accreditor the patching justification?

No

Q16. Did the accreditor accept the justification, if not what was their response?

N/A