Patch Management in ISO/IEC15408 & ISO/IEC18045

Request for a new study period

Javier Tallón

Version: 1.1
Date: 03/09/2019

Index

1 EXECUTIVE SUMMARY

This document tries to serve as a base for the request of the reopening of the study period regarding patch management in ISO/IEC15408.

In the second section, Introduction, we show a background of the problem and the motivation behind this request: final users of certified are forced to choose between security and certification when a vulnerability is made public.

Third section, Industry approaches to patch management, shows different approaches followed by cybersecurity industry standards and how they deal with patch management and which lessons could we learn from them.

Fourth section, Other initiatives, shows the outcome of the past study period and the in-progress work of the ISCI WG1.

Finally, fifth section, A new ISO/IEC15408 Patch Management proposal, propose the reopening of the SC27 study period and provides a basis to start working from the premise that it is better to advance towards continuous assurance step by step, and that the most critical issue to solve is the solution of public vulnerabilities. Based on the lessons learned from others standards and the previous work done, we need to attack the problem from several points:

- Evaluating the security of the development and distribution of the updates through the creation of a new ALC_FLR evaluation activity.
- Evaluating the TOE capability of applying security updates implementing specific security objectives.
- Improving the certification processes of the evaluation authorities to give priority to products where the two previous points have been evaluated.

2 INTRODUCTION

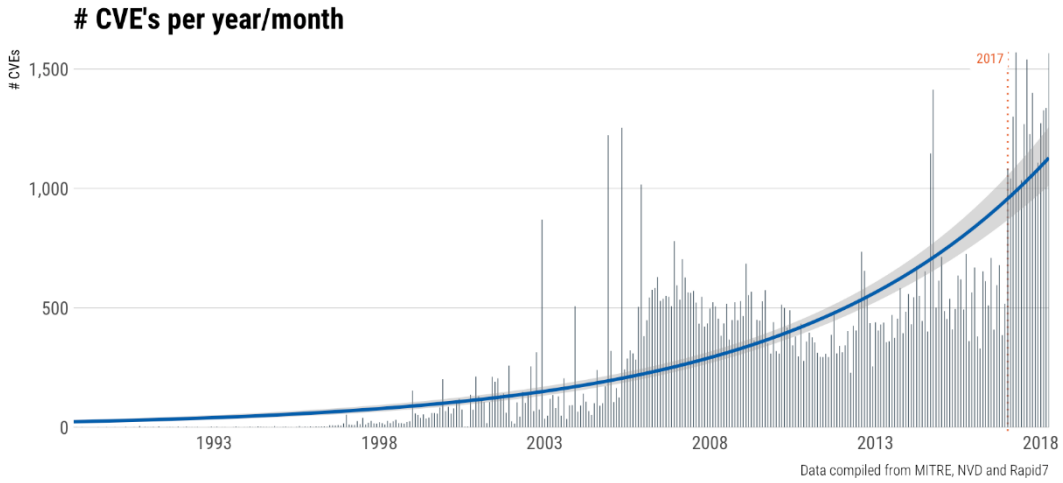
ISO SC 27 – WG3 is in charge of IT product security evaluation. Moreover, ISO/IEC15408 and ISO/IEC18045 better known as Common Criteria (CC) are the most recognized standard for security evaluations.

CTN 320 is in charge of monitoring and promoting the ISO SC 27 activity at the Spanish national level. I, Javier Tallon, am member of CTN320 and COO at jtsec (ITSEF).

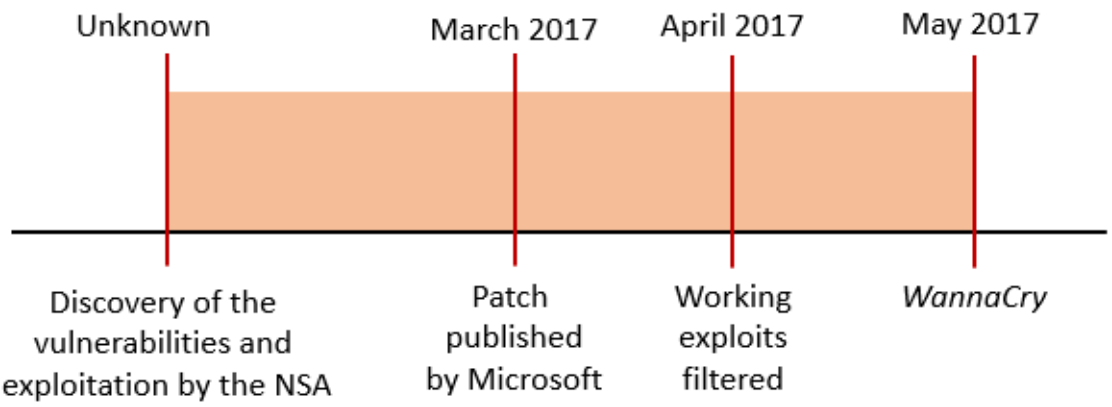
This document contains a contribution to SC27 WG3 regarding ISO/IEC15408 Patch Management.

2.1 BACKGROUNDS

The number of vulnerabilities has not stopped growing since the creation of the National Vulnerability Database (NVD). It is undeniable that every day that passes we are facing a more complex threat landscape where constant updating and patching is a must in a connected world.



The time needed by exploit writers to get a working exploit is shorter and shorter as malware developers professionalize. Evil hackers with monomaniacal intentions of a globe-disrupting nature have long dominated pop culture sensibilities. But when it comes to for-profit hacking, it's important to remember that cybercrime has been, and remains, predominantly a business-driven concern. As the ongoing ransomware scourge demonstrates, furthermore, the barriers to entry for cybercrime have never been lower and the easy profits to be made never been higher.



Meanwhile, it's no surprise that IT organizations in nearly all industries feel pressure to deliver value faster, get to market ahead of the competition, and continuously improve the customer experience. After all, market and customer expectations dictate constant innovation. The market leaders are the ones who get it and have aligned their organizations to deliver on innovation again and again.

For instance, Puppet Labs reported in its 2015 State of DevOps Report that high-performing IT organizations deploy 30 times more frequently with 200 times shorter lead times. According to an article in TechBeacon, Etsy deploys to its production servers 50 times a day with fewer disruptions than when the company used a waterfall approach. After moving to its own cloud, Amazon engineers deploy code every 11.7 seconds, on average—reducing both the number and duration of outages at the same time. Netflix engineers deploy code thousands of times per day.

Certified products patch management was a hot topic at ISO SC 27. During the last study period, several presentations were carried out about this topic but finally it was considered difficult to achieve something in a reasonable time. Patch management is one of the main concerns and drawbacks regarding certification. The time to market has been reduced dramatically in the last years and it is required to adapt certifications to take into account this. In our opinion is necessary to include in the scope of the certifications the patch management process to ensure that developers are meeting the expected requirements during the development of a patch.

When a developer creates a patch to its certified product because a vulnerability has been discovered, customers can't use it until certification granted, making the product vulnerable meanwhile.

It is almost 2020 and we are still making our users choose between certification and security, between using latest security version over the last certified one with bugs.

We need to adapt!

2.2 MOTIVATION

Despite the closing of the study period, the industry, security evaluation facilities and governments are still worried about how to handle patch management.

Having a patch management certified methodology will ensure providing a fast and efficient way for the industry to provide certified security updates in a timely and responsive manner, increasing its competitivity.

Maybe, having a good starting point would make working group 3 reconsider the reopening of the study period and luckily, we will finally reach to an agreement about how to handle the patch management issue. It is in favor of certification to value security.

In this document, we would like summarize the previous work to adapt the existent ISO 15408 and 18045 to include evaluation methodology that takes into account an effective and affordable patch management process.

3 INDUSTRY APPROACHES TO PATCH MANAGEMENT

3.1 THE CLASSIC COMMON CRITERIA APPROACH

When a product is certified under Common Criteria or ISO/IEC15408, only the product version stated in the Security Target is the one that holds the certificate.

This makes sense as this is the only version that has been tested by the evaluation lab and is the one for which the applicable certification body or evaluation authority can make an assertion of the testing done from the confidence in their accredited laboratories.

This way of work relies on three assumptions:

1. Every change in the code may introduce new vulnerabilities
2. The versions of the product are adequately handled by the manufacturers development process as shown from ALC_CMC.1
3. Evaluation authorities have an appropriate level of trust in the developer and in any developer-supplied evidence

If a certified product is updated, the certificate will no longer be valid. In general terms it is a good idea because as seen in (1), every change in the code may introduce new vulnerabilities, but is quite a weird situation when the update in the code is created to close a vulnerability that has been discovered after the end of the evaluation, because the not certified version is more secure than the certified one.

This put users in the crossroad of having to choose between the certified version and the patched version.

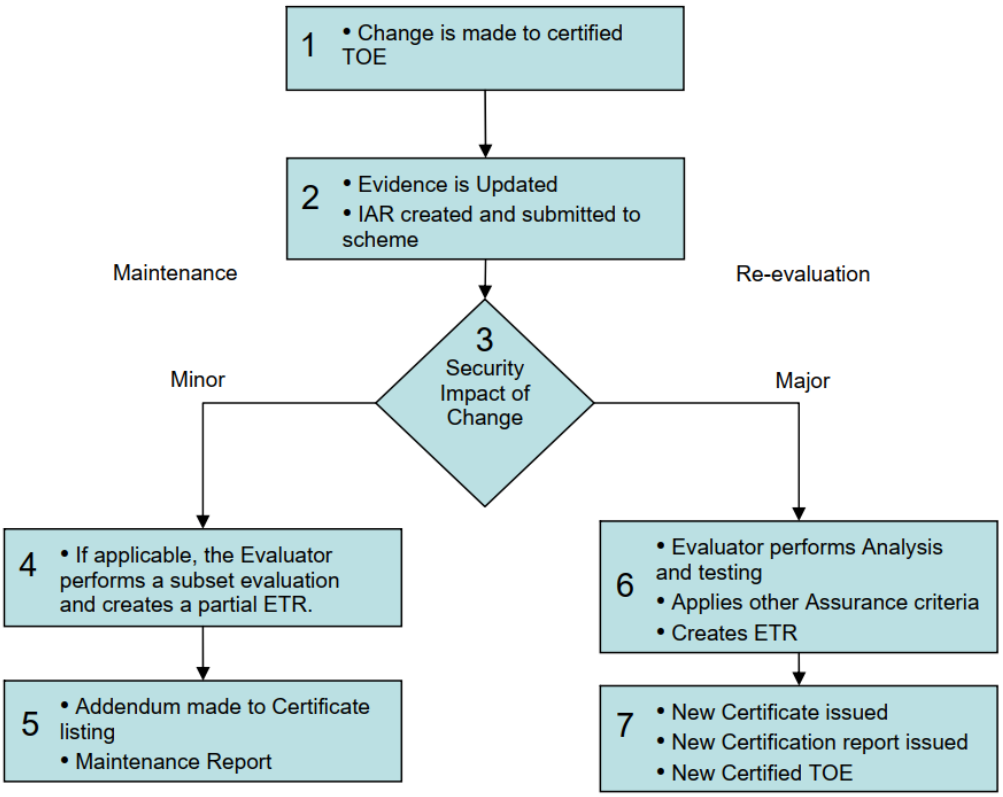
Trying to solve this situation the CCRA created [CCRA-AC] to define a mutually and recognizable approach to maintenance and reevaluation activities.

The document current version was created in 2004 and was later updated in 2008 to include the changes introduced in the version 3.1 of Common Criteria and introduce the concepts of subset evaluation and partial ETR. The last version is from 2012.

[CCRA-AC] states that as changes are made to a certified TOE or its environment, evaluation work previously performed might not be required to be repeated depending on the circumstances and ensures consistency among evaluation authorities in the characterisation of major and minor changes.

It is important to note that [CCRA-AC] is not intended to provide assurance in regard to the resistance of the TOE to new vulnerabilities or attack methods discovered since the date of the initial certificate. Such assurance can only be gained through re-evaluation. Maintenance only considers the effect of TOE changes on the assurance baseline; it does not consider an evolving threat environment.

An overview of the process is shown in the following figure:



Under this scheme, the developer generates an IAR (Impact Analysis Report) which records the analysis of the impact of changes to the certified TOE.

The evaluation authority will review the IAR to determine if changes are minor or major.

If changes are minor and there have been no changes in the development environment that can affect the TOE, and the TOE is still under the maintenance period, then, an evaluation lab may perform a subset evaluation of the affected assurance classes (if any) generating a partial ETR and the evaluation authority will generate a maintenance report. The lab involvement is required only if assurance classes are affected.

If changes are major, the modified TOE shall undergo evaluation reusing as much work as possible and generating a new certificate.

This way, if the change is deemed minor and there have been no modifications in the development environment, the time needed to include the changed TOE in the Maintenance Report is only subject to:

- The time needed by the developer to create the IAR
- The time needed by the evaluation authority to review the IAR and create the Maintenance Report
- Any bureaucracy delays

Let's say one month in the best case.

If the change is deemed major the time needed to generate the new certificate will be a percentage of the time required by the full evaluation thanks to the reuse of previous work. Let's say from one to six months.

3.1.1 LESSONS LEARNED

The most important aspect is therefore the characterization of the changes in minor/major. Luckily [CCRA-AC] offers a general guideline on the differences between major and minor. Sadly, it should be noted that a bug fix has no predictable extent of change to the certified TOE, nor a predictable effect upon the assurance of the certified TOE. Therefore, a "bug fix" might constitute either a major or minor change.

We can say that typically, a security bugfix will not fall in any of the three samples that [CCRA-AC] provide to characterize a major change:

- It is not a change to the set of claimed assurance requirements
- It is not a change to the set of claimed functional requirements
- It is not necessarily a set of minor changes that together have a major impact upon the security

The only hint that we have to characterize a bug fix in major or minor is if it will or not affect to the assurance evidence. For example, if a TOE has been certified to EAL1, a change to the source code and/or hardware schematics would not have an impact upon the assurance documentation.

Therefore, it shall be easy to have a maintenance report for low assurance levels than for high assurance (EAL4 or higher), as the bugfix will always affect source code. This however shall always be analyzed in detail as part of the IAR and the decision of classifying as major or minor is made by the applicable CB.

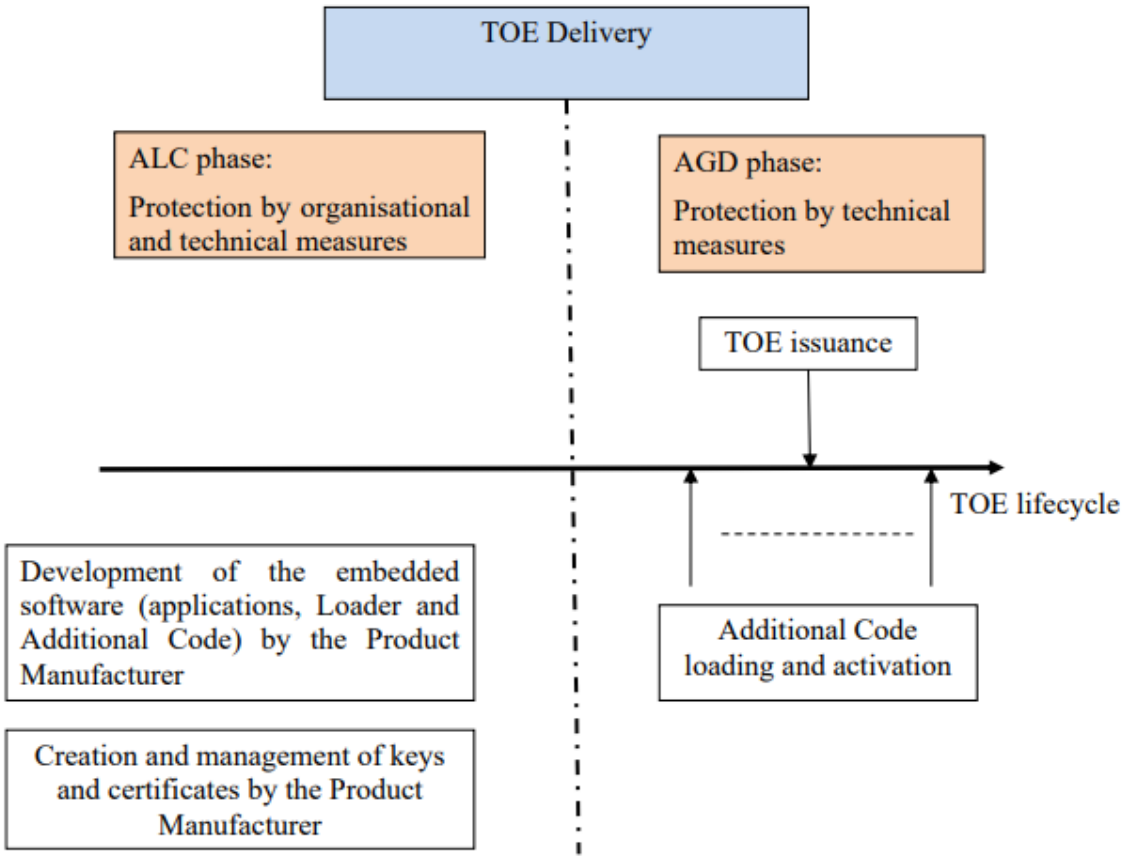
Anyway, the minimum time required to have a bugfix certified is deemed to be at least one month which will usually mean to make users perform a risk analysis to choose between using certified or secure (patched) products.

The timeframe provided by the Common Criteria classical approach is unquestionably insufficient to the nowadays evolution of the threat landscape.

3.2 JOINT INTERPRETATION LIBRARY – SECURITY REQUIREMENTS FOR POST-DELIVERY CODE LOADING

[JIL-CODELOAD] address the systematic evaluation of the code loading mechanism that is embedded in a growing number of products like "micro-electronic components embedding software". The purpose of the document is to define the concepts and the methodology applicable to the evaluation of a TOE embedding a code loading mechanism ("Loader") and the usage of this Loader as part of the assurance continuity process, it is applicable for the evaluation of products like "smart cards and similar devices" under the SOGIS scope.

The document does not deal with the assurance of the "Final TOE", that has to be successfully carried out in accordance with the assurance continuity procedure described in [CCRA-AC].



The TOE lifecycle is defined by two phases separated by the TOE Delivery: - a first phase called “ALC phase” corresponding to the product development phases covered by organizational and technical measures; - a second phase called “AGD phase” corresponding to the operational life of the product covered by guidance and technical measures.

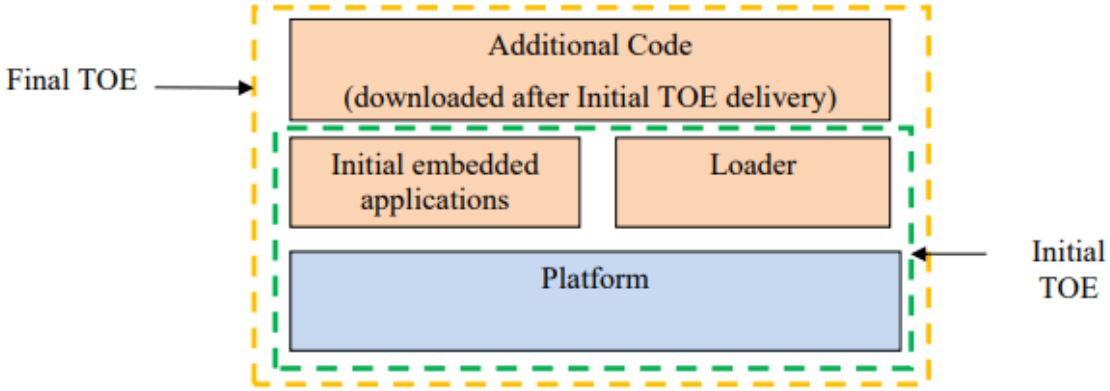
- ALC phase: Initial TOE and Additional codes are developed in a secure and audited environment as part of a CC evaluation. Keys and certificates have to be created by the Product Manufacturer and managed in a secure and audited environment. The Additional code is signed with a cryptographic key and the generated proof is linked to the Additional Code.
- TOE Delivery: The content and presentation requirements of ALC_DEL.1.1C already address this topic, but they must especially describe the protection measures of the proof associated to the Additional Codes and the protection measures of the cryptographic keys used to generate this proof. The measures described in the guidance will have to be audited.
- AGD phase: The proof verification functionality linked to the Additional Code is used by the Initial TOE or the Final TOE to check the integrity and authenticity of the Additional Code before its activation. The document provides three security objectives that address the implementation of the loading and activating code in the TOE: O.Secure_Load_ACode, O.Secure_AC_Activation and finally O.TOE_identification to allow users to uniquely identify Initial TOE and Additional Code(s) which are embedded in the Final TOE.

3.2.1 LESSONS LEARNED

Although specifically designed for the smartcard field, this document provides a useful terminology that may be used out of its initial scope. Moreover, its simple approach, putting some requirements

in the developer's assurance activities, and some requirements in the TOE functionality through the inclusion of implementation independent security objectives while reusing the existing Common Criteria assurance classes without reinventing the wheel, makes this document a very good starting point to achieve our objectives.

The main criticism that could be made is that the description of O.TOE_identification is too related to the smartcard field or maybe to any environment where apps-like code may be loaded, as is demonstrated in the following figure included in the document:



3.3 THE FIPS 140-2 APPROACH

The Federal Information Processing Standard (FIPS) Publication 140-2, (FIPS PUB 140-2), is a U.S. government computer security standard used to approve cryptographic modules.

The National Institute of Standards and Technology (NIST) issued the FIPS 140 Publication Series to coordinate the requirements and standards for cryptography modules that include both hardware and software components. Protection of a cryptographic module within a security system is necessary to maintain the confidentiality and integrity of the information protected by the module. This standard specifies the security requirements that will be satisfied by a cryptographic module. The standard provides for increasing qualitative levels of security intended to cover a wide range of potential applications and environments. The security requirements cover areas related to the secure design and implementation of a cryptographic module. These areas include cryptographic module specification; cryptographic module ports and interfaces; roles, services, and authentication; finite state model; physical security; operational environment; cryptographic key management; electromagnetic interference/electromagnetic compatibility (EMI/EMC); self-tests; design assurance; and mitigation of other attacks.

An updated version of a previously validated cryptographic module can be considered for a revalidation rather than a full validation depending on the extent of the modifications from the previously validated version of the module. (Note: the updated version may be, for example, a new version of an existing cryptographic module or a new model based on an existing model.)

FIPS 140-2, in its implementation guidance [FIPS-IG], offers a predefined multi-level model, breaking down a change into several submission scenarios or SUB for short, (1SUB, 1ASUB, 1BSUB, 2SUB, 3SUB, 3ASUB, 4SUB, 5SUB, etc.).

This model allows labs more "jurisdiction" to decide on the level of the change. Based on that decision the lab may proceed with the steps described in [FIPS-IG] which may involve the lab performing

applicable testing on the changed cryptographic module and submitting the test results (along with other required materials) to the CMVP for re-validation.

There are seven possible change Scenarios (1, 1A, 1B, 2, 3, 4, 5):

1SUB: administrative modifications or those that do not affect any FIPS 140-2 security relevant items. The vendor will provide the required information and the lab will determine additional testing as necessary. A NIST fee is not required.

1ASUB: there are no modifications to a module and the new module is a rebranding of an already validated Original Equipment Manufacturer (OEM) module. A $2000 NIST fee is applicable.

1BSUB: 1SUB scenario where the new lab is different from the original one. A $2000 NIST fee is applicable.

2SUB: Scenario 2 is for extending the module's sunset date when a module has not changed.

3SUB: Modifications are made to hardware, software or firmware components that affect some of the FIPS 140-2 security relevant items. An updated cryptographic module can be considered in this scenario if it is similar to the original module with only minor changes in the security policy and FSM, and less than 30% of the modules security relevant features. In addition to the tests performed against the affected assertions, the laboratory shall also perform the regression test suite of operational tests.

4SUB: Modifications are made only to the physical enclosure of the cryptographic module that provides its protection and involves no operational changes to the module. The CST laboratory shall fully test the physical security features of the new enclosure to ensure its compliance to the relevant requirements of the standard.

5SUB: If modifications are made to hardware, software, or firmware components that do not meet any of the above criteria, then the cryptographic module shall be considered a new module and shall undergo a full validation. A $8000 (Level 1) or $10000 (Levels 2-4) NIST fee is applicable.

Following ICMC 2016, the FIPS 140 Cryptographic Module User Forum (CMUF) decided to tackle the subject of improving the FIPS 140 security certification process creating the working group "Revalidation in Response to CVEs" with the following mission statement: "Updating FIPS 140 validation process to better integrate one or more CVE revalidation updates with a quick update, and create a dedicated entry point such as not to perform a full 3SUB."

The result of this effort was the creation of the new scenario 3ASUB.

3ASUB: A laboratory has been contracted to perform a revalidation for a module on which the vendor has made FIPS 140 security-relevant changes in response to one or more CVEs (Common Vulnerability and Exposure).

The purpose of the 3A revalidation scenario is to provide the vendor a means to quickly fix, test and revalidate a module that is subject to a security-relevant CVE, while at the same time providing assurance that the module still meets the FIPS 140-2 standard. If a CVE does not require security relevant changes to address it, then the vendor may pursue a Scenario 1 revalidation.

In this context, a security-relevant CVE is one that affects how the module meets the requirements of the FIPS 140-2 standard.

Please, note that having a CVE number basically means in other terms that the vulnerability has been publicly disclosed, which is the key point for being able to use this revalidation scenario.

3.3.1 LESSONS LEARNED

What are the advantages of Alternative Scenario 3A?

- The CMVP typically approves Alternative Scenario 3A submissions much faster than other 3SUBs.
- CVE can be mitigated by operational measures
- To improve speed, which is key in all this discussion, the "Revalidation in Response to CVEs" provided:
    - IAR Template for the developer.
        - Source code review against the original certified
    - Report Template for the lab.

Scenario 3A revalidations uses the joint forces of vendors, labs and the evaluation authority (NIST) to make security a priority while saving the guarantees of certification. Therefore, it is considered the example to follow, although it still can be improved. How? Here are some ideas:

- Trust the developer: traditionally certification has denied any kind of trust in the vendor. Everything is designed around suspicion and it is good because this gives confidence to final users that everything has been independently tested. However, an exception shall be made when security is at risk. What if we first approve, and then we test, putting penalties to vendor that are trying to cheat?
- Automatize?: fixing a vulnerability usually only involve changing a few bytes, e.g. modifying a call to an unsecure function to the secure version of the same function, adding a boundary checking, adding an extra check, … For those cases where the new patch only contains a few bytes we may automatically allow updating the certified version.

3.4 THE PCI-PTS APPROACH

The Payment Card Industry (PCI) Security Standards Council has established the PIN Transaction Security framework, to address the security evaluation and approval of payment security devices.

The Payment Card Industry PIN Transaction Security (PTS) Device Testing and Approval Program Guide provides information for vendors regarding the process of evaluation and approval by PCI SSC of payment security devices, and reflects an alignment of the participating card payment brands to a standard.

In this sense, PCI-PTS is similar to FIPS 140-2 because it establishes a methodology to evaluate and certificate the security of commercial off-the-shelf products, although in this specific case, the norm only deals with payment related devices.

The PCI SSC recognizes that vendors may need to make maintenance fixes to PTS validated devices that the vendor has already sold but still supports. In addition, vendors may wish to port updated versions of validated firmware that were assessed against newer security requirements to products for which the approval has expired. This may occur when customers wish to standardize their deployments against a given version of firmware and/or to add functionality to those devices.

For this matter, the PCI-PTS standard provides in its Program Guide [PCIPTS-PG] a guide for what they call Delta evaluations. Revisions to approved devices are termed “deltas.” Delta reviews involve the Recognized PTS Laboratory (or “PTS Lab”) assessing the changes

Vendors should contact one of the PTS Labs for guidance. PTS Labs will consult with PCI on an as-needed basis in advance of submitting a delta report to determine whether a set of changes is too great to be addressed under the delta process. The laboratories will determine whether the change impacts security. In all cases, changes that impact security require an assessment that must be presented in the delta report.

In general, any and all changes made to the firmware that runs on a previously approved PTS device may be considered in a single delta assessment except where the change is viewed as too pervasive, such as a change in the OS—e.g., changing from a proprietary to an open-based system. All changes made to PTS Approved Devices must be disclosed by the PTS vendor.

It is recommended PTS vendors submit a Change Analysis document to the PTS Lab.

The requirements of the standards are described in [PCIPTS-DTR]. Specifically, the requirement L1 deals with Change Control and Delta evaluation and requires that:

Change-control procedures are in place so that any intended change to the physical or functional capabilities of the POI causes a re-certification of the device under the impacted security requirements of this document. Immediate re-certification is not required for changes that purely rectify errors and faults in software in order to make it function as intended and do not otherwise remove, modify, or add functionality that impacts security. Approval of delta submissions is contingent on evidence of the ongoing change control and vulnerability management process.

This way, if the change is a bug fix, immediate re-certification is not required, prioritizing security, and giving the consumers the opportunity to update their devices while staying under the umbrella of the certification as long as any such changes are later bundled and ultimately passed to a PCI lab for evaluation.

3.4.1 LESSONS LEARNED

This way of working is possible because:

- The scheme trusts the developer and work together to achieve better security
- The change control procedures have been evaluated as part of the requirements of the standards.

3.5 THE EMVCO APPROACH

EMV is a payment method based upon a technical standard for smart payment cards and for payment terminals and automated teller machines that can accept them.

EMV cards are smart cards (also called chip cards or IC cards) that store their data on integrated circuits in addition to magnetic stripes (for backward compatibility). These include cards that must be physically inserted (or "dipped") into a reader, as well as contactless cards that can be read over a short distance using near-field communication (NFC) technology. Payment cards that comply with the EMV standard are often called Chip and PIN or Chip and Signature cards, depending on the authentication methods employed by the card issuer.

EMV originally stood for Europay, Mastercard, and Visa, the three companies that created the standard. The standard is now managed by EMVCo, a consortium with control split equally among Visa, Mastercard, JCB, American Express, China UnionPay, and Discover.

EMVCo facilitates worldwide interoperability and acceptance of secure payment transactions. Supported by dozens of banks, merchants, processors, vendors and other industry stakeholders, EMVCo manages and evolves the EMV® Specifications and related testing processes. This includes, but is not limited to, card and terminal evaluation, security evaluation, and management of interoperability issues.

EMVCo acts as the security certification entity for all approvals relating to the security of IC, Platform, and ICC products intended for use in payment cards issued by EMVCo members. EMVCo oversees and administers the security evaluation process and maintains security guidelines.

[EMV-SEP] describes the EMVCo Security Evaluation Process requirements and procedures for Integrated Circuit (IC), Platform (IC + OS), and Integrated Circuit Card (IC + OS + App) products. All product providers shall follow the process outlined in this document in order to obtain and maintain security evaluation certificates for their products. Such certification will allow product providers to sell ICC products to issuers of ICCs bearing the brand marks of American Express, Discover, JCB, MasterCard, UnionPay, or Visa.

The EMVCo Security Evaluation Process evaluates the security features of IC, Platform, and ICC products. It also takes into account the security of the design, development, and delivery processes. It is based on a complete set of published EMVCo documents (specifications, requirements, and security guidelines) made available to product providers and security evaluation laboratories for the development and security evaluation of their products.

Regarding changes to previously approved products, [EMV-SEP] states that any change to a previously approved product will require a Security Impact Analysis (SIA) which must be provided to, and approved by, the EMVCo Security Evaluation Secretariat. Based on the Security Impact Analysis, a delta evaluation may need to be performed before the EMVCo Compliance Certificate can be issued for a changed product.

The product provider sends the appropriate material (updated samples, guidance documentation, etc.) to the selected security evaluation laboratory, which runs a delta evaluation process. This process will focus on the changes in the product design or its documentation, and may lead to additional code review and testing activities.

In cases where the Security Impact Analysis concludes that the changes made to the product are minor and have no security impact (and therefore no additional testing is performed), a Fast Track report review process will be applied, as defined in SEWG Bulletin 10 [EMV-BL10].

[EMV-BL10] outlines the EMVCo security evaluation review process for product changes that do not have a security impact. This “Fast Track” review process allows products to be approved within less than two weeks.

The Lab must clearly indicate in the submitted email that the SIA concludes the changes have no security impact on the product and is therefore eligible for the Fast Track review.

The SEWG Secretariat will Fast Track review SIAs within one (1) working week upon reception date of the SIA. If the Secretariat successfully validates the Laboratory’s analysis, a Summary Report will then be submitted to SEWG for final approval. If approved by SEWG, an updated product certificate will be issued within the following working week.

3.5.1 LESSONS LEARNED

This short time can be achieved because:

Since the beginning of Q2 2013, EMVCo has outsourced security evaluation report reviews to an independent Secretariat with dedicated resources.

Fast Track acts like a priority queue.

4 OTHER INITIATIVES

As stated in the introduction, this is not the first time that someone tries to bring attention to patch management in ISO/IEC15408. In fact, Patch Management will bring great benefits to everyone, especially the end users and the industry.

Let's take a look to what other working groups are trying and what have already been done in the past:

4.1 ISO SC27 WG3 2018 STUDY PERIOD

The initial idea behind the Introduction of new assurance requirements in ISO/IEC 15408-3 and evaluation methodology in ISO/IEC 18045 covering patch management and deployment activities was to benefit of standardized patch management evaluation while adding some patch (or software update) related vocabulary to CC. This study period intended to create a set of predefined SFRs and SARs covering patch management functionality, while taking into account the particularities of different technologies.

A report ([SC27WG3-2018SPR]) was presented during the 57th SC 27/WG 3 meeting in Gjøvik, Norway, 30th September – 4th October 2018.

Very few contributions were received between meetings and few contributing people during meetings and confcalls (only 5 contributions in two calls for contribution) so it was called into question if patch management was a real-world problem and if WG3 members would really like to standardize.

Some WG3 members required explicitly to perform a process evaluation instead of product evaluation which is not in line with CC framework (ISO 15408) based on product evaluation.

They worked on a document to collect SFRs and SARs, definition of packages and they had good progress defining SFR packages but had trouble to identify and describe sufficient generic evaluation actions for assurance requirements.

Some Technical Domains have already solved the problem for their own situation like the Multi-Function Printer (MFP) Technical Domain, which uses yearly re-certification and therefore there is no time for patch certification.

4.1.1 LESSONS LEARNED

The conclusion of the report is that we have a real-world problem which needs to be solved and therefore work has to be continued. The WG3 produced ideas but at the moment they did not have enough experience so the work was suspended because the need for more experience on this topic between vendors, labs and CBs before standardize.

It seems that the scope of the working group was too ambitious, trying to contemplate all kind of situations and providing a lot of new CC concepts instead of trying to use what we already have. Creating new SFRs and SARs was a good idea but it looks like the work was way too wide (ADV_PMP, ALC_PDP, ALC_AFR, ALC_PYP, AGD_UPD, ADV_DIF, AVA_PVA…).

4.2 ISCI WG1

ISCI WG1 recently (June 2019), created a sub-group to build a process to rationalize the updates of certified products while keeping the product in question certified. The initial goal of the subgroup is to either find a more effective way to facilitate the defined Assurance Continuity process, or find an alternative process, and define it in detail, allowing changes to be deployed in the field within a short period of time.

To achieve the above described goal, the subgroup will do the following:

1. Review how the Assurance continuity process is handled by other CBs and Schemes e.g., the fast track process in EMVCo (which has been already described as part of this document).
2. The subgroup will then work on various models for updating a certified TOE, and propose the following:
   - Methodology document containing a process description about the secure update of the TOE while also maintaining the certificate.
   - The subgroup will also look at how this process could be handled within the existing Assurance Continuity process. The document will contain an according interpretation of the Assurance continuity process.

The work of this WG is clearly aligned with the intended proposal of this document.

5 A NEW ISO/IEC15408 PATCH MANAGEMENT PROPOSAL

It's time to make a proposal from the case studied to implement a patch management methodology that allow us to be able to deploy security updates in a secure, fast and efficient way.

Throughout this document we have seen how different schemes are handling the patch management process. Common Criteria provides a powerful versatility that could allow us creating a patch management methodology that surpasses all the others standards in the security and flexibility provided so we shall use this as an advantage, but as we may have learned during the creation of this document this problem must be attacked from several perspectives:

- Security in the development and distribution of updates
- Evaluation of the TOE capability to securely apply updates
- Improvements in the certification processes to handle security updates more efficiently

5.1 SECURITY IN THE DEVELOPMENT AND DISTRIBUTION OF UPDATES

Common Criteria already establishes a good amount of evaluation work to test the security of the development environment and processes, ensuring that a change control methodology is already been applied with a level of effort commensurate with the evaluation level.

This relationship between the evaluation assurance level and the effort put in all the activities related to patch management shall be kept as is. It makes sense that the higher the EAL, the deeper the assurance that the patch management processes are being used in a secure way.

[CCPART3] already provides an assurance family to address flaw remediation. In its higher level, ALC_FLR.3 Systematic flaw remediation, the norm requires the developer to be able to act appropriately upon security flaw reports from TOE users, and to know to whom to send corrective fixes, TOE users need to understand how to submit security flaw reports to the developer, and how to register themselves with the developer so that they may receive these corrective fixes. Flaw remediation guidance from the developer to the TOE user ensures that TOE users are aware of this important information.

The family however does not deal with the integrity and authenticity of the updates so we propose creating the following component:

ALC_FLR.4 Systematic flaw remediation with secure update procedures

Dependencies: No dependencies

Objectives:

In order for the developer to be able to act appropriately upon security flaw reports from TOE users, and to know to whom to send corrective fixes, TOE users need to understand how to submit security flaw reports to the developer, and how to register themselves with the developer so that they may receive these corrective fixes. Flaw remediation guidance from the developer to the TOE user ensures that TOE users are aware of this important information **including instructions on how to securely apply "Additional Code" patches. "Additional Code" patches signing and distribution procedures shall be handled by the developer.**

Developer action elements:

ALC_FLR.4.1D The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.4.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.4.3D The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.4.4D The developer shall document and provide "Additional Code" patch creation procedures to TOE developers.

ALC_FLR.4.5D The cryptographic keys and proof generation related to the Additional Code management will be carried out in a secure and audited environment.

ALC_FLR.4.6D The developer shall provide evidence that the same assurance level provided for ADV, AGD, ATE and ALC_DEL is also provided for the "Additional Code".

Content and presentation elements:

ALC_FLR.4.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.4.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.4.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.4.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.4.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.4.6C The flaw remediation procedures shall include a procedure requiring timely response and the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

ALC_FLR.4.7C "Additional Code" patch creation procedures shall demonstrate how the Additional code is signed with a cryptographic key and the generated proof is linked to the Additional Code.

ALC_FLR.4.8C "Additional Code" patch creation procedures shall demonstrate how the cryptographic keys are handled so they have sufficient quality.

ALC_FLR.4.9C "Additional Code" patch creation procedures shall demonstrate how the process of key generation is appropriately secured to ensure the confidentiality, authenticity and integrity of the cryptographic key.

ALC_FLR.4.10C "Additional Code" patch creation procedures shall demonstrate how the process of proof generation related to the Additional Code is appropriately secured to ensure the authenticity and integrity of the proof.

ALC_FLR.4.11C "Additional Code" patch creation procedures shall include the creation of a detailed Impact of Changes following evaluation authority templates.

ALC_FLR.4.12C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.4.13C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.4.14C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.4.15C The flaw remediation guidance shall describe a means by which TOE users may register with the developer, to be eligible to receive security flaw reports and corrections.

ALC_FLR.4.16C The flaw remediation guidance shall identify the specific points of contact for all reports and enquiries about security issues involving the TOE.

ALC_FLR.4.17C The flaw remediation guidance shall provide detailed instructions for users on how to check the availability of new "Additional Code" patches and how to apply them.

Evaluator action elements:

ALC_FLR.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.4.2E The evaluator shall exercise the "Additional Code" update process to verify that it is correctly handled including the documentation of the Impact of Changes.

ALC_FLR.4.3E The evaluator shall verify that a commensurate level of assurance is provided for the design, development, testing and delivery of "Additional Code".

As in [JIL-CODELOAD], we have not dealt with the delivery of the "Additional Code" as the assurance component of the family ALC_DEL (delivery procedure) already deals with the TOE delivery or parts of it to the user, but all the guidance describing the delivery procedures shall be taken into account, and must especially describe the protection measures of the proof associated to the Additional Codes.

5.2 EVALUATION OF THE TOE CAPABILITY TO SECURELY APPLY UPDATES

The first part of the equation is solved with the creation of a process on the developer side to securely create and distribute "Additional Code" patches to final users.

The second part of the equation requires the TOE to implement functionality to verify the integrity and authenticity of these updates.

It may be tempting to create Security Functional Requirements (SFRs) specifically for this task, but this probably will be against the flexibility provided by the norm and may left some TOE types out of the applicability of this methodology.

This way, as in [JIL-CODELOAD], we have decided to only use security objectives which gives the developer the flexibility to decide how to implement them:

O.Secure_Load_ACode Secure loading of the Additional Code

The Loader of the Initial TOE shall check an evidence of authenticity and integrity of the loaded Additional Code. The Loader enforces that only the allowed version of the Additional Code can be loaded on the Initial TOE. The Loader shall forbid the loading of an Additional Code not intended to be assembled with the Initial TOE. During the Load Phase of an Additional Code, the TOE shall remain secure.

O.Secure_AC_Activation Secure activation of the Additional Code

Activation of the Additional Code and update of the Identification Data shall be performed at the same time in an Atomic way. All the operations needed for the code to be able to operate as in the Final TOE shall be completed before activation. If the Atomic Activation is successful, then the resulting product is the Final TOE, otherwise (in case of interruption or incident which prevents the forming of the Final TOE such as tearing, integrity violation, error case…), the Initial TOE shall remain in its initial state or fail secure.

O.TOE_Identification Secure identification of the TOE by the user

The Identification Data identifies the Initial TOE and Additional Code. The TOE provides means to store Identification Data in its non-volatile memory and guarantees the integrity of these data. After Atomic Activation of the Additional Code, the Identification Data of the Final TOE allows identifications of Initial TOE and Additional Code. The user shall be able to uniquely identify Initial TOE and Additional Code(s) which are embedded in the Final TOE.

5.3 IMPROVEMENTS IN THE CERTIFICATION PROCESSES

The above changes will be insufficient without being able to apply some kind of advantages in the certification process to those developers that have been demonstrably compliant with them.

This is a topic for great discussion and which will require international consensus (with all what this means), but it is a must if public schemas want to stay useful and up-to-date.

The following requirements describe how the patch management process shall be handled by evaluation authorities. Currently it has been restricted to public vulnerabilities because this is without a doubt the main scenario where the applicability of this process is more justified.

- REQ.1: Evaluation authorities shall allow developers that implement O.Secure_Load_ACode, O.Secure_AC_Activation and O.TOE_Identification and ALC_FLR.4 as described in this document to access the Patch Management Process.
- REQ.2: The Patch Management Process shall only be used to patch public vulnerabilities.
- REQ.3: Evaluation authorities shall provide templates to analyze the impact of changes in a patch
- REQ.4: Developers shall send evaluation authorities an analysis of the impact of the changes in a specific patch.
- REQ.5: Evaluation authorities shall create a priority queue for processing and reviewing the impact of changes in a patch.
- REQ.6: Evaluation authorities shall trust by default developers in order to harmonize security and certification. This way the impact of changes shall be accepted by default.
- REQ.7: Developers are required to evaluate the changes with a Security Evaluation Facility under the evaluation's authority certification body before 6 months.
  REQ.8: Evaluation authorities shall put penalties on cheaters and measures to find them.

6 CONCLUSIONS AND FUTURE WORK

We have provided a first insight in how to handle the patch management procedures, especially in vulnerability fixing scenarios, but there are still a lot of work to be done:

- We shall contrast with the international community if this document is accepted as a starting point and create a technical specification that contains an agreed set of security objectives that allows a secure patch management implementation.
- We shall agree a set of SARs to be included in the next versions of ISO/IEC15408 to enforce the evaluation of the patch management processes.
- We shall achieve consensus between certification bodies creating a technical specification that describes how to handle patch management improving their responsiveness.
- We shall test the applicability of this methodology in pilot evaluations.
- We shall discuss the applicability of this methodology to non-security patches.
- We shall walk towards scenarios where assurance continuity is a reality, certifying the patch management process and certifying the current version of the product in a yearly fashion.

7 REFERENCES

Reference	Document
[CCRA-AC]	COMMON CRITERIA ASSURANCE CONTINUITY: CCRA REQUIREMENTS VERSION 2.1 JUNE 2012
[FIPS-IG]	Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program May 7, 2019
[PCIPTS-PG]	Payment Card Industry (PCI) PIN Transaction Security (PTS) Device Testing and Approval Program Guide Version 1.8 March 2018
[PCIPTS-DTR]	Payment Card Industry (PCI) PIN Transaction Security (PTS) Point of Interaction (POI) Modular Derived Test Requirements Version 5.1 March 2018
[EMV-SEP]	EMVCo Security Evaluation Process Version 5.1 June 2016
[EMV-BL10]	EMVCo SEWG Bulletin 10 First Edition June 2014
[SC27WG3-2018SPR]	Introduction of new assurance requirements in ISO/IEC 15408-3 and evaluation methodology in ISO/IEC 18045 covering patch management and deployment activities: Study Period Report
[JIL-CODELOAD]	Joint Interpretation Library – Security Requirements for Post-Delivery Code Loading Version 1.0 February 2016
[CCPART1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model April 2017 Version 3.1 Revision 5
[CCPART2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components April 2017 Version 3.1 Revision 5
[CCPART3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components April 2017 Version 3.1 Revision 5