Xen Project Security Whitepaper

Author: Lars Kurth

Version 1.0 (published May 18, 2018)

The last time we updated the Xen Project Security Process, was 3 years ago (in March 2015). Last year, we attempted to clarify what constitutes a Xen Vulnerability, which was killed off by The Register. To see whether we need to consider further changes

This document contains

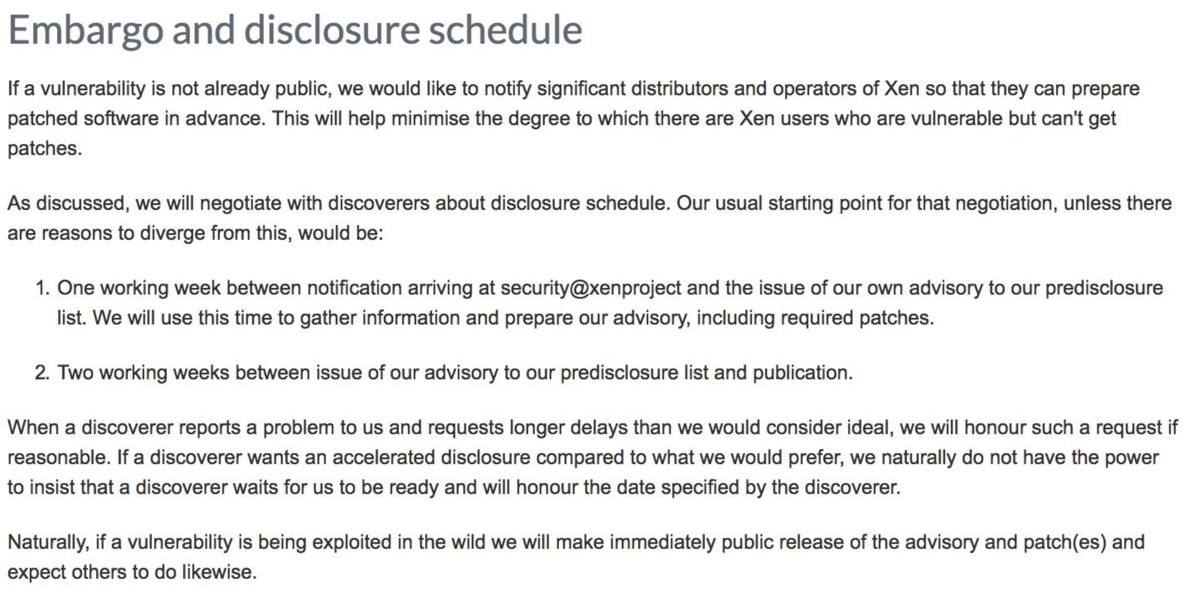
- Baseline: an analysis of our XSAs and how we dealt with XSAs.
- Community Consultation
    a. Feedback received from a community consultation
    b. Analysis
- Recommendations and policy changes

1. Baseline: Analysis of XSAs and Security Related Activities

1.1. Batching of Security Issues

Batching security issues: we have been paying greater attention to the benefits of batching security issues for about 12 months, sparked by a discussion on xen-devel@. That discussion did not lead to a formal change to the process document, but it did lead to a difference in emphasis in Security Team practice..

The policy says:



Embargo and disclosure schedule

If a vulnerability is not already public, we would like to notify significant distributors and operators of Xen so that they can prepare patched software in advance. This will help minimise the degree to which there are Xen users who are vulnerable but can't get patches.

As discussed, we will negotiate with discoverers about disclosure schedule. Our usual starting point for that negotiation, unless there are reasons to diverge from this, would be:

1. One working week between notification arriving at security@xenproject and the issue of our own advisory to our predisclosure list. We will use this time to gather information and prepare our advisory, including required patches.
2. Two working weeks between issue of our advisory to our predisclosure list and publication.

When a discoverer reports a problem to us and requests longer delays than we would consider ideal, we will honour such a request if reasonable. If a discoverer wants an accelerated disclosure compared to what we would prefer, we naturally do not have the power to insist that a discoverer waits for us to be ready and will honour the date specified by the discoverer.

Naturally, if a vulnerability is being exploited in the wild we will make immediately public release of the advisory and patch(es) and expect others to do likewise.

Over approximately the past 12 months, the Security Team have tended to regard the existence of several advisories which could be combined into a batch, as a reason for diverging from the basic Timetable. It is important to note that our process does **not require adherence** to the time-table laid out in the policy document.

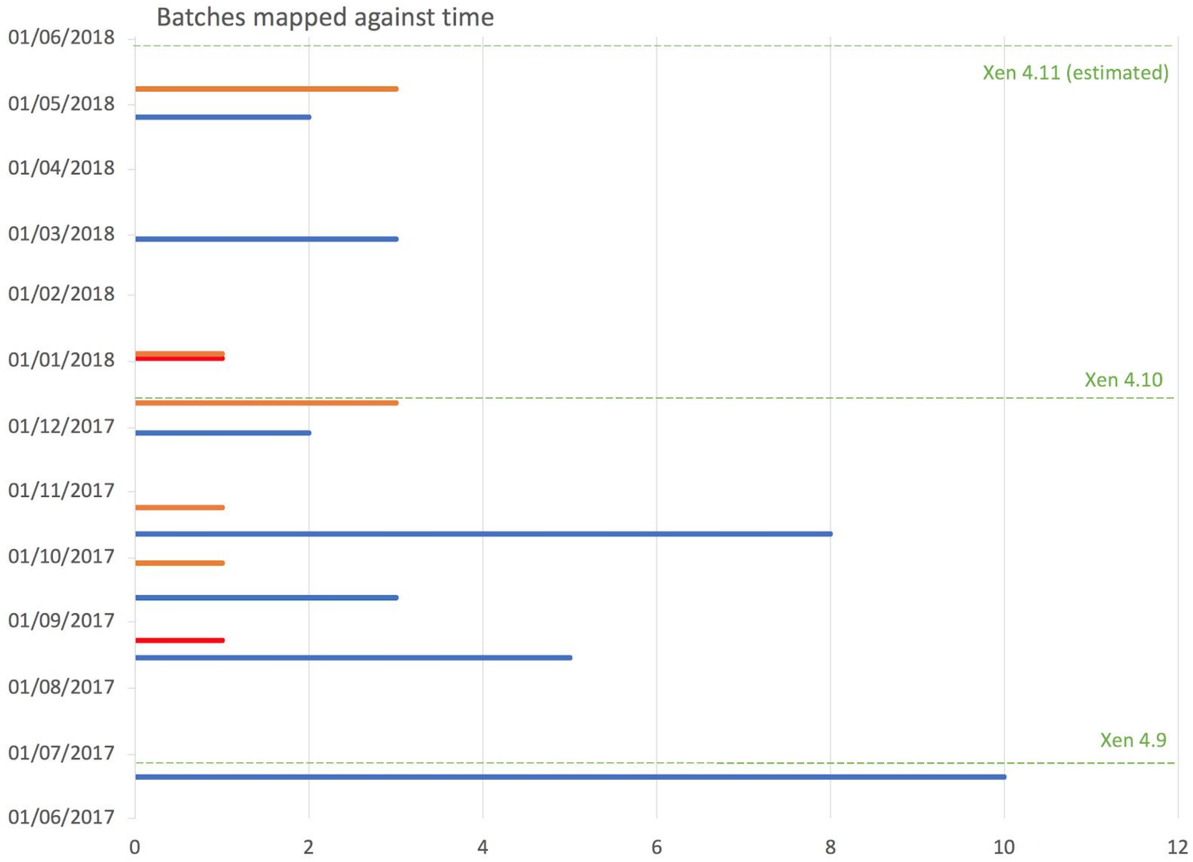
The following table, gives an overview over batched and non-batched security issues, covering this time-frame.

XSAs	Batch Size	Public Release (Weeks since last batch)	Comment
260,261,262	3	2018-05-08 (1.9)	2nd Tue of May XSA-260 had a date set by the discoverer
258,259	2	2018-04-25 (8.1)	4th Wed of Apr
252,255,256	3	2018-02-27 (7.7)	4th Tue of Feb
253	1	2018-01-04 (0.1)	Released as a Xen 4.10 only update
254	1	2018-01-03 (3.1)	Meltdown/Spectre: publicly disclosed by discoverers
248-251	3	2017-12-12 (2.0)	Batch released because it blocked the 4.10 release
246-247	2	2017-11-28 (5.0)	4th Tue of Nov
236	1	2017-10-24 (1.7)	Could not identify reason for release date Possibly date set by discoverer
237-244	8	2017-10-12 (2.0)	2nd Thu of Oct
245	1	2017-09-28 (2.3)	ARM only Date set by discoverer
231-234	3	2017-09-12 (2.9)	2nd Thu of Sept
235	1	2017-08-23 (1.1)	Was not embargoed: The issue was discussed publicly before being recognized as a security issue
226-230	5	2017-08-15 (8.0)	3rd Tue of Aug
216-225	10	2017-06-20 (N/A)	3rd Tue of Jun Date impacted by 4.9 release
No real attempt to batch security issues prior to this.			

Legend:
Batching succeeded
Batching could have succeeded
Batching was out of our control

The Security Team has almost always tried to negotiate a 2-week predisclosure period with discoverers - i.e. without diverging in this respect from the basic timetable set out in the policy.

The following graph shows the batches mapped against time: dates are public disclosure dates.



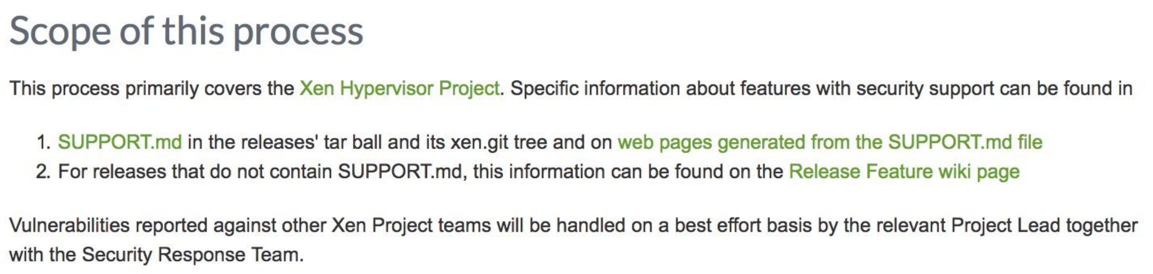
Releases are displayed as green dotted lines

<u>In summary:</u>

- We released 14 batches of XSAs in 12 months, of which 2 we had no control over (the issue was publicly released already or the issue was disclosed by the discoverer)
- Some smaller batches (mostly those shown in orange) could probably have been folded into a larger batch.
- We seem to struggle coordinating the date of Xen releases with XSA releases, which frequently leads to two batches in short succession.
- Generally though, the graph shows that on average we release a batch more or less every month

1.2. Becoming a CVE Numbering Authority

In addition we took steps to become a CVE Numbering Authority. This has resulted in the creation of SUPPORT.MD for Xen 4.10 and 4.11, a generated comparison table at https://xenbits.xen.org/docs/unstable/support-matrix.html and minor changes to the security policy:



Scope of this process

This process primarily covers the Xen Hypervisor Project. Specific information about features with security support can be found in

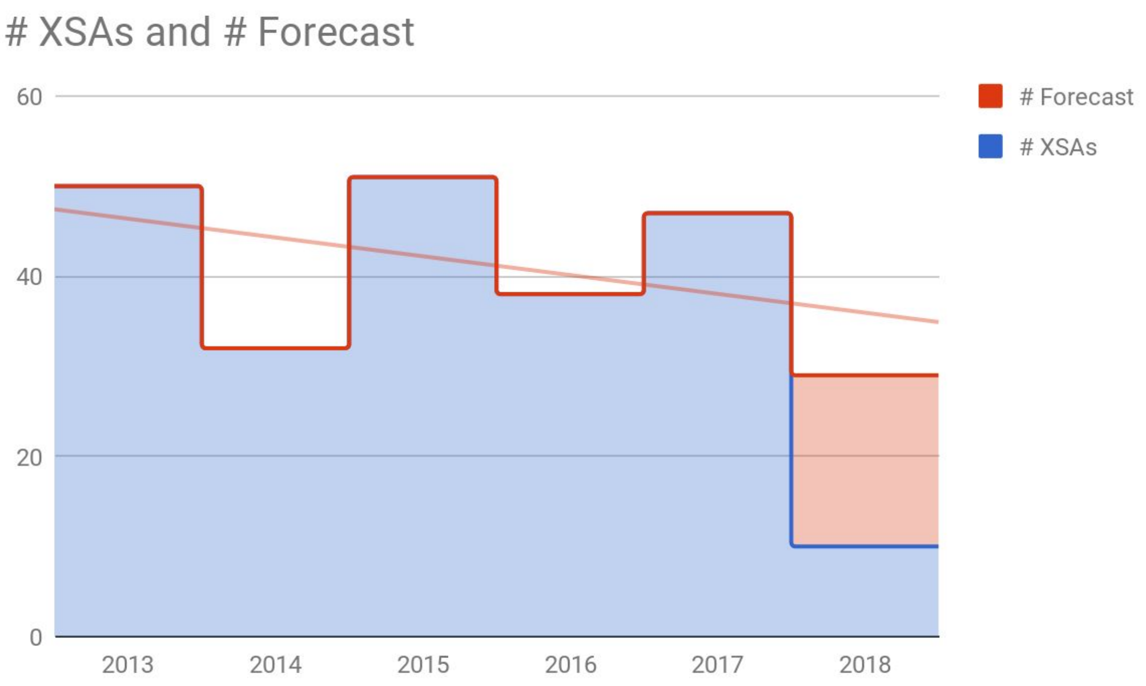
1. SUPPORT.md in the releases' tar ball and its xen.git tree and on web pages generated from the SUPPORT.md file
2. For releases that do not contain SUPPORT.md, this information can be found on the Release Feature wiki page

Vulnerabilities reported against other Xen Project teams will be handled on a best effort basis by the relevant Project Lead together with the Security Response Team.

Efforts in this direction are continuing and we have formally started the application to become a CNA.

1.3. Historical XSA Numbers and Forecast

I added historical data on XSA volume based on **public release dates** in the chart below, which includes a forecast for 2018, based on the number of XSAs raised in 2018 so far.

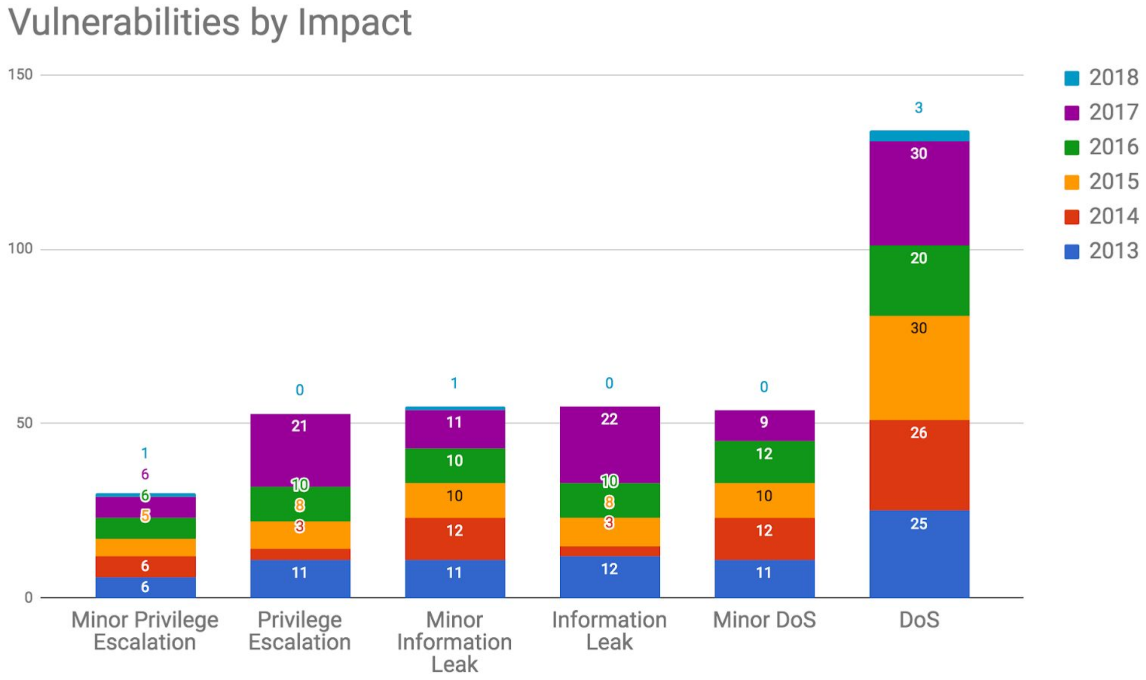


The next two sets of diagrams are from cvedetails.com using the Xen vendor category. These graphs deliver an approximation of the severity of XSAs, because

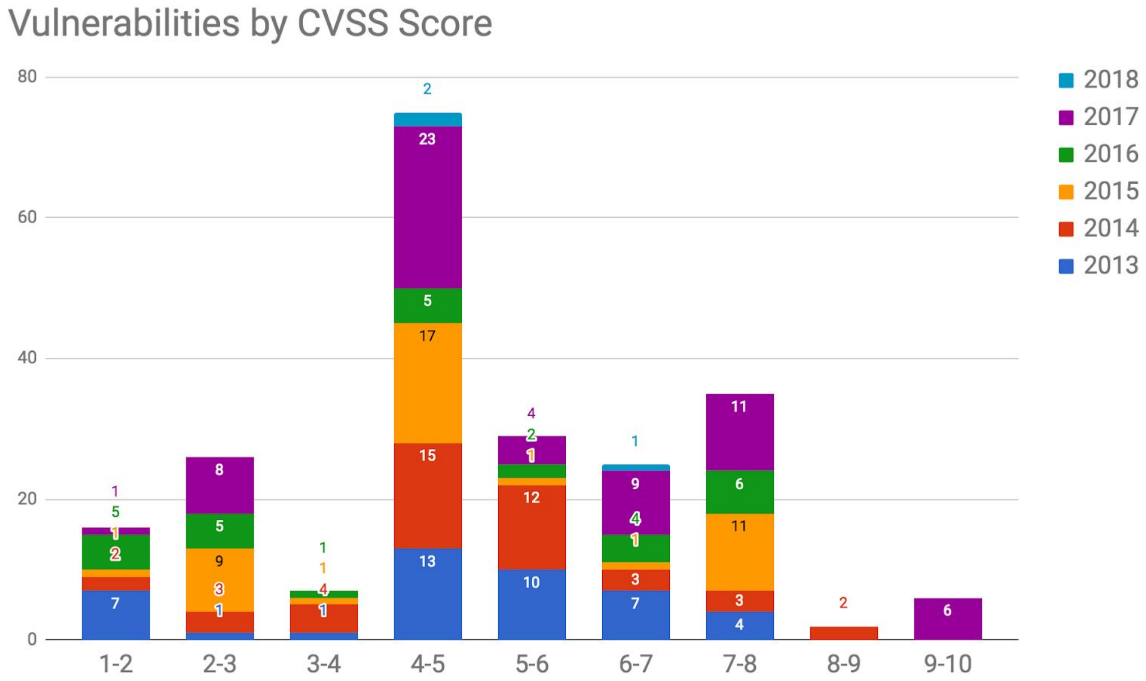
- Some XSAs have no CVE numbers, while others have multiple
- Some industry wide CVEs (e.g. Meltdown and Spectre) for which we have issued XSAs are not counted against the Xen vendor category
- The year mapping relates to when the CVE, not the XSA was issued. This often means that XSAs issued at the end of a year, get counted in the following year.

The Vulnerabilities by impact uses the CVSS Impact metrics and calculated in the following way:

- Integrity Impact = Partial: **Minor Privilege Escalation**
- Integrity Impact = Complete: **Privilege Escalation**
- Confidentiality Impact = Partial: **Minor Information Leak**
- Confidentiality Impact = Complete: **Information Leak**
- Availability Impact = Partial: **Minor DoS**
- Availability Impact = Complete: **DoS**



Note: A vulnerability may be in several impact groups.



2. Community Consultation

Given that we have not changed the Xen Project Security Process for 3 years and that we experimented with batching of Security Issues, it is worthwhile to review how well the process works. Note that there was an attempt to change the process in Dec 2016 (see here), which fizzled out due to lack of engagement.

To avoid this, I am intending to run a more hands-on consultation with the following pattern:

- **Done:** Collected some data on pain points based on conversations I had in the last 6 months with a number of stakeholders that approached me (included in this document)
- **Done:** Collect additional data via a public consultation: see here (included in this document)
- **We are here:** Distribute White Paper for discussion on xen-devel@ and proactively invite community members to comment. Note that without sufficient engagement from users, I do not want to spearhead a process change.
- Condense output of this discussion into a concrete change proposal to be voted on by the Project Leadership team. This may require several iterations.

2.1. Feedback Received

I have received feedback from the following organisations: Citrix, Gandi, Gentoo package maintainer, Invisible Things Lab, Star Lab, Rackspace, Oracle.
And individuals: Steven Haigh, John Thomson, Xen Release Managers

It is also worth saying that there was positive feedback also, which due to the nature of this document may be missed. Here are a few quotes:

- *"In general the way things work are present is fine for us. I like the suggested idea of a (where practical) single monthly window. 14 days notice works well for us, especially if we decide we have to notify our customers of disruptive reboots after working through the reports"*
- *"Overall very happy with XSA process from the perspective of packaging Xen for personal use. Just would like clarification, consistency, and a little more metadata."*
- *"Thanks for the thoughtful analysis. From our point of view: Batching is ok and desirable, unless conflicts with other goals (such as keeping information confidential)."*

2.2. Pain Points Identified

These section contains pain points that were highlighted as part of feedback from the consultation. The table below explains the format used

Group of issues			
X. Issue Headline	Issue description *Information on feedback. Note that Freq column (in this example 9/10) indicates that nine out of ten respondents have highlighted this issue.*	Possible issue resolutions	*9/10*

Feedback related to the Security Vulnerability Process and/or its application

Issue	Description	Possible Mitigation	Freq
2.2.1. PROCESS RELATED			
A. Batching	See section 1.1 *The vast majority of respondents liked batching or did not see any downsides, unless it contradicts with other process goals.*	Continue what we do now. However it may be sensible to formalize batching within our process.	*9/10*
B. Workload	Too many security issues published in one batch, leading to capacity issues in downstreams *Workload for large batches was an issue for product companies as well as individuals. However 2 organisations have no issues even with larger batches of XSAs.*	No Batching Batching with extended pre-disclosure period	6/8
C. Predictability	Unpredictability of new pre-disclosure announcements impacting scheduled plans for upgrades, new releases, holidays, etc. *This was somewhat mixed. Of the 6 product companies, 4 highlighted that lack of predictability (in particular for large batches of more than 4 XSAs) is a problem. 2 would not like to extend pre-disclosure periods just to achieve better predictability.*	Batching with a fixed publication schedule	4/6
D. Agreeing Release Timing of an XSA	This is from a discoverer of an issue: *"Maybe this is exceptional case, but I was not happy about XSA-XXX handling. The issue was reported to security@xenproject.org on date X, the patches were ready 6 weeks later and due to batching pre-disclosed 5 weeks later. The communication around agreeing the public release date was poor."*	Improve Communication Fixed publication schedule	*1/10*

In summary:

- Item D appears to represent a failure of the process, which is supposed to give discoverers control of disclosure schedules. Hopefully it is a one-off. If discoverers of other issues have similar concerns, we will need to address this.

- A, B and C are related and are worth investigating further and section 3 contains some discussion related to this item, which looks at different trade-offs. Whatever we do, it must not lead to an increase of workload on Security Team members.
- Note that there is also a cross-over with issues in section 2.2.4 of this document

Things which should be looked at, because they are either easy to address or occur frequently,

2.2.2. WORKFLOW / TOOLS RELATED REPORTED FREQUENTLY OR EASY TO FIX			
A. XSA Re-issues	Fixes continue to be refined during the 2-week embargo, thereby reducing time available to packaging and testing. This is in particular true for substantial re-issues close to the end of the pre-disclosure issue. *The vast majority of respondents identified this as an issue.*	Unclear at this stage.	*9/10*
B. XSAs without CVE numbers	XSAs without CVE numbers are painful for distro package maintainers as well as some product vendors: an updated version of an XSA with CVE number requires to rebuild packages. *This is a variant of A, but it is easier to fix this.*	Become CNA, such that we are not dependent on 3rd parties to issue CVE numbers.	*5/10*
C. Livepatch creation	Creation of viable live patches Security team does this informally now *The majority of respondents would welcome a more organized approach to handling live patches as part of the XSA process. However, this can only be done, if there is no extra workload on the security team.*	Unclear at this stage.	*7/10*
D. Inconsistent Meta Data and XSA prerequisites	The XSA Meta Data is not consistently applied to all XSAs and the meta data structure is not well documented (or it is not known where the documentation is stored). That makes it hard to develop tooling that helps with automation/verification. *Issues in this area were raised by a few respondents. However only one had a number of concrete suggestions for how meta-data could be improved.*	Clarify and/or document meta data structure. I think the best approach here would be the security team and the provider of the feedback to discuss.	*4/10*

E. No XSA update number in email subject	XSA announcements currently do not contain a version number in the subject line *Only one respondent, but this looks like an easy fix.*	Change tooling such that we send “**Xen Security Advisory ABC vD …**” instead of “**Xen Security Advisory ABC …**”	` *1/10*

In summary:

- Easy to fix: Item B is already being addressed and E could easily be fixed.
- A and C are related and potentially difficult to resolve, because consumers of XSAs may have their own patch queues and other differences from upstream Xen. Looking at the discuss-list as an indicator of issues with livepatch-capability and issues discovered by consumers of XSAs show
  - **2018:** 50% were related to backports (aka someone providing a packport), 50% were non-issues
  - **2017:** 33% were related to backports, 6% were non-issues, 22% were issues discovered by consumers (XSAs affected: 209, 224, 226) and 39% questions about the patches or general questions.

  This isn’t a strong indicator that we have a problem. But, I have not trawled through the history of XSA re-releases and thus no exact data on re-sends of XSAs during and after pre-disclosure. So I will probably have to do some further analysis.
- For D, it may be worthwhile to set a community call and/or public discussion before considering further steps. Today the security team mostly uses the meta-data for their internal tools. Some of these tools depend on non-public git repos. However, this is an area where relatively little effort may improve the life of downstreams.

Things which are very dependent on individual workflows and thus are likely not to be addressed.

2.2.3. WORKFLOW / TOOLS RELATED REPORTED INFREQUENTLY OR NOT EASY TO FIX			
A. Tedious to identify which XSAs apply	Distro package maintainers (in particular those that have to support several Xen versions) find it hard to wade through the complexities of identifying which patches to apply. *Feedback from one person*		*1/10*
B. Git baseline of patches	It is often necessary to 'tweak' the patches to make sure they apply cleanly (in particular for live patches), which is often non-trivial. For auditing and verification it would be helpful to know exactly the base commit for each xsa patch stated in in the XSA notice. *Feedback from two people*	Publish a non-embargoed git commit ID or tag per patch that describes the base of a patch. This should include whether this is staging or master (or which tree in general - see [1]).	*2/10*

In summary:

- Issue A and B are likely related: in other words A may be a symptom of B
- B may be easy to fix: this may be worth a discussion

The next section contains issues that are release cycle related. The items in this list is not directly related to feedback, but has come up several times in private conversations, on IRC and on xen-devel@

2.2.4. RELEASE CYCLE RELATED			
A. Too many security supported Xen releases	Security team has to backport security fixes to too many releases. With the 6 months release cycle and 36 months of security support, 6 releases have to be supported.	Change release cycle to a longer cycle than 6 months. A release cycle of 9 months would lead to 4 security supported releases, a cycle of 12 months to 3. Another alternative would be to have some releases with shorter security support life times.	
B. Coordination with Xen Releases	The unpredictable nature of XSA publication, together with hardening activities in the lead-up to a release, is leading to race conditions which either delay the release and also make batch planning difficult. It is also potentially an issue for downstream releases.	Batching with a fixed publication schedule aligned with a release schedule could address this issue.	

In summary:

- A should be part of a wider consultation on whether the 6 month release cycle works and whether to keep or change it.
- B could potentially be solved by a fixed XSA publication schedule. However, possible side effects need to be explored. This item is discussed in section 3.1.

3. Recommendations

3.1. PROCESS RELATED

Earlier we established that issues 2.2.1 A - C and 2.2.4 B are directly related to the timing under which we release XSAs. Note that recommendations are slightly re-ordered.

R1) Recommendation: Batching

Generally, batching (see 2.2.1 A) is well received by the community, but it does require extra coordination amongst discoverers of an XSA. It also is not technically compatible with item 1 in section “Embargo and disclosure schedule” of our process, which says “*One working week between notification arriving at security@xenproject and the issue of our own advisory to our predisclosure list. We will use this time to gather information and prepare our advisory, including required patches*.”

I think generally, we have not been able to hold up the 1 working week between discovery and pre-disclosure. For large and complex security issues, it is also not possible to fix an issue and back-port it to multiple releases. Batching makes the situation worse.

As a large proportion of security issues have been discovered by Security Team members and in almost all cases an agreement with the discoverer can be agreed, which has enabled batching.

> We should formalize usage of batching in our policy.
>
> For reasons of transparency, we should change the timing requirement in “Embargo and disclosure schedule” to cover for complex issues and batching. We should also explicitly highlight the practice of batching within our process: we have informally done this for a year and get community approval and formalise the practice.

R2) Recommendation: Workload

Batch sizes of more than 4 to 5 XSAs can cause problems with workload by XSA consumers. It is worth looking into this in more detail, as there is a per batch and a per XSA cost associated with security vulnerabilities. The total overhead and resource needs

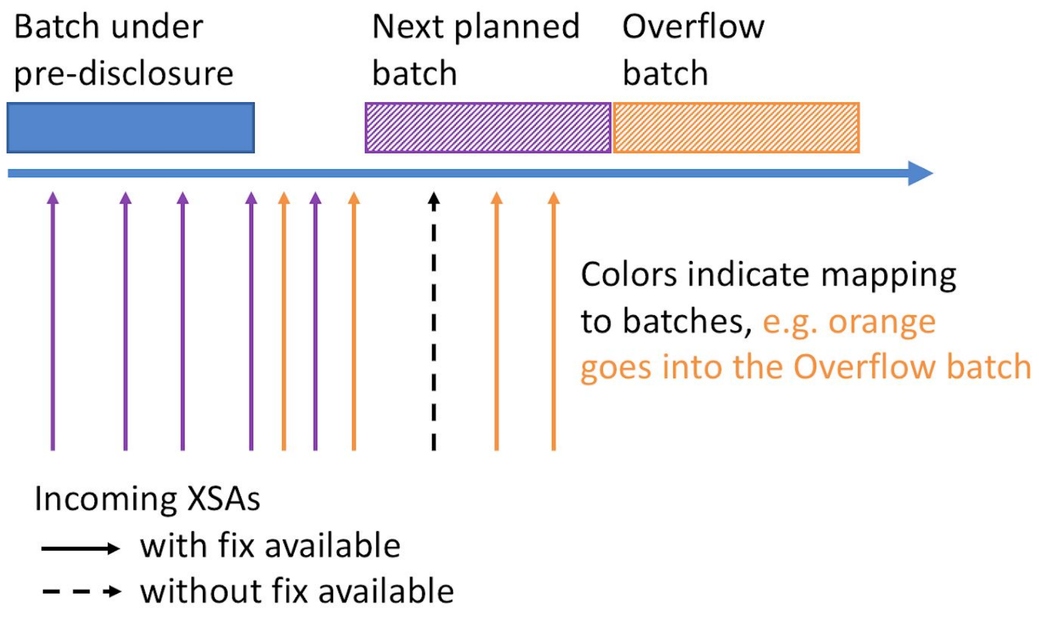
Per XSA: This includes

- Evaluating whether and how an XSA impacts a specific Xen implementation
- Effort to create PoCs and test cases for issues (that is something cloud providers as well as commercial distros do)
- Possibly backports to a vendors specific Xen environment including testing of these
- Creation of a viable live patch (which an increasing number of hosting vendors do)
- Creation of vendor specific XSA descriptions

For large scale operations that is significant and may cover several Xen Project releases and/or configurations. Note that re-issues of XSAs during pre-disclosure (see 2.2.2 A) have a significant impact on the cost per XSA.

Per batch: The main overhead is the import / build / test / deployment testing the patches or patch packages. In some cases, this process to be repeated for XSA re-issues (see 2.2.2 A) and for XSAs without CVE numbers (see 2.2.2 B) when CVE numbers become available.

Solution 1: One solution for this problem is to limit batch sizes, while keeping pre-disclosure constant at two weeks.

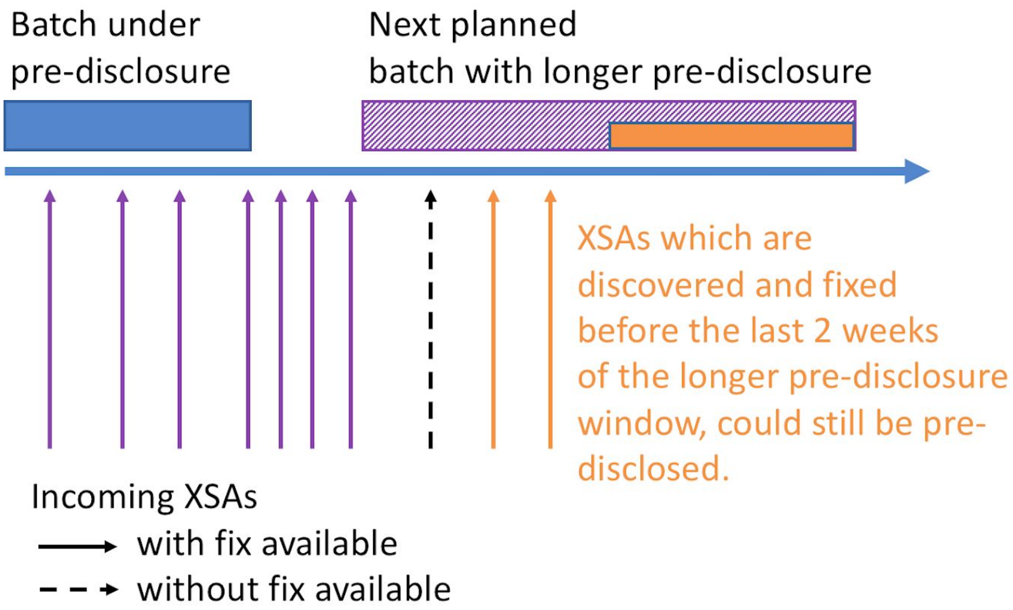

Batch under pre-disclosure
Next planned batch
Overflow batch
Colors indicate mapping to batches, e.g. orange goes into the Overflow batch
Incoming XSAs
with fix available
without fix available

In this scenario, we would try and limit a batch size and immediately afterwards release a second batch for pre-disclosure which would contain any XSAs that do not fit a given batch size threshold. As we normally will know upfront that we need an overflow batch, **both the next planned batch and overflow batch should be announced together** on xenbits.xenproject.org/xsa/ or in a similar location.

In the past, we have typically created two subsequent sets of batches towards the end of release cycles, with the goal to avoid publishing new major releases that do not contain the latest set of fixes. This has led to coordination issues and/or release slippage.

Such an approach appears to be workable, if we set the threshold of how many batches we use to a rate that allows for some slack. Let's look at the last 12 month of data of XSAs where we control the date and one batch per month, where we would create "overflow" batches if needed: in this case we had 42 XSAs, which makes an average batch size of 3.5. Thus a threshold for carry over into of 5 seems reasonable.

Solution 2: An alternative would be to increase the pre-disclosure period for large sets of batches, at the Security Team's discretion. This would look as follows


Batch under pre-disclosure
Next planned batch with longer pre-disclosure
XSAs which are discovered and fixed before the last 2 weeks of the longer pre-disclosure window, could still be pre-disclosed.
Incoming XSAs
with fix available
without fix available

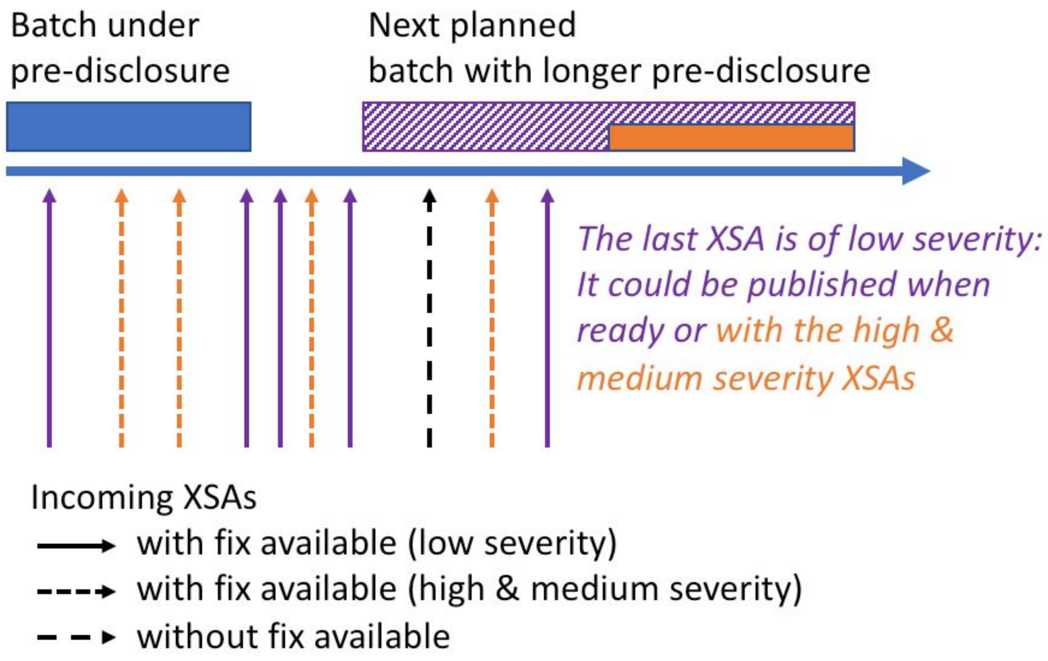
The primary drawback in this scenario is the extended pre-disclosure period, which would be double. **I do not believe community consensus for such an approach is achievable.** The following comment on batching from one of the contributors to this document highlights the problem.

> *With the current number of pre-disclosure members, keeping the information confidential is questionable. In case of critical bugs (reliably exploitable domU ➜ dom0 breakout) 2 weeks embargo are quite long. But in practice 2 weeks seems reasonable. I think it's ok to allow longer embargo periods for **low** severity issues. Maybe also for **medium** severity issues, but definitely not for **high** severity issues.*

However, determining the severity is potentially problematic, as it very much depends on the context of use. So an approach that is simple to administer and not context specific would be needed. An example of how this could be done comes from the contributor quoted earlier.

> *If embargo times were to be extended for low impact issues, a reasonable impact scale could be:*
>
> - ***Low:** DoS only, privilege escalation within domU (like domU user ➜ domU kernel)*
> - ***Medium:** Information leak*
> - ***High:** Privilege escalation domU ➜ dom0 (or hypervisor, or another domU)*
>
> *Or*
>
> - ***Low:** DoS only*
> - ***Medium:** Information leak, privilege escalation within domU (like domU user ➜ domU kernel)*
> - ***High:** Privilege escalation domU ➜ dom0 (or hypervisor, or another domU)*

This would look as follows

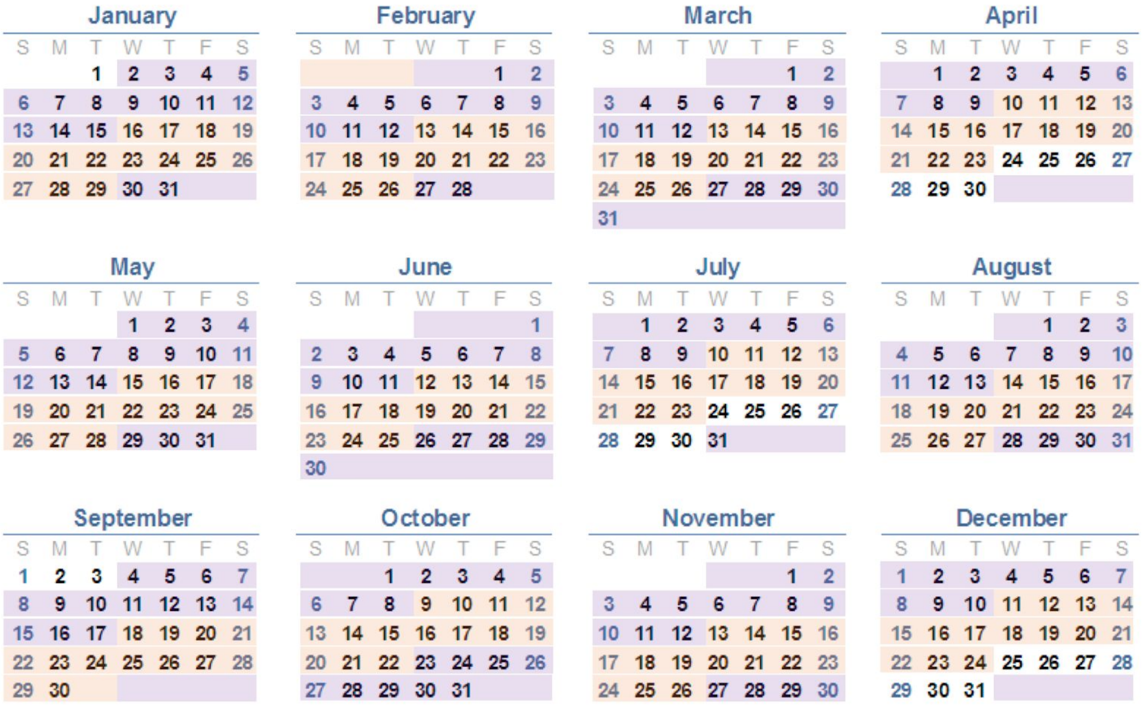

Batch under pre-disclosure
Next planned batch with longer pre-disclosure
The last XSA is of low severity: It could be published when ready or with the high & medium severity XSAs
Incoming XSAs
with fix available (low severity)
with fix available (high & medium severity)
without fix available

Comparison:

Stakeholder / Issue	**Solution 1**	**Solution 2**
Security team	Easy to administer	Adds complexity and workload to the security team.
Longer pre-disclosure	N/A	For low issues
Per XSA cost	Independent of the solution chosen	
Per batch cost		Double that of Solution 1
Public releases by downstreams	One	Two

<u>Impact on a year:</u>

The following diagram maps out the worst case scenario, XSAs with an overflow batch or extended pre-disclosure period every month for 2019. Note, that I have assumed a fixed release schedule for XSAs, in this case mainly for ease of generating the schedule.



January

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

February

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

March

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

April

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

May

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

June

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

July

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

August

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

September

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

October

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

November

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

December

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

In a nutshell, in this scenario, the project would be in a pre-disclosure period all year round. This would make managing Xen Project releases extremely difficult and compound 2.2.4 B (Coordination with Xen Releases), if a large number of issues are discovered in the months we are trying to cut Xen Project releases.

Observations:

- First of all, this graph shows that overflow batches in December are not a good idea, as it would collide with the winter holidays ⇨ *The security team would need to be able to defer XSAs from the overflow batch into January or release a large batch as an exception.*
- Also, the January batch, cannot be pre-disclosed until people are back from holidays ⇨ *Possible workarounds are: a later January disclosure date, or a one-off extended disclosure date that spans XMas.*

<u>Batch threshold and impact on XSAs:</u>

The table below shows historical XSAs per year and month. Assuming roughly a batch per month, this gives us an approximation of how many batches would hit a maximum threshold per batch.

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2013	**6**	**5**	0	**5**	4	**6**	1	1	**5**	**7**	**7**	3
2014	2	4	2	4	1	**5**	0	3	4	1	**5**	1
2015	2	1	**9**	1	1	**7**	2	2	2	**11**	4	**9**
2016	2	2	2	2	4	3	3	0	**5**	1	**9**	**5**
2017	0	**5**	2	1	3	**10**	0	**6**	**5**	**9**	2	4
2018	2	3	0	2	3							

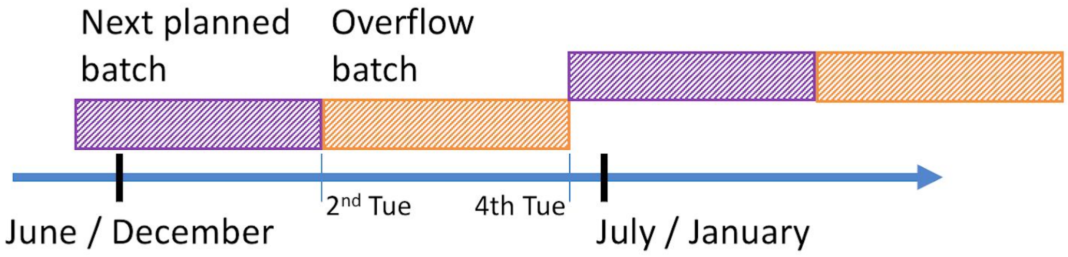
The table below outlines the months a year above different maximum threshold per batch and calculates the number of batches that require an overflow batch or extended pre-disclosure period per year:

Year	Threshold= 4	Threshold= 5	Threshold= 6	Threshold= 7
2013	7	4	2	0
2014	2	0	0	0
2015	4	4	4	3
2016	3	1	1	1
2017	5	3	2	2
2018	0	0	0	0

Assuming that we would want to address the workload problem, we would likely not want to end up with more than 3-4 batches per year above the threshold: this means we would have to set it at 6 or 7 XSAs per batch.

Impact on creating Xen Project releases:

It is necessary to look at 2.2.4 B (Coordination with Xen Releases) and evaluate the impact. The graph below shows a planned XSA batch with a possible overflow batch in months of releases. Again, I have assumed a fixed release schedule for XSAs, in this case mainly for ease of generating the schedule.



In most cases, there would not be a gap between an overflow batch or extended pre-disclosure period. This would make scheduling a release, which on average takes between 2-3 days and 1 week to prepare impossible. Even if there was a gap, we have a maximum of a week to make a release, which is manageable, but would not allow for anything going wrong.

The only ways to solve this issue, is to

1. Cut a new release in the knowledge that shortly after release a set of security issues would be published.
2. To suspend the mechanism to limit the workload of a batch to a certain in the month we are trying to release.

Recommendation:

Trying to limit the size of a batch through any of the possible methods proposed is **overly complex**, **would likely be hard to achieve community consensus on** and has also **negative effects on the project's capability to make releases**. In addition, the number of times when there is a spike leading to workload issues for consumers of XSAs is relatively low (e.g. with threshold > 6, there would have been 3 affected batches in 2 years and 5 in 3 years). Thus, **implementing a formal mechanism to address this issue is not recommended**.

Assuming we formalize batching in our process, this is probably best **handled by giving the Xen Security Team a certain degree of discretion** to move undisclosed XSAs into a future batch to try and informally limit batch size for the one or two instances this may happen per year.

R3) Recommendation: Predictability

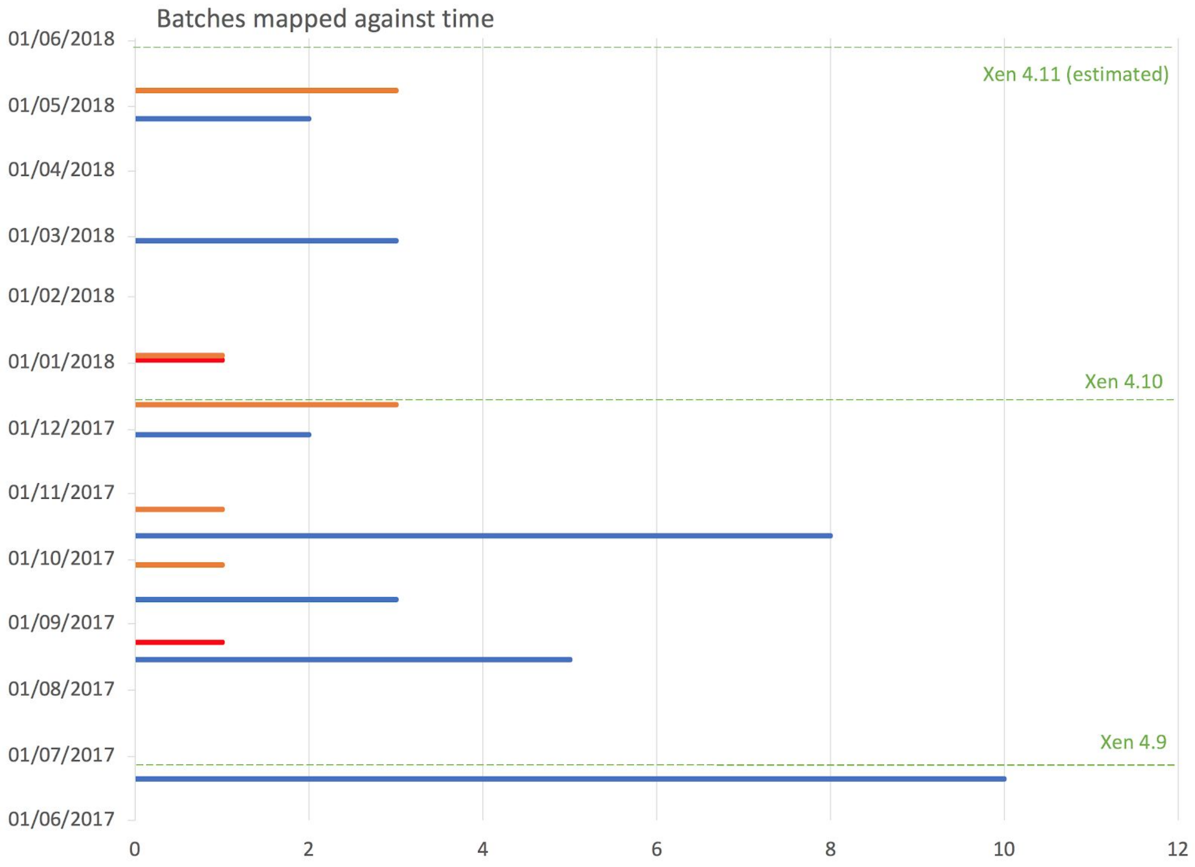
4/6 respondents highlighted lack of predictability as an issue (see 2.2.1 C). However, the survey has not specifically targeted end-users, which I believe would benefit from predictable publication dates of XSAs in particular if it were aligned with **Microsoft security updates are publicly released on the second Tuesday of each month.** This would also align the Xen Project with public release dates for issues which are disclosed in an industry wide fashion.

The natural solution to predictability would be to agree a time-table of XSAs and/or align with similar fixed release schedules of other projects. It is important to understand, that this has some implications:

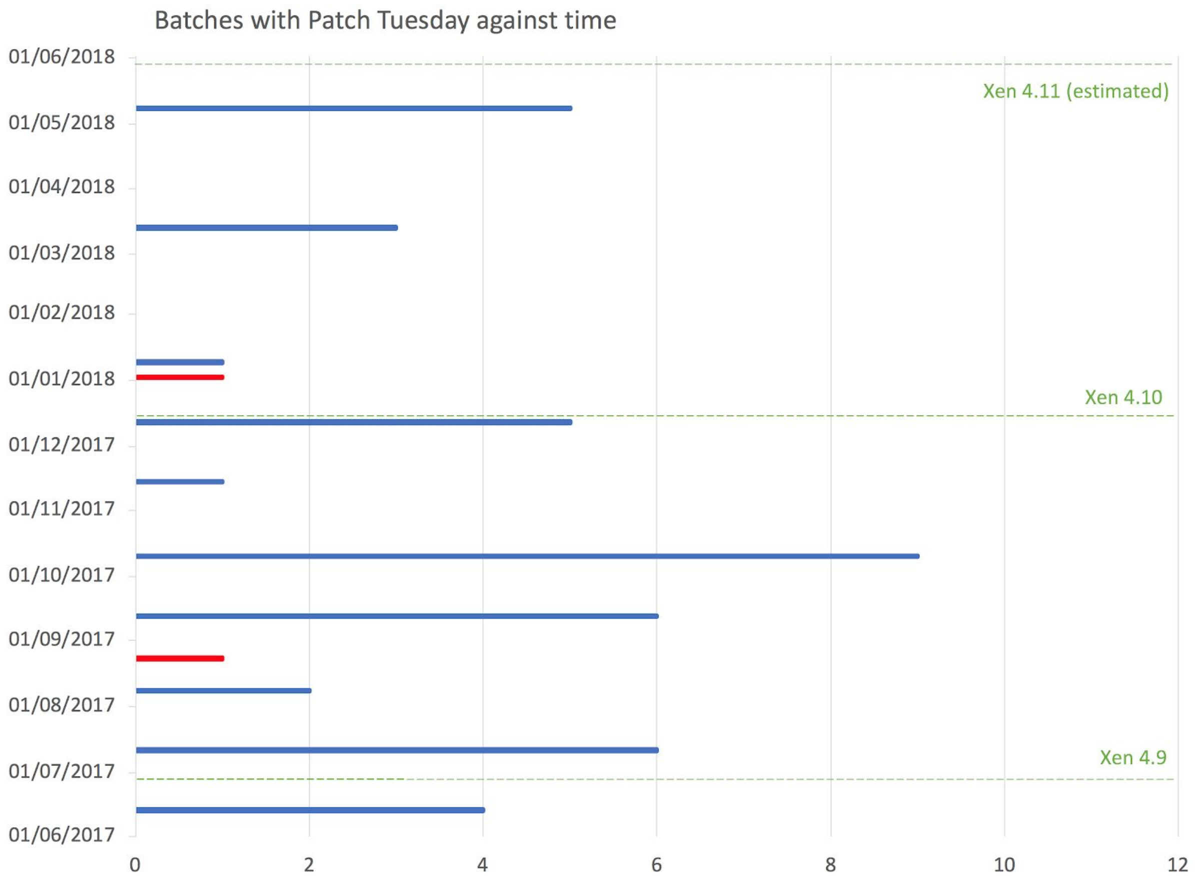
- First of all such a policy implies that the security team will have fixes for some XSAs and not immediately release them. Note that there has been discussion of such proposals in the past, and a number of security team members had objections ⇨ *I would counter this, that in practice over the last 1-2 years, some XSAs have not been released for a number of reasons such as batching as executed in the last 12 months, activities related to developing fixes and patching of issues.*
- Agreeing on monthly batch release date could lead to a maximum delay of 4.3 weeks from reporting to pre-disclosure and 6.3 weeks from reporting to public disclosure. This may lead to some criticism ⇨ *I would counter that in practice fixes for most XSAs take between 2-4 weeks to prepare: in particular if several are worked on in parallel.*
- Batching does not require changes to the 2 weeks of the pre-disclosure period

Note that no respondent has highlighted the time between a fix being available to the pre-disclosure date as an issue: admittedly. The length of a pre-disclosure period has however been raised by several as an issue.

To illustrate the impact of a batching policy around a fixed day, I have taken last year's data (below) and modeled for **publicly releasing batches of XSAs on the second Tuesday of each month**.



Note that lines marked in red, were XSA where we had no control over the publication date. Lines marked in orange, we could have batched. This diagram shows the impact of applying a fixed day XSA release policy, without changing XSAs we don't have control over.



The June/July 2017 column is interesting: looking at the data the 10 strong batch would have split into two in this case.

Besides the issue of a delay between a fix being available and pre-disclosure, there could also be issues negotiating a pre-release date / public release date with the discoverer of an issue:

1. In most cases, issues are discovered by committers ➪ *there should be no issue*
2. In most other cases discovers follow the recommendations of the security team ➪ *no issue*
3. In rare cases the discoverer will not follow recommendations ➪ *don't batch / adjust batch (see below)*
4. We have no control over a date ➪ *don't batch / adjust batch (see below)*

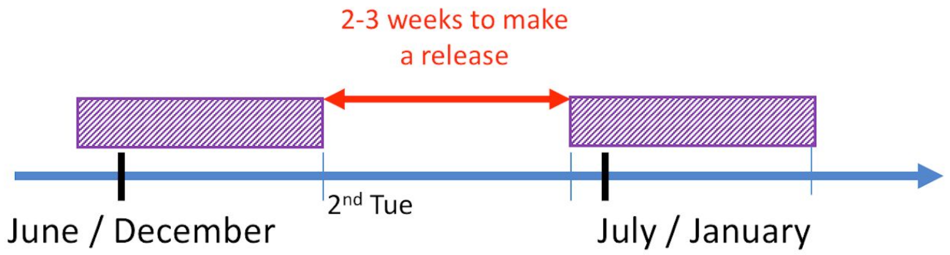
3 is relatively rare. However in general, we should retain the discretion that the Security Team has today, to allow for situations such as

- The capability to roll an XSA into the next batch (see R2)
- The capability to pro-actively extend the pre-disclosure period for individual XSAs by one or two days: let's say the discoverer does not want his XSA (let's call it XSA1) pre-released on a Tuesday, but insists on doing so on a Monday. In such a case, discretion could be used to to pre-release XSA1 on a Tue with a pre-disclosure period of 1 extra day. The other XSAs are pre-disclosed on the next day. However all XSAs are publicly disclosed on separate days.

We would should probably make the discretionary capabilities of the security team clearer than it is in today's process.

Impact on creating Xen Project releases:

The following graph shows the impact on Xen Project releases.



A fixed release schedule should make scheduling final release dates much easier: depending on the month, there are 2-3 weeks between pre-disclosure periods. Typically it takes slightly less than a week to cut a release if there are no issues, and two if there are. This fits well into the time period available.

However, the December release is potentially problematic as typically the last date to publish an effective press release is the second Thursday before XMas Eve. In most cases this would only give us 2-3 days to make a release.

There are two ways around this

- Release on the 1st Tue of each month, but this will cause problems with the January batch
- Change the release cycle from June/December to May/November

Recommendation:

The downside of releasing batches of XSAs once a month seems generally fairly low and appears to have big benefits for end-users. Recommended pe-disclosure should remain unchanged at 2 weeks, as there would be no increased risk in pre-disclosure members leaking information. Managing expectations of discoverers of security issues should become easier and planning releases also. Based on historical data, there should not be a huge impact on batch sizes: in general these would be similar to batch sizes we published in the last 12 months.

Aligning with Patch Tuesday (2nd Tue of each months) makes the most sense, but has the drawback that a pre-disclosure cycle would almost always start during the prime XMas holidays.

I would also recommend to give the Security team some discretion with regards to the process for exceptional circumstances: but these would need to be discussed and agreed.

3.2. WORKFLOW OR TOOLS RELATED

This section contains a number of recommendations

2.2.2 A. XSA Re-issues

At this stage it is not clear how much of an issue this is. If this issue is deemed important enough, I can write a little script that will extract some information from xsa.git and provide a more detailed analysis.

2.2.2 B. XSAs without CVE numbers

> We are already resolving this and have completed all steps to become a CNA: the application has been made to DWF and we are waiting for a reply.

2.2.2 C. Livepatch creation

The Security Team currently only considers live patchability informally. 7/10 respondents would welcome that this would be formalized.

> This is unlikely to happen, as such an activity would increase the workload of the Security Team significantly. In addition, at least in the last 6-9 months no issues related to live patching were reported via security-discuss@. A single issue was reported, which turned out to be a configuration issue by the reporting vendor and had nothing to do with the patches itself.

2.2.2 D. Inconsistent Meta Data and XSA prerequisites

> A discussion on xen-devel@ about the completeness of the meta-data API should be started. We should probably document the API.

2.2.2 E. No XSA update number in email subject

> This appears trivial and should be fixed (maybe we need a TODO list).

2.2.3 A. Tedious to identify which XSAs apply

> Do not address directly, but roll into 2.2.3 B, as these appear to be related.

2.2.3 B. Git baseline of patches

> I was originally going to recommend to not address, however this seems to affect most distro package maintainers (2 of which gave feedback on this issue). Talking to other package maintainers or people interfacing with them since, indicates that this issue is more widespread.
>
> A discussion on xen-devel@ on how to possibly solve this

3.2. RELEASE CYCLE RELATED

A discussion about the release cycle cadence is due by now anyway to test whether what we introduced is working. Issue 2.2.4 B (Coordination with Xen Releases) is suitably addressed by R3.

> However a discussion about length of security support, which may include some releases with shorter security support lifetimes and possibly an LTS type model would be worthwhile as part of this discussion.