TRUSTWORTHY SOFTWARE

PATCHING SUMMARY

April 2016
Issue 1.1

TS543-6-01

FOREWORD

TRUSTWORTHY SOFTWARE INITIATIVE

The Trustworthy Software Initiative (TSI) is part of the UK Government's National Cyber Security Programme to improve the UK's ability to combat cyber risks and ensure that the UK leads the way in trustworthy software systems and expertise.

The objective of TSI is to provide the knowledge, skills and capability for supply, demand and education communities such that trustworthy software can be designed, implemented, sustainably maintained and assured in a risk-based, whole-life process.

TSI works with organisations and individuals in the UK, including government, academia, private/public companies, software developers and users, to achieve a recognised level of trust of software by providing targeted education, skills, standards and guidance.

For more information visit the TSI website: www.uk-tsi.org

LICENCE CONDITIONS

DISCLAIMER

MAINTENANCE/CONTRIBUTIONS

TSI welcomes input from Stakeholders on all aspects of its activity, including any additions or amendments to the Trustworthy Software Library.

If you have any suggestions please feel free to engage with TSI, either through your existing contact, or by emailing enquiries@uk-tsi.org.

1 OVERVIEW

1.1. TRUSTWORTHY SOFTWARE

With our daily lives and industrial processes becoming increasingly reliant on a wide range of underpinning software, there is a pressing need to address the quality and robustness of that software in order to establish its "trustworthiness" and therefore ensure that it performs as it should, when it should and how it should, enshrining the relevant elements of the Facets of Trustworthiness.

- Safety - the ability of the system to operate without harmful states
- Reliability - the ability of the system to deliver services as specified
- Availability - the ability of the system to deliver services when requested
- Resilience - the ability of the system to transform, renew and recover in timely response to events
- Security - the ability of the system to remain protected against accidental or deliberate attacks

1.2. TYPES OF IRREGULARITY

No software, other than those of a trivially small nature, can be proven, or even be expected, to be completely free of irregularities, comprising of either a Deviation or a Defect:

A deviation can be classed as:

- an unexpected outcome during run-time that is not caused by a coding error or mistake; and/or
- non-conformity with an explicit/implicit software requirement, due to misinterpretation (or otherwise) of the System Requirement Specification (SRS).

A defect can be classed as:

- a coding error/mistake; and/or
- non-fulfilment of an explicit/implicit software requirement.

It is the latter category, Defects, that are typically of most concern, and the rectification of such problems whilst in service is typically referred to as Patching.

1.3. PURPOSE AND APPLICABILITY OF DOCUMENT

This document provides guidance as to how Patching practices should be adopted by organisations in order to ensure the trustworthiness of the software they produce, procure or use, by mapping the overall concepts, principles and techniques for trustworthy software to this particular lifecycle activity, and providing signposts to related sources of amplifying information that may not otherwise been known to the readership.

1.4. OTHER DOCUMENTS

A fuller version of this information is available as the TS Patching Guidance, TS543-1-01.

2 PATCHING LIFECYCLE

2.1. OBJECTIVE

There are several *de facto* standardised models relating to System Lifecycle Processes [e.g. ISO/IEC 15288:2015] and Software Lifecycle Processes [e.g. ISO/IEC 12207:2008], but for Patching a distinct 4 Phase lifecycle model is more relevant, as illustrated in Figure 2-1.

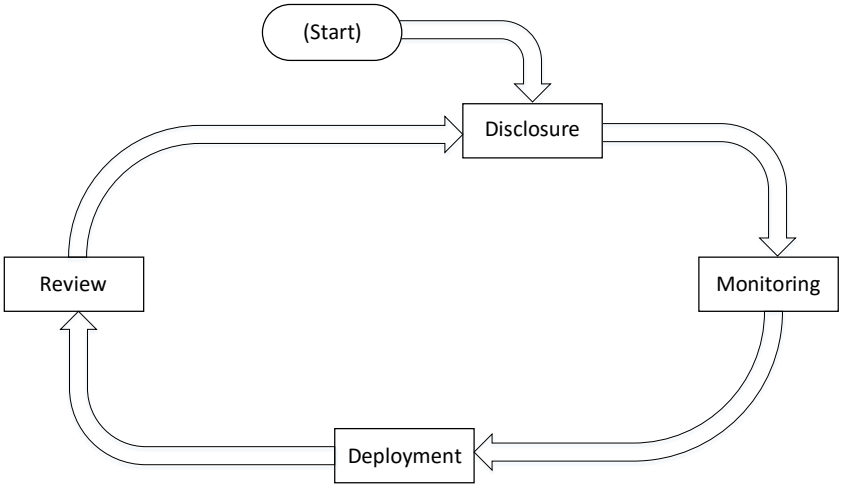

Figure 2-1

2.2. DISCLOSURE

The Disclosure Phase can be considered as consisting of 3 Sub-phases, as illustrated at Figure 2-2.

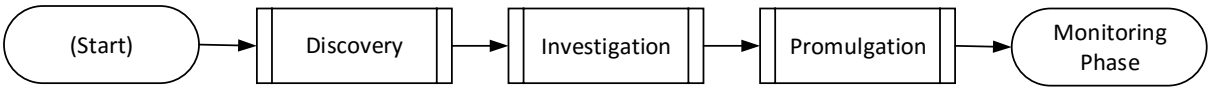

Figure 2-2

DISCOVERY

Those who discover Weaknesses and/or Vulnerabilities are generically referred to as “Researchers”, but this does not necessarily imply persons with a Research role. Once discovered, there are 3 primary ways in which Researchers outside of Producer organisations may act:

- Responsible Disclosure, whereby the information is passed to the Producer organisation to manage the Investigation and Promulgation Phases
- Non-disclosure, whereby the discovery is not released publicly, either as producer organisations are unable and/or unwilling to perform remediation, or by being kept by those of malign intent
- Full Disclosure is where there full details, including potentially means of exploitation, are put into the public domain, raising the risk that exploitation may occur before any remediation has been implemented

There are standardised approaches to formatting information for exchange, using the Common Weakness Enumeration (CWE) [ITU-T X.1524] and Common Vulnerabilities and Exposures (CVE) [ITU-T X.1520].

INVESTIGATION

Once discovered, the producer organisation investigates the Weakness and/or Vulnerability, and attempts to remediate the problem(s), as described in the relevant standard [ISO/IEC 30111:2013].

The remediation may take several forms:

- As a dedicated "patch", being a partial, incremental replacement of existing software solely to deal with one set of issues
- As an unplanned, additional minor update[1] which fully replaces existing software, and may be solely related to a single remediation and/or include multiple remediations
- As part of a planned minor[2] or major[3] update which fully replaces existing software, which will include multiple remediations and other changes

PROMULGATION

Once a remediation has been found and tested, the producer organisation will inform those affected, as described in the relevant standard [ISO/IEC 29147:2014].

RESPONSIBILITIES

The final Promulgation of information by a Producer organisation to interested parties should be sanctioned explicitly by their Trustworthy Software Release Authority (TSRA) [BS PAS754:2014].

RECORDING REQUIREMENTS

The UK standardised approach to Trustworthy Software [BS PAS754:2014] identifies documentation that will need to be amended accordingly during this Phase:

- Realisation Trustworthy Software Defect and Deviation List (R-TSDDL)
- Trustworthy Software Constraint and Dependency Model (TSCDM)
- Trustworthy Software Release Notice (TSRN)

[1] Typically reflected as the 3rd element of the Version Number, for instance the "Z" of Version X.Y.Z
[2] Typically reflected as the 2nd element of the Version Number, for instance the "Y" of Version X.Y
[3] Typically reflected as the 1st element of the Version Number, for instance the "X" of Version X.Y

2.4. MONITORING

The Monitor Phase can be considered as consisting of 3 Sub-phases, as illustrated at Figure 2-3.

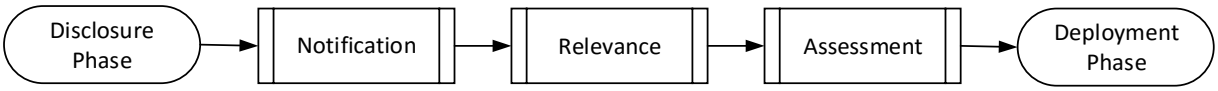

Figure 2-3

NOTIFICATION

In order to understand the emergent Weaknesses and/or Vulnerabilities that may affect an organisation, the Monitoring process needs to anchor to relevant sources of Notification, be they directly from Producer organisations or from other public domain feeds.

There are standardised approaches to formatting information for exchange, using the Common Weakness Enumeration (CWE) [ITU-T X.1524] and Common Vulnerabilities and Exposures (CVE) [ITU-T X.1520], but these are not universally followed.

RELEVANCE

The relevance of Notifications will initially depend on the nature the organisation:

- Those producing software will be interested in both Weaknesses that could apply to their software, and Vulnerabilities that could apply to externally sourced software components.
- End user organisations will only be interested in Vulnerabilities that could apply to externally sourced software components

A maintained Software Asset Register, including detailed version information, is fundamental to being able to Triage incoming Notifications for relevance to the organisation, and a *de facto* standardised approach called the Common Platform Enumeration [ITU-T X.528] may be used to format information for exchange.

ASSESSMENT

Provided that a Notification is Relevant, it is then necessary to carry out a risk-based assessment of how Important and Urgent any action may be, based on two main sets of considerations:

- The generic issues presented by the potential problem, which are often characterised as being Routine, Critical or Emergency
- The particular issues relating the potential problem to the organisational usage

There are *de facto* standardised approaches to this Assessment Phase of Monitoring, with probably the best known being a methodology that takes a Security lens to the challenge, the Common Vulnerability Scoring System (CVSS) [ITU-T X.1521]. It should be noted that underlying assumptions within the CVSS approach do not necessarily easily map to scenarios where:

- A defect will manifest itself in a time-dependent and/or random manner, without any need for network level or direct interaction with the affected system
- The impacts are not of a Security nature

- The system affected is beyond the "classical IT" realm, for instance embedded controllers

RECORDING REQUIREMENT

The UK standardised approach to Trustworthy Software [BS PAS754:2014] identifies documentation that will need to be amended accordingly during this Phase:

- In-service Trustworthy Software Defect and Deviation List (I-TSDDL)

2.5. DEPLOYMENT

The Deployment Phase can be considered as consisting of 3 Sub-phases, as illustrated at Figure 2-4.

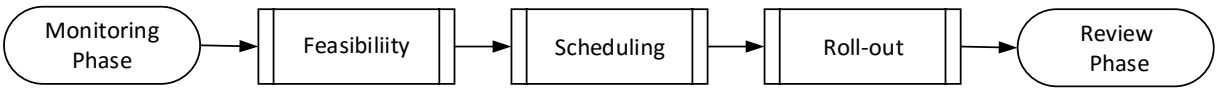

Figure 2-4

A variety of other publications provide specific and/or domain specific guidance on this Phase, including but not limited to:

- "Good Practice Guide – Patch Management", UK Centre for Protection of National Infrastructure (CPNI), 24 October 2006
- "Recommended Practice for Patch Management of Control Systems", US Department of Homeland Security (DHS) National Cyber Security Division (NCSD), December 2008
- SP800-40 "Guide to Enterprise Patch Management Technologies", US National Institute of Standards and Technology (NIST), Revision 3, July 2013
- "Cyber Essentials Scheme", HM Government (HMG), April 2014

Reviewing the information in this document alongside other such practices could allow the identification of gaps and enhancements.

FEASIBILITY

Once a patch has been Assessed for Importance and Urgency, the Patching Scenarios for the affected software will drive Feasibility of Deployment, as not all software can be realistically remediated. In the latter instance, then a review will be required as to alternative means of mitigation, which in the most extreme cases may require the software to be removed from use.

SCHEDULING

The ability to Schedule Deployment will need to consider a number of potentially Confounding Factors, including:

- Mission criticality of the service being provided, which can impact in two distinct manners:
    - A restriction on slots provided for outages, if the application of a patch cannot be performed without causing an interruption to service

  - Concerns over deleterious collateral effects (either direct, or on other elements of the system or systems to which it is connected) of the application of patches, which may require offline "regression testing" against a representative configuration to ensure there are no such deleterious collateral effects

- Nature of network connectivity including bandwidth and time limitations, or physical accessibility
- Ability of the organisation to control its own patching destiny, including any dependence on 3rd parties
- The organisational authorisation processes for deployment of patches

These considerations are obviously not unique to a patch deployment, as they will also apply to other forms of update, so an aligned approach to achieving "current maintained version" in both instances is to be commended.

ROLL OUT

The recommended approach is that opportunities should be sought to achieve current maintained version (for both patches, and other updates) at the earliest appropriate juncture, as the longer the the more likely a problem from the defect and/or deviation is to manifest itself.

The exact timing of software deployments will vary with organisation and time, but an indicative approach to the various ways of achieving current maintained version is provided in Table 2-1.

<table>
<tr><th rowspan="2">Deployment Model</th><th colspan="3">Update Characteristic</th></tr>
<tr><th>Routine</th><th>Critical</th><th>Emergency</th></tr>
<tr><td>Patch
Additional unplanned update[5]</td><td>Balance against Trade-offs and Confounding Factors, but aim to achieve within short period[4]</td><td>Urgent deployment will typically dominate Trade-offs and Confounding Factors</td><td>Immediate deployment should dominate Trade-offs and Confounding Factors</td></tr>
<tr><td>Planned minor update[6]</td><td>Allow initial period[7] for problems to identified and resolved, then plan deployment a.s.a.p.</td><td colspan="2" rowspan="2">Not applicable</td></tr>
<tr><td>Planned major update[8]</td><td>Allow reasonable period[9] for problems to identified and resolved, then plan deployment a.s.a.p.</td></tr>
</table>

Table 2-1

RECORDING REQUIREMENT

The UK standardised approach to Trustworthy Software [BS PAS754:2014] identifies documentation that will need to be amended accordingly during this Phase:

[4] Typically no greater than 1 month for OP/AP deployments, and 3 months for DP deployments
[5] Typically reflected as the 3rd element of the Version Number, for instance the "Z" of Version X.Y.Z
[6] Typically reflected as the 2nd element of the Version Number, for instance the "Y" of Version X.Y
[7] Typically no less than 1 month for "IT" deployments
[8] Typically reflected as the 1st element of the Version Number, for instance the "X" of Version X.Y
[9] Typically no less than 1 quarter for "IT" deployments

- In-service Trustworthy Software Defect and Deviation List (I-TSDDL)

2.6. REVIEW

The Review Phase can be considered as consisting of 3 Sub-phases, as illustrated at Figure 2-5.

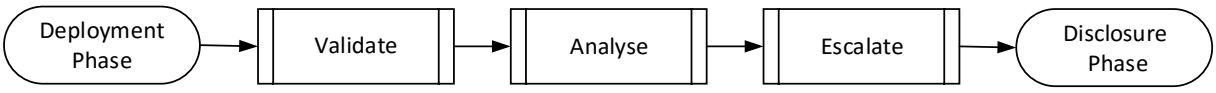

Figure 2-5

VALIDATE

Once a patch has been Rolled Out, it is prudent to validate the situation:

- By spot checking a number of installations to ensure the correct version(s) have been applied
- By using such spot checks to ensure that 100% of the targeted sample have had patches applied

ANALYSE

Once in service, system events should be monitored for any potentially deleterious effects arising from the application of patches:

- Any collateral effects that may have arisen from the specific implementation configuration, which either may not have been detected during regression testing, or where no regression testing was performed
- Any new weakness and/or vulnerability that may be revealed subsequent to the installation of the patch

ESCALATE

If the Analysis sub-phase identifies any problem:

- Any collateral effects will need to be appropriately escalated and treated internally or with any relevant 3rd party service provider
- Any new weakness and/or vulnerability will need to appropriately escalated to the producer organisation

RECORDING REQUIREMENT

The UK standardised approach to Trustworthy Software [BS PAS754:2014] identifies documentation that will need to be amended accordingly during this Phase:

- In-service Trustworthy Software Defect and Deviation List (I-TSDDL)