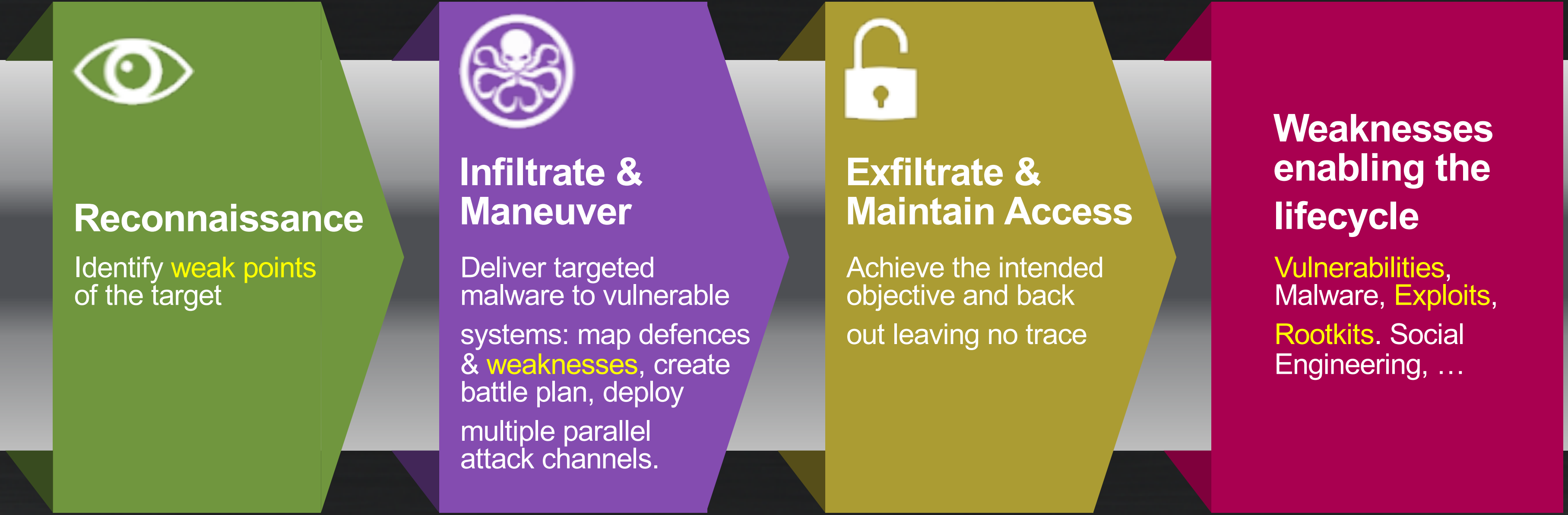
# Disclosure Policies in the World of Cloud: A Look Behind the Scenes

Lars Kurth
Community Manger, Xen Project
Chairman, Xen Project Advisory Board
Director, Open Source, Citrix

lars_kurth

# Setting the Scene: Definitions, Roles, …

# Simplified lifecycle of a cyber attack



### Reconnaissance

Identify weak points of the target

### Infiltrate & Maneuver

Deliver targeted malware to vulnerable systems: map defences & weaknesses, create battle plan, deploy multiple parallel attack channels.

### Exfiltrate & Maintain Access

Achieve the intended objective and back out leaving no trace

### Weaknesses enabling the lifecycle

Vulnerabilities, Malware, Exploits, Rootkits. Social Engineering, …

*Template designed by PresentationGO.com*

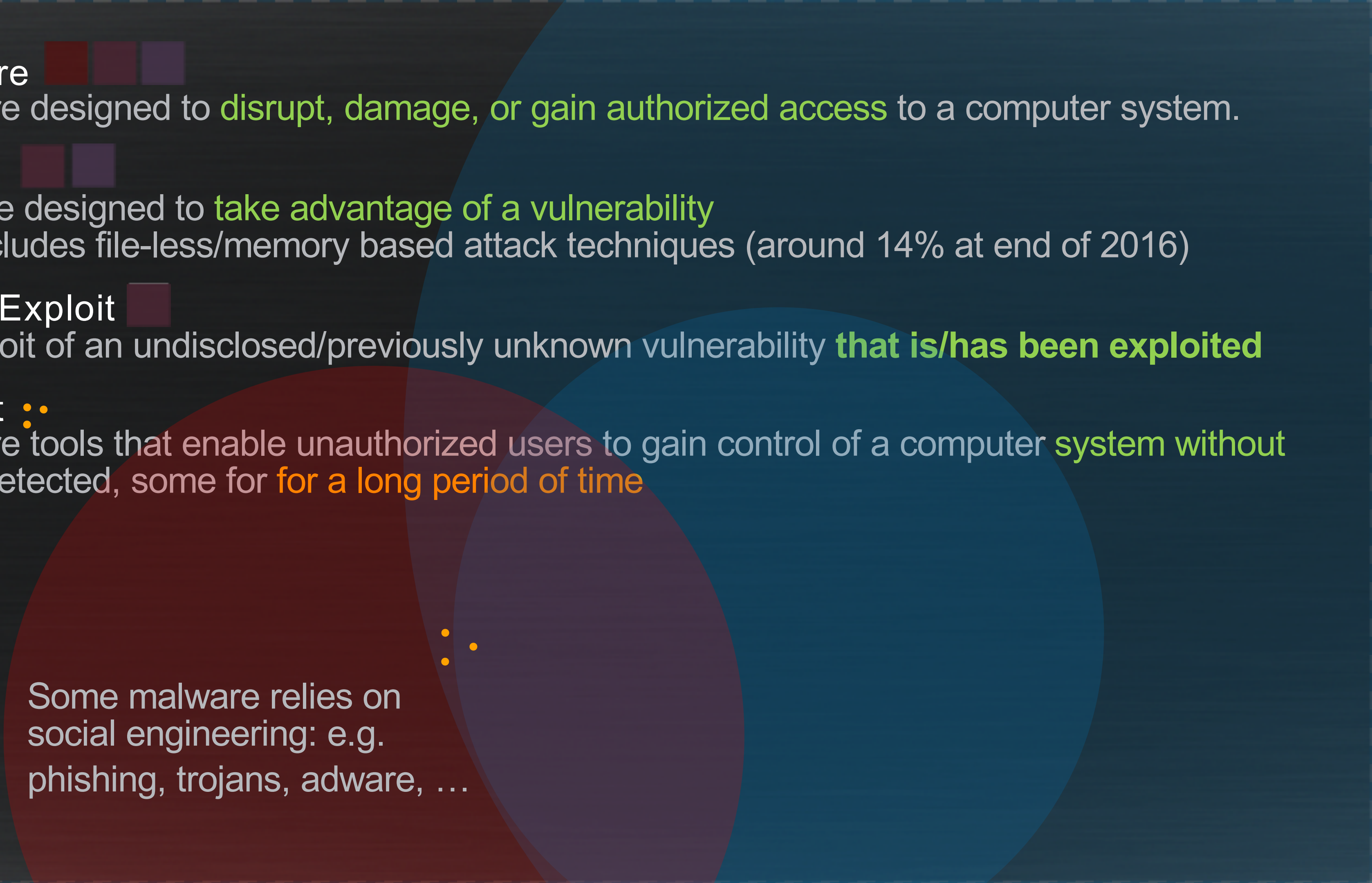
## Vulnerability

A weakness in the computational logic of software that, when exploited, results in a negative impact to confidentiality, integrity, OR availability of that software.

## Patch / Live Patch

A fix for a security vulnerability.
A live patch can be applied to a running system

Known vulnerabilities with patches



### Malware ■ ■ ■

Software designed to disrupt, damage, or gain authorized access to a computer system.

### Exploit ■ ■

Malware designed to take advantage of a vulnerability
This includes file-less/memory based attack techniques (around 14% at end of 2016)

### 0-Day Exploit ■

An exploit of an undisclosed/previously unknown vulnerability **that is/has been exploited**

### Rootkit ∴

Software tools that enable unauthorized users to gain control of a computer system without being detected, some for for a long period of time

Some malware relies on social engineering: e.g. phishing, trojans, adware, …

# CSIRTs

## (Computer|Product) Security (Incident) Response Team

| Reactive Services | Proactive Services | Security Quality Management |
|---|---|---|
| Alerts and Warnings<br>**Incident Handling:** analysis, response on site, response support, response coordination<br>**Vulnerability Handling:** Vulnerability analysis, response, response coordination<br>**Artefact Handling:** analysis, response, response coordination | Announcements<br>Technology Watch<br>Security Audits or Assessments<br>Configuration and Maintenance of Security Tools, Applications, and Infrastructures<br>Development of Security Tools<br>Intrusion Detection Services<br>Security-Related Information Dissemination<br>Vulnerability Bounty Programs | Risk Analysis<br>Business Continuity and Disaster Recovery Planning<br>Security Consulting<br>Awareness Building<br>Education/Training<br>Product Evaluation or Certification |

# CSIRT Vulnerability Handling: End-goal

## Products and Distros

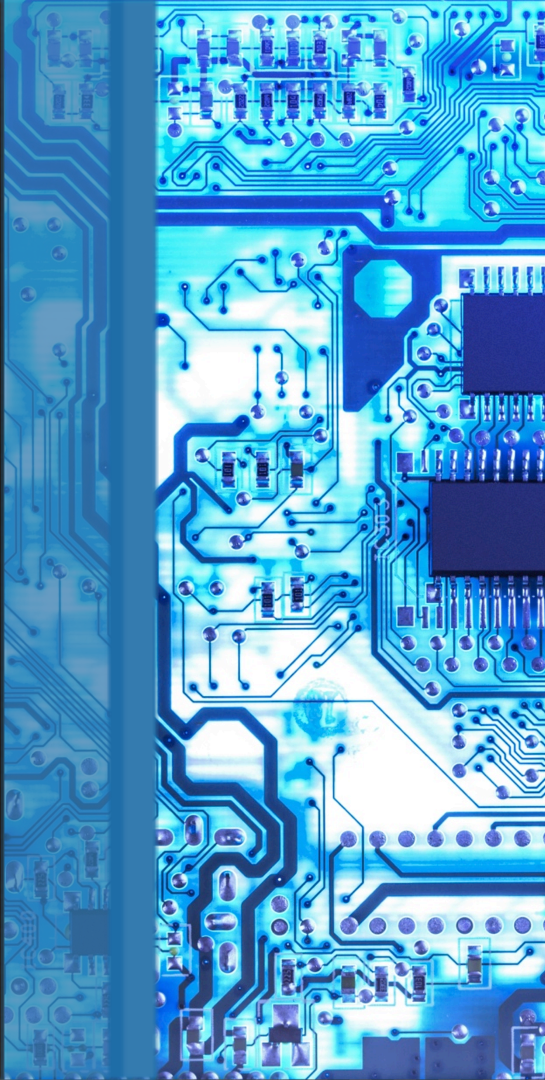
Ensure that a fix is available in **all product variants** that were affected by the vulnerability and ideally deployed by key customers at the time information of the security issue becomes public.

## Cloud Providers and other public facing services

Ensure that a fix is deployed on **all public facing hosts** that were affected by the vulnerability at the time information of a security issue becomes public.

# Patterns of Vulnerability Handling

**Standard:** ISO/IEC29147 ➔ ISO/IEC DIS 29147
https://www.first.org



# What we will look at

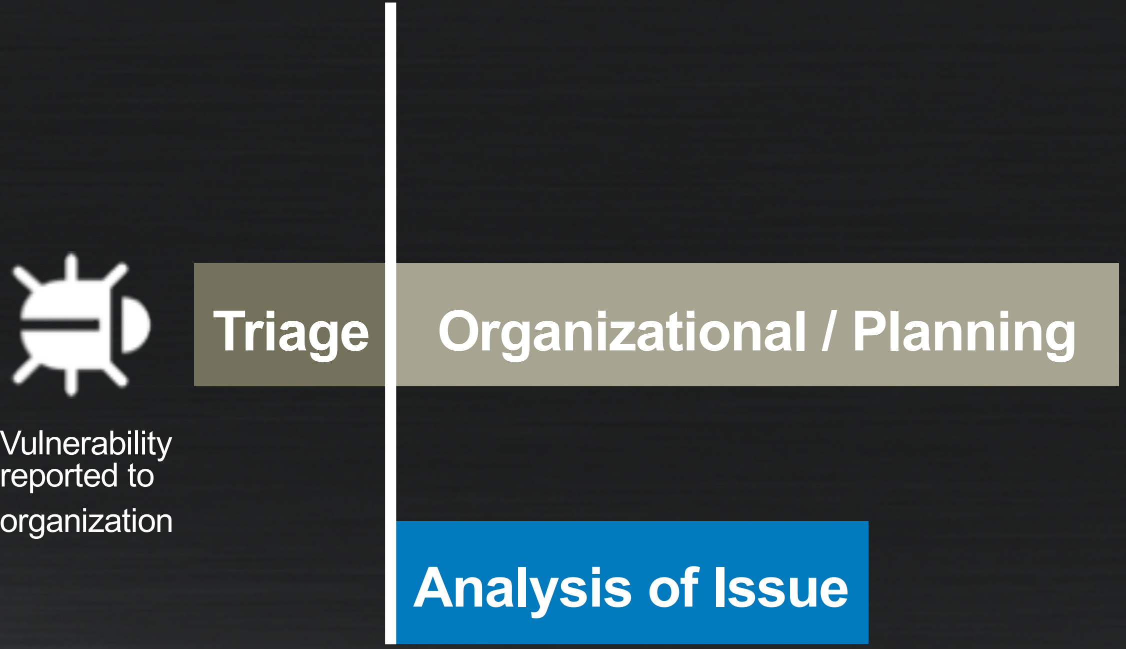
The overall process a CSIRT team goes through when they discover a vulnerability themselves or one is reported to them directly

In this case we assume that the CSIRT team has full control over all components in their system, which is normally only the case when the Software stack is proprietary.

The process an open source project's CSIRT team goes through (modelled on best practice as used by Linux distros, Xen Project and others)

How a commercial CSIRT team interacts with an open source team and how this constrains what the commercial team can do

# Vulnerability Fixing Pattern : Proprietary



Vulnerability reported to organization

Triage

Organizational / Planning

Analysis of Issue

Negotiate Disclosure Schedule

Draft issue description

Plan deployment: large organizations need to make sure that staff across impacted teams is available

Allocate CVE Number

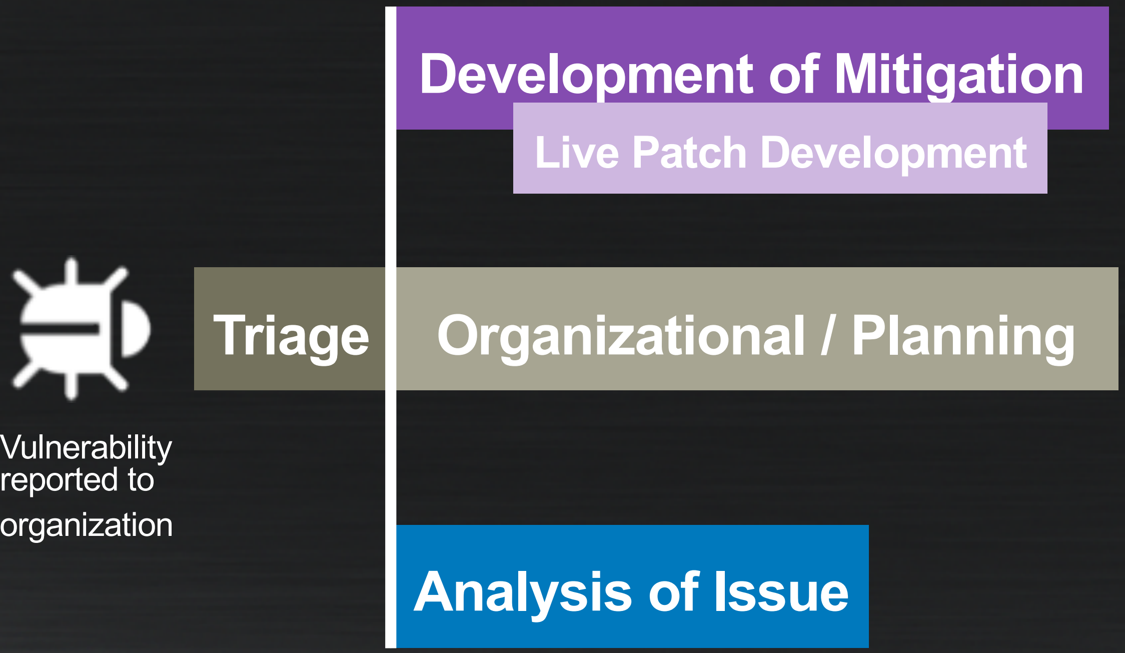
Issue description

Establish Impacted Releases / Customers

Create PoC / Test Case: Understand the issue, investigate its impact, enable test that issue is fixed

Impact assessment

# Vulnerability Fixing Pattern : Proprietary


Development of Mitigation
Live Patch Development
Triage
Organizational / Planning
Analysis of Issue
Vulnerability reported to organization

Develop fix

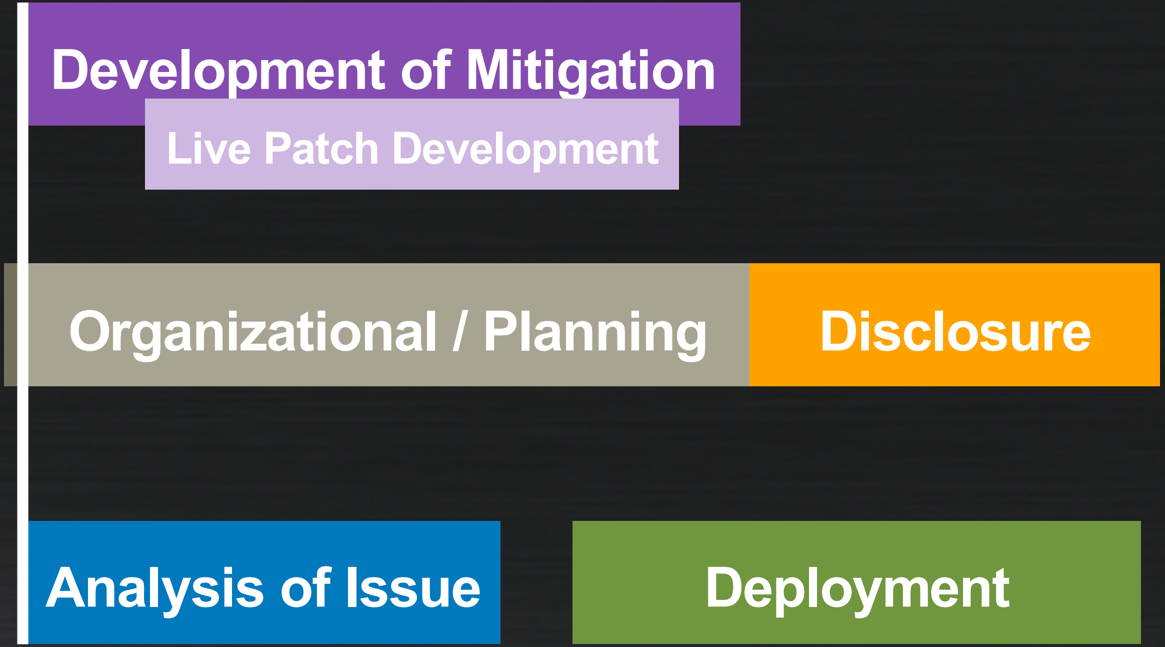
Q&A of fix
(formal proof that fix addresses the exploit)

Develop Live Patch

Backport fix/live patch to affected products

Per product validation

# Vulnerability Fixing Pattern : Proprietary



Disclosure to impacted FOSS project (if applicable)

Pre-disclosure to important customers typically under an NDA type agreement

Publish Documentation

Customer notification

Resource planning

Packaging (bundling of several fixes)

Q&A of packages against affected releases / customer types

Deployment to affected customers

# Observations

This model assumes that the managing CSIRT organization is only constrained by the wishes of the issue reporter

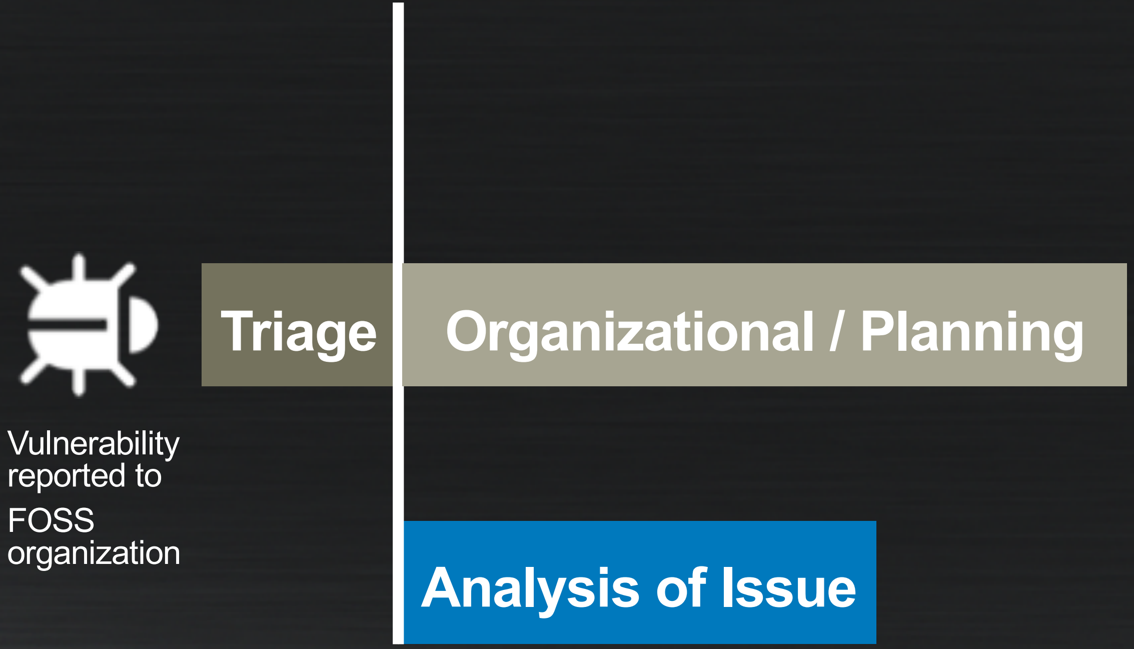
- This gives such an organization the freedom to for example pre-disclose information to “important customers”
- And a maximum of freedom on any implementation schedule, assuming it can convince the discoverer of an issue
- The possibility to not disclose any issues publicly: aka quietly fix them

When open source components are used, significant constraints are put on what the CSIRT team can do

A similar situation also arises in situations where multiple parties are impacted (e.g. Spectre, Meltdown)

# FOSS Project: Pre-disclosure

(Applies to Xen, Linux Distros, OpenStack, ...)


Vulnerability reported to FOSS organization
Triage
Organizational / Planning
Analysis of Issue

Negotiate Disclosure Schedule

Draft issue description

~~Plan deployment: large organizations need to make sure that staff across impacted teams is available~~

Allocate CVE Number

Issue description

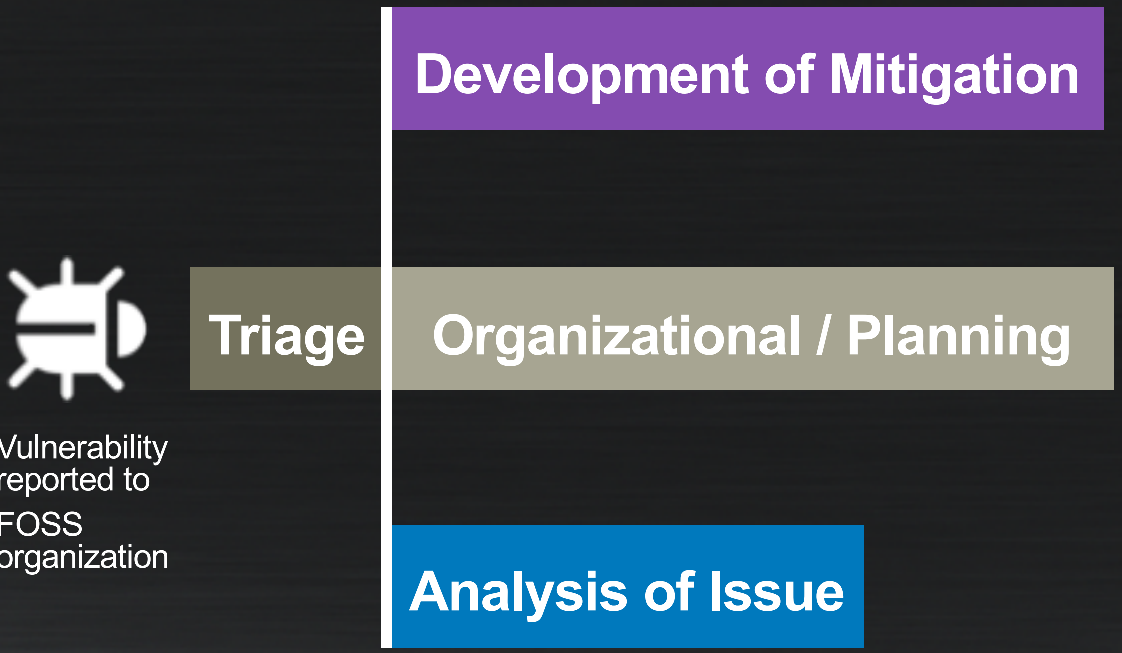
Establish Impacted Releases

Create Test Case: Understand the issue, investigate its impact, enable test that issue is fixed

Impact assessment

# FOSS Project: Pre-disclosure

(Applies to Xen, Openwall distros, OpenStack, ...)



Vulnerability reported to FOSS organization

Triage

Development of Mitigation

Organizational / Planning

Analysis of Issue

Develop fix

Q&A of fix

~~(formal proof that fix addresses the exploit)~~

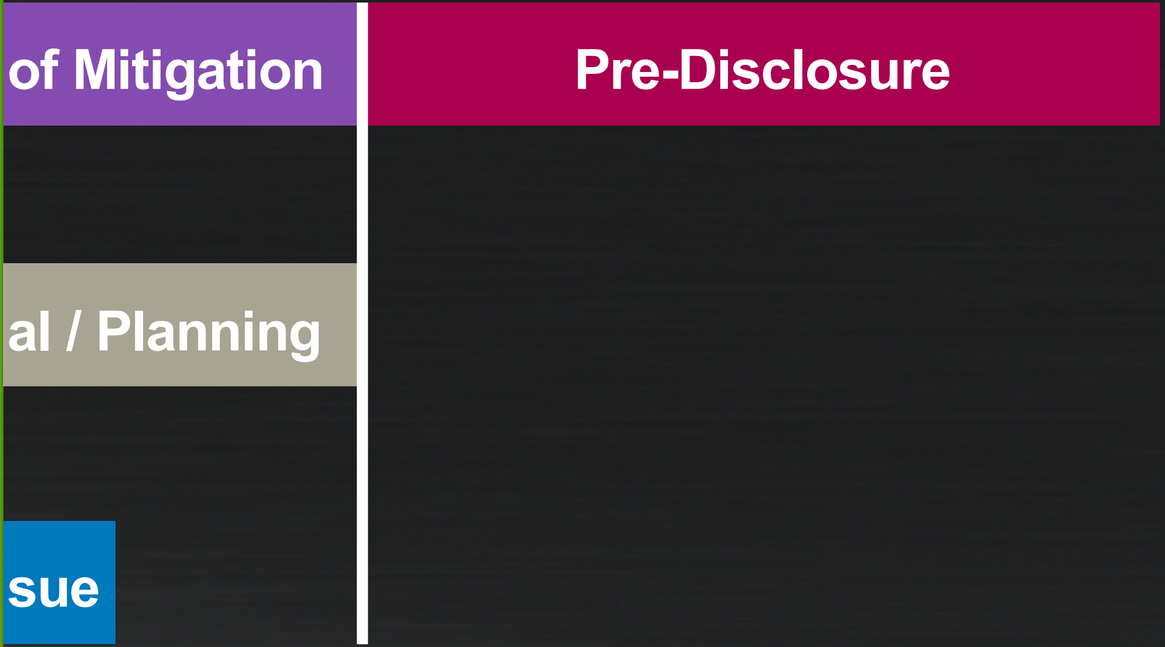
~~Develop Live Patch~~

Backport fix~~/live patch~~ to affected products

Per product validation

# FOSS Project: Pre-disclosure

(Applies to Xen, Openwall distros, OpenStack, ...)



of Mitigation

Pre-Disclosure

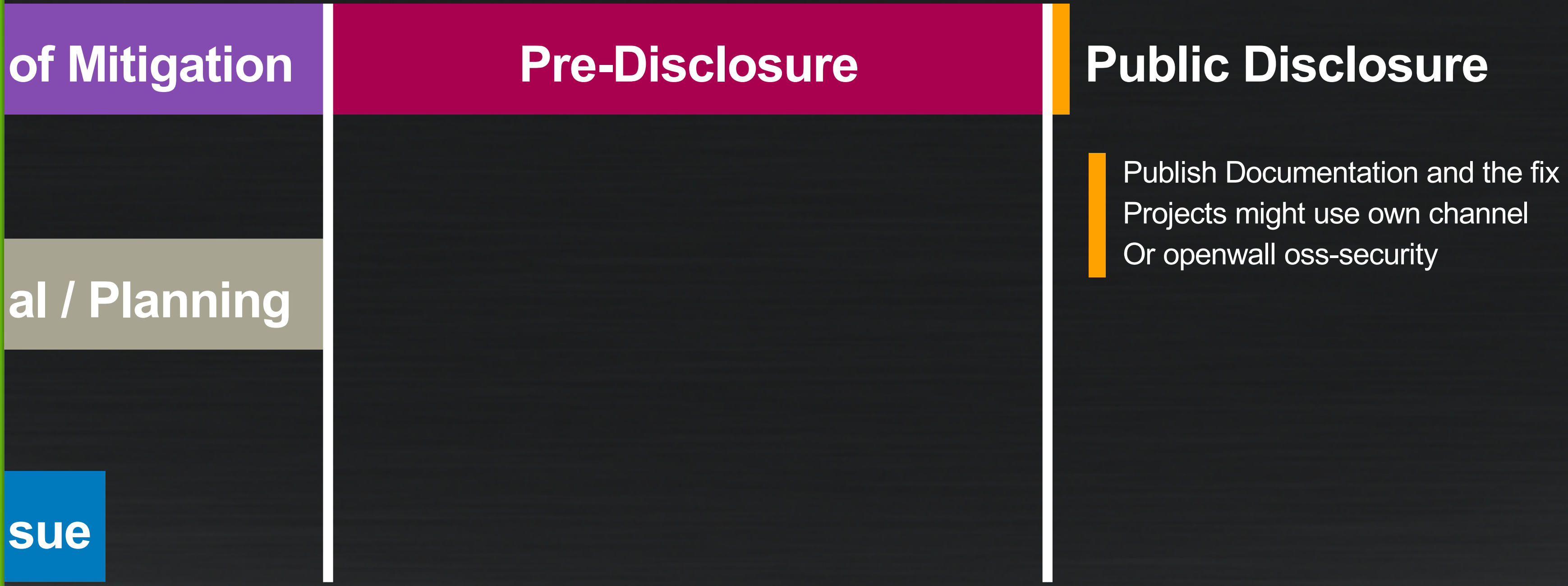
al / Planning

sue

- Pre-disclosure to qualifying downstream CSIRTS
- Typically fixed time-table (e.g. 2 weeks)
- Restricts what CSIRT can do (e.g. no deployment of fix during embargo)
- Collaboration between qualifying CSIRTs can be restricted

# FOSS Project: Pre-disclosure

(Applies to Xen, Openwall distros, OpenStack, ...)



of Mitigation

Pre-Disclosure

Public Disclosure

al / Planning

sue

- Publish Documentation and the fix
- Projects might use own channel
- Or openwall oss-security

# CSIRT Vulnerability Handling: End-goal

## Products and Distros

Ensure that a fix is available in **all product variants** that were affected by the vulnerability ~~and ideally deployed by key customers~~ at the time information of the security issue becomes public.

## Cloud Providers and other public facing services

Ensure that a fix is deployed on **all public facing hosts** that were affected by the vulnerability at the time information of a security issue becomes public.

Whether these goals are achievable, depends on the FOSS Vulnerability Management Process

# The CSIRT perspective

## Pre-Disclosure Period

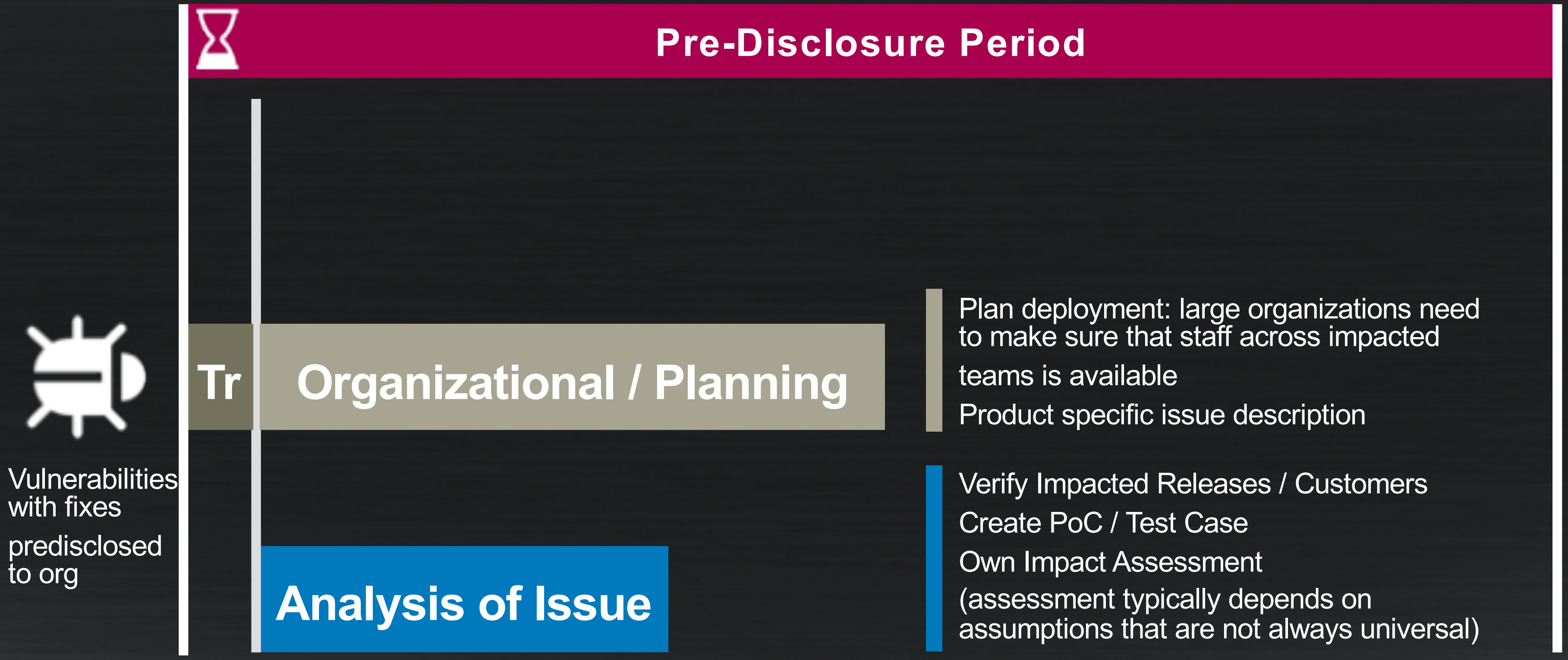
Vulnerabilities with fixes predisclosed to org

Tr

Will usually perform triage, because OSS based product/service typically is modified (patch queue) and/or use an unusual configuration.

May choose not to fix a lower severity issue immediately.

# The CSIRT perspective



Vulnerabilities with fixes predisclosed to org

Plan deployment: large organizations need to make sure that staff across impacted teams is available

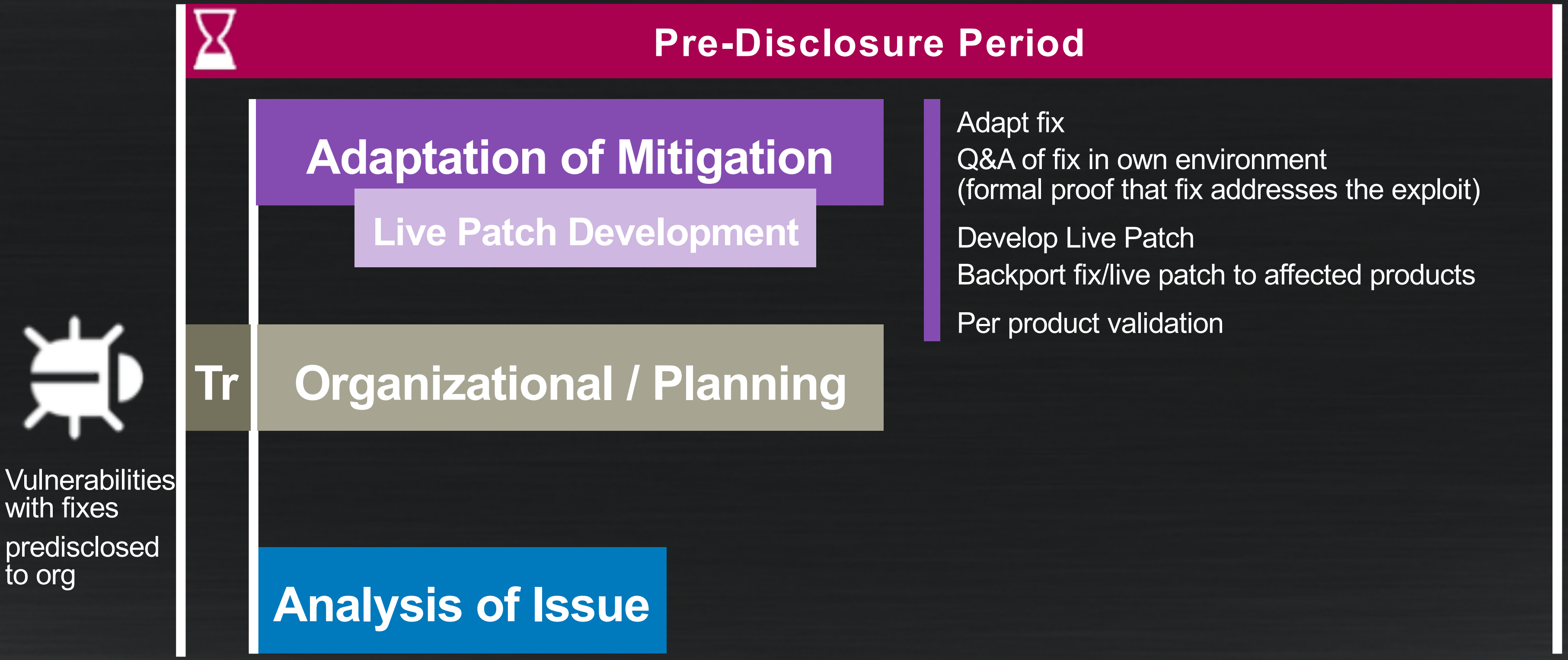
Product specific issue description

Verify Impacted Releases / Customers

Create PoC / Test Case

Own Impact Assessment (assessment typically depends on assumptions that are not always universal)

# The CSIRT perspective



Pre-Disclosure Period

Adaptation of Mitigation

Live Patch Development

Tr

Organizational / Planning

Analysis of Issue

Vulnerabilities with fixes

predisclosed to org

Adapt fix

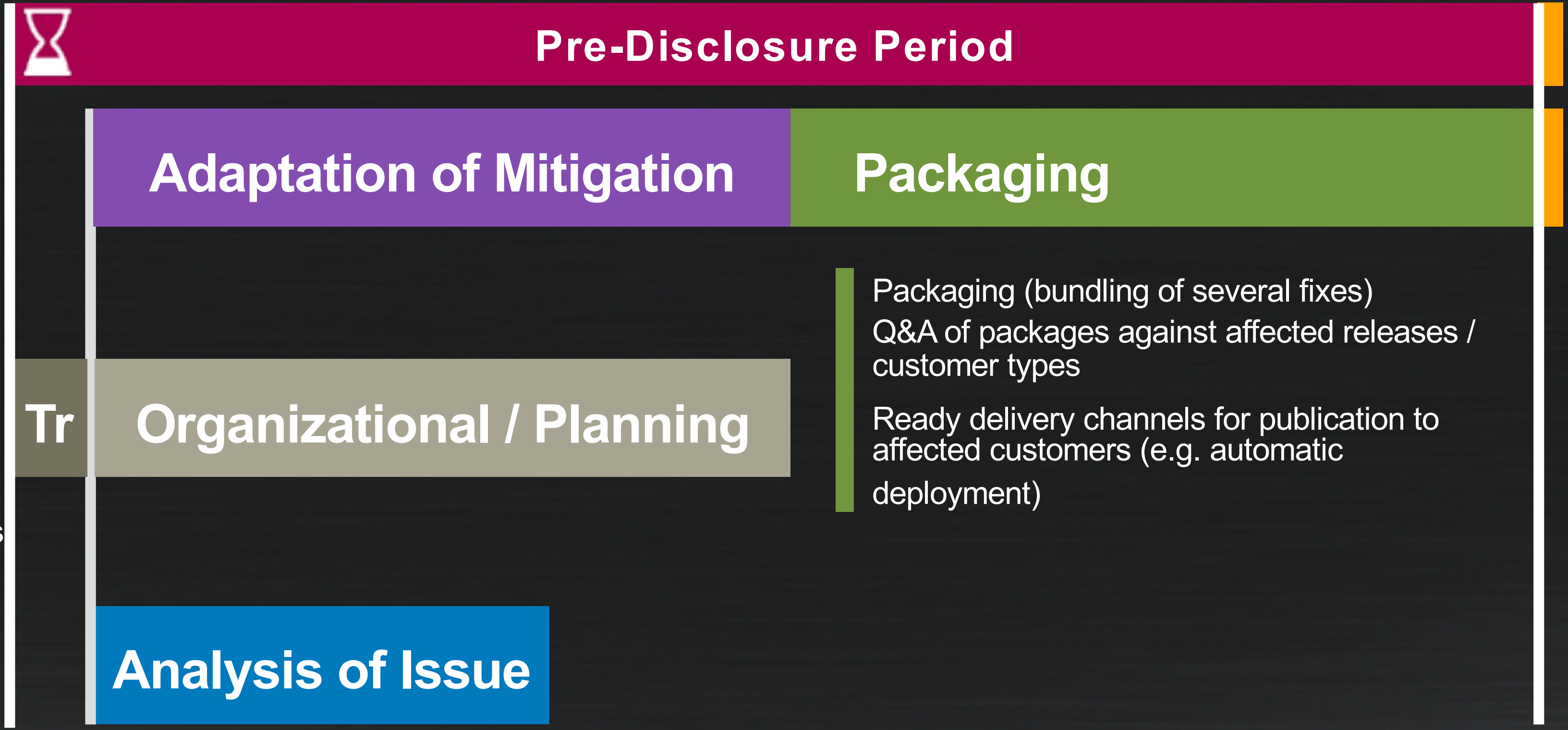
Q&A of fix in own environment
(formal proof that fix addresses the exploit)

Develop Live Patch

Backport fix/live patch to affected products

Per product validation

# The CSIRT perspective: Product Company



Pre-Disclosure Period

Adaptation of Mitigation

Packaging

Tr

Organizational / Planning

Analysis of Issue

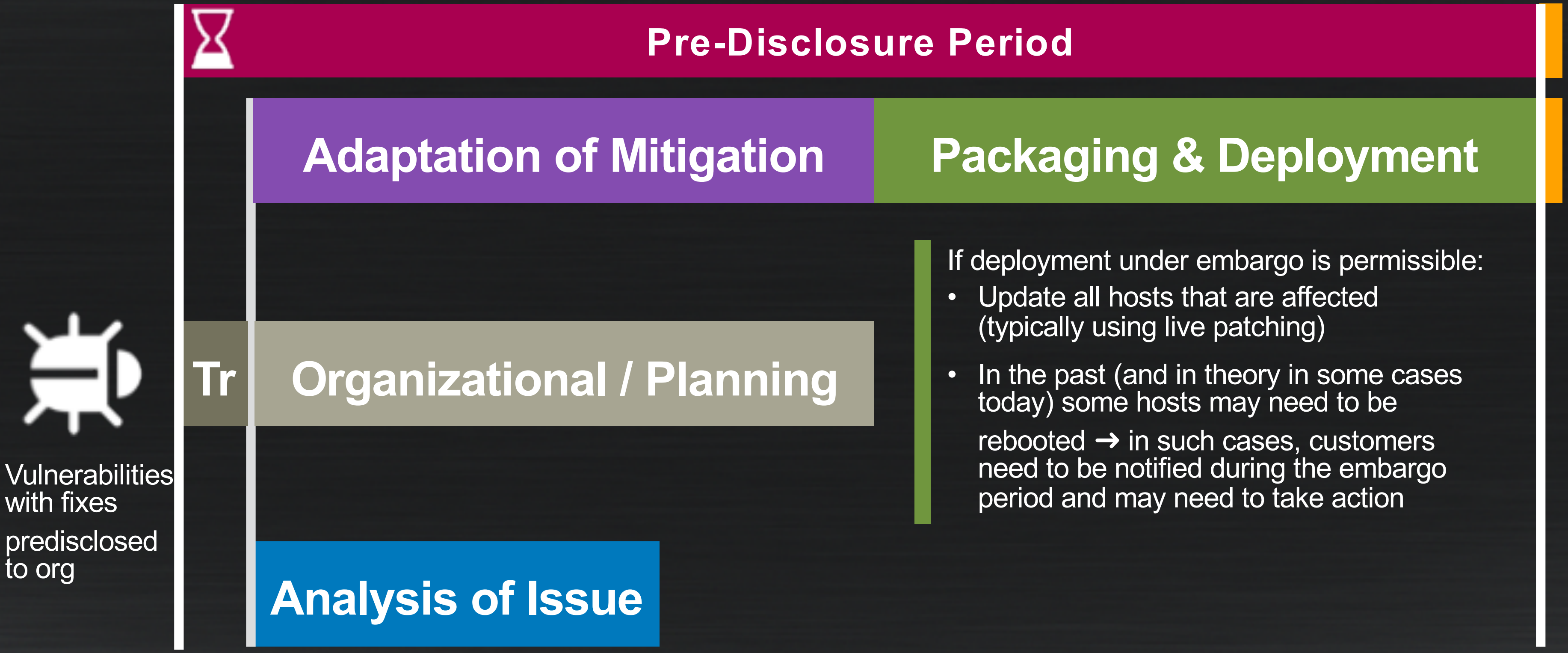
Vulnerabilities with fixes predisclosed to org

Packaging (bundling of several fixes)

Q&A of packages against affected releases / customer types

Ready delivery channels for publication to affected customers (e.g. automatic deployment)

# The CSIRT perspective: Service Provider



If deployment under embargo is permissible:

- Update all hosts that are affected (typically using live patching)
- In the past (and in theory in some cases today) some hosts may need to be rebooted ➔ in such cases, customers need to be notified during the embargo period and may need to take action

# Restrictions on CSIRTs

**Does it qualify to be on a FOSS project's pre-disclosure list?**

**Timetable:**

- When is an issue pre-disclosed?
- How many bugs are pre-disclosed at once?
- How long is the disclosure period?

**What can be done with the privileged information from the list:**

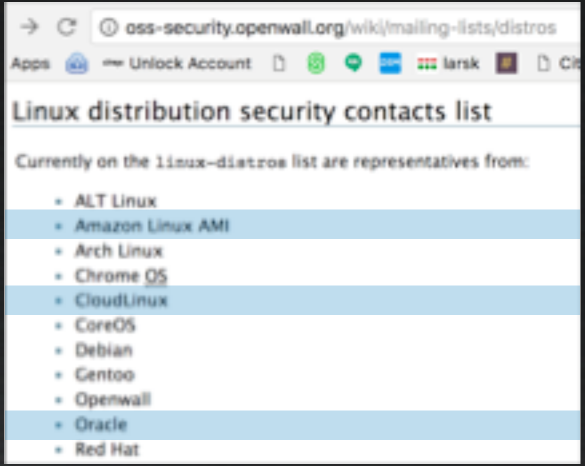
- Deployment under embargo?
- Communicating to affected customers that they need to take action (e.g. scheduled reboots)
- Collaboration with other pre-disclosure list members

**Technical:**

- FOSS projects may do less than the CSIRT would do normally
- E.g. no live patches, no PoCs, no severity assessment, backports to older versions, …

# Service Providers

| | linux-distros @openwall.org | Xen Project | OpenStack |
|---|---|---|---|
| **Orgs qualifying for pre-disclosure** | FOSS & Commercial **Distros** | Distros, products, **public services, large users** | Commercial & FOSS **Downstreams** |



Linux distribution security contacts list

Currently on the `linux-distros` list are representatives from:

- ALT Linux
- Amazon Linux AMI
- Arch Linux
- Chrome OS
- CloudLinux
- CoreOS
- Debian
- Gentoo
- Openwall
- Oracle
- Red Hat

*Service Providers that are on Openwall Distros*

This means that cloud and hosting providers **generally do not know** of Linux related security issues under pre-disclosure and cannot plan for, fix or deploy fixes until they become public

# Service Providers & Xen Project

## Policy:

xenproject.org/security-policy.html

## Pre-disclosure list membership:

**FOSS projects/distros:** 7

**Products based on Xen:** 11

**Public Service Providers/Large Users:** 57

*That is a large pre-disclosure list, but we haven't had a single leakage since we made the list inclusive 4 years ago*

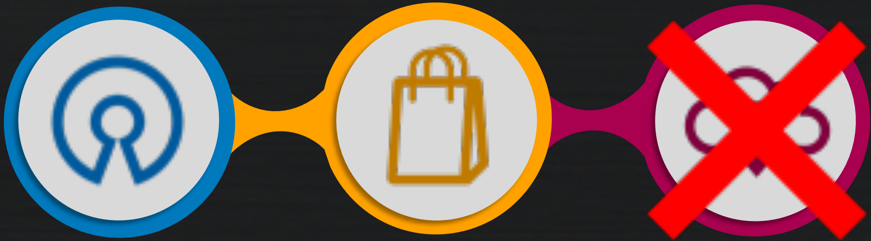
# Service Providers

| | linux-distros @openwall.org | Xen Project | OpenStack |
|---|---|---|---|
| **Allow updating of public facing systems during embargo** | Only in **rare and extreme cases** | **Yes**, unless discoverer objects | Not documented |

Enables hosting and cloud providers to **update their systems during the pre-disclosure period**

In addition, Xen allows pre-disclosure member to make non-specific announcements to customers, aka "*please reboot X by date because of an upcoming Xen Project security issue*"

# Customers of Commercial Products



An open source project, of which a commercial variant is delivered to a service provider

Pre-disclosure does not allow to inform customers of the issue

*Template adapted from one designed by PresentationGO.com*

# Customers of Commercial Products

Pre-disclosure list members are allowed to securely share information, including binaries

Service provider has capability to deploy fixes under embargo

This is also useful for live patching and back-ports for releases out of security support

Service providers, even those using commercial derivatives, qualify to be on the pre-disclosure list

Service provider has visibility and can plan deployment

# Timing

**Openwall Distros:** unspecified

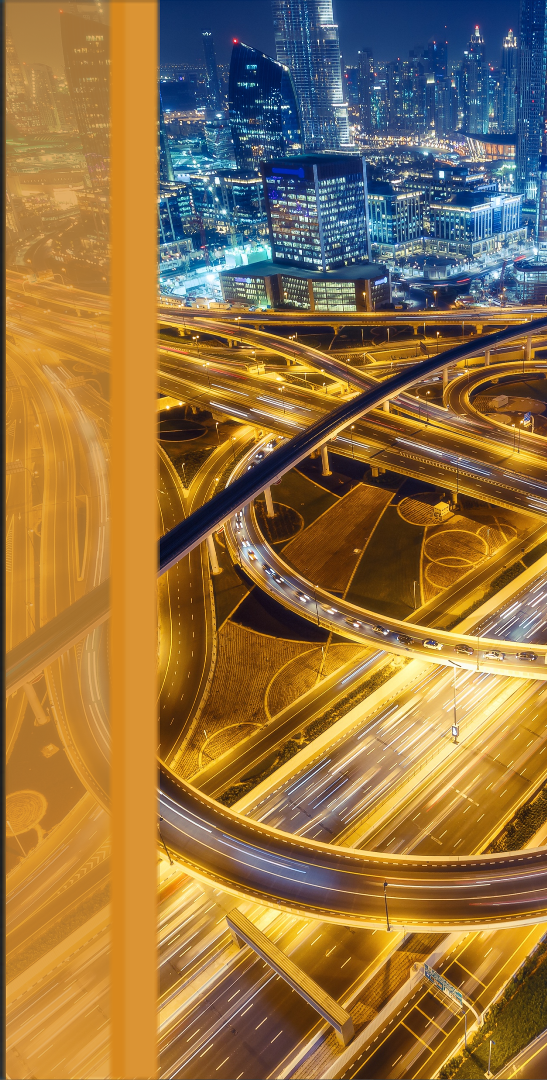
**Openstack:** 3-5 days

- Generally too short for deployment at scale

**Xen Project:** 2 weeks, usually in batches of several issues

- 2 weeks have proven to be enough time for most organizations to handle 5-6 issues in one go (assumes live patching)
- The project is currently reviewing its security process: see https://lists.xenproject.org/archives/html/xen-devel/2018-05/threads.html#01127 looking at
  - Batching of Issues
  - Workload batches have on consumers of security issues
  - Making publication of security issues more predictable

# Even more complexity: Multi-vendor vulnerabilities and disclosure



# Examples

**Intel SYSRET** (2012)
**Heartbleed** (2014)

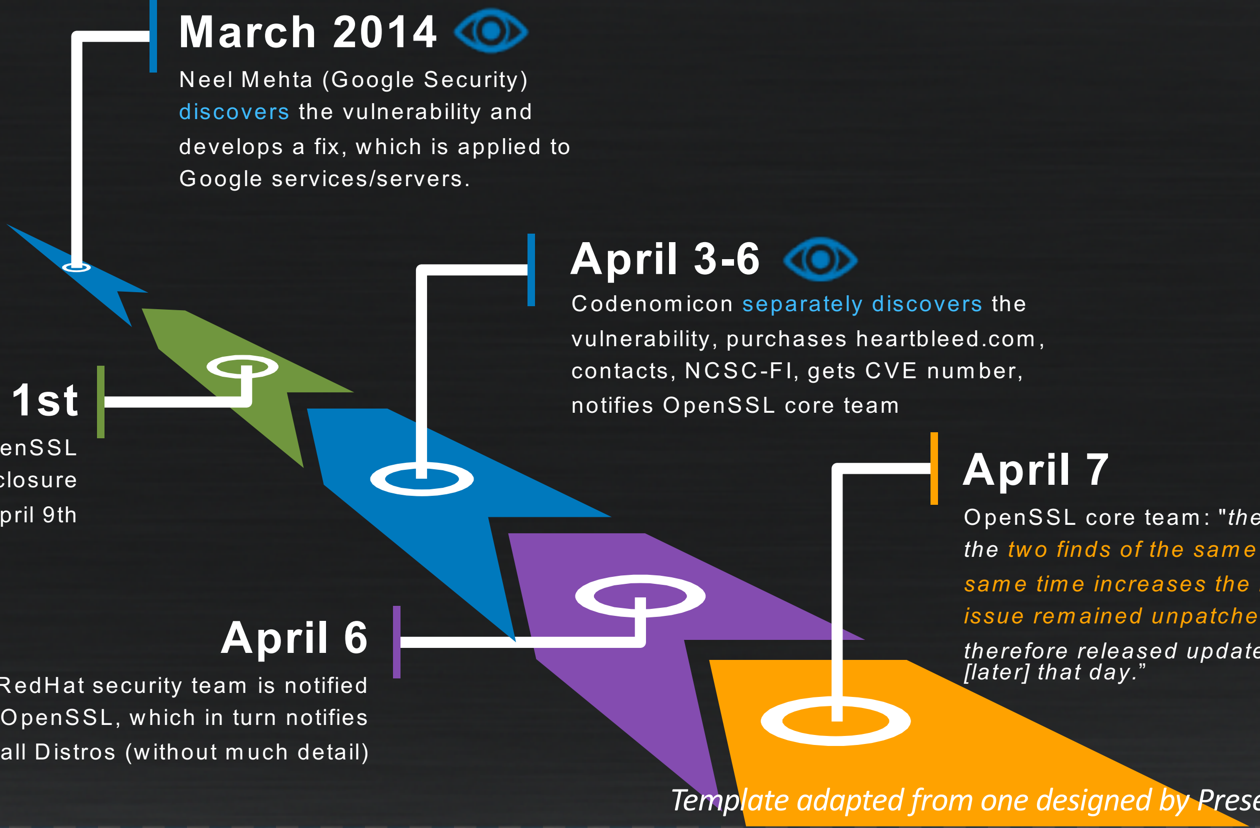
**Shellshock** (2014)
**Meltdown & Spectre** (2018)

**CVE-2018-8897** (2018)

**Speculative Store Bypass** (2018)
**Speculative register leakage from lazy FPU context switching** (2018)

# Heartbleed: Who knew what when

www.smh.com.au/technology/heartbleed-disclosure-timeline-who-knew-what-and-when-20140414-zqurk.html



## March 2014

Neel Mehta (Google Security) discovers the vulnerability and develops a fix, which is applied to Google services/servers.

## April 1st

Google Security notifies OpenSSL core team. Plan for public disclosure on April 9th

## April 3-6

Codenomicon separately discovers the vulnerability, purchases heartbleed.com, contacts, NCSC-FI, gets CVE number, notifies OpenSSL core team

## April 6

The RedHat security team is notified by OpenSSL, which in turn notifies OpenWall Distros (without much detail)

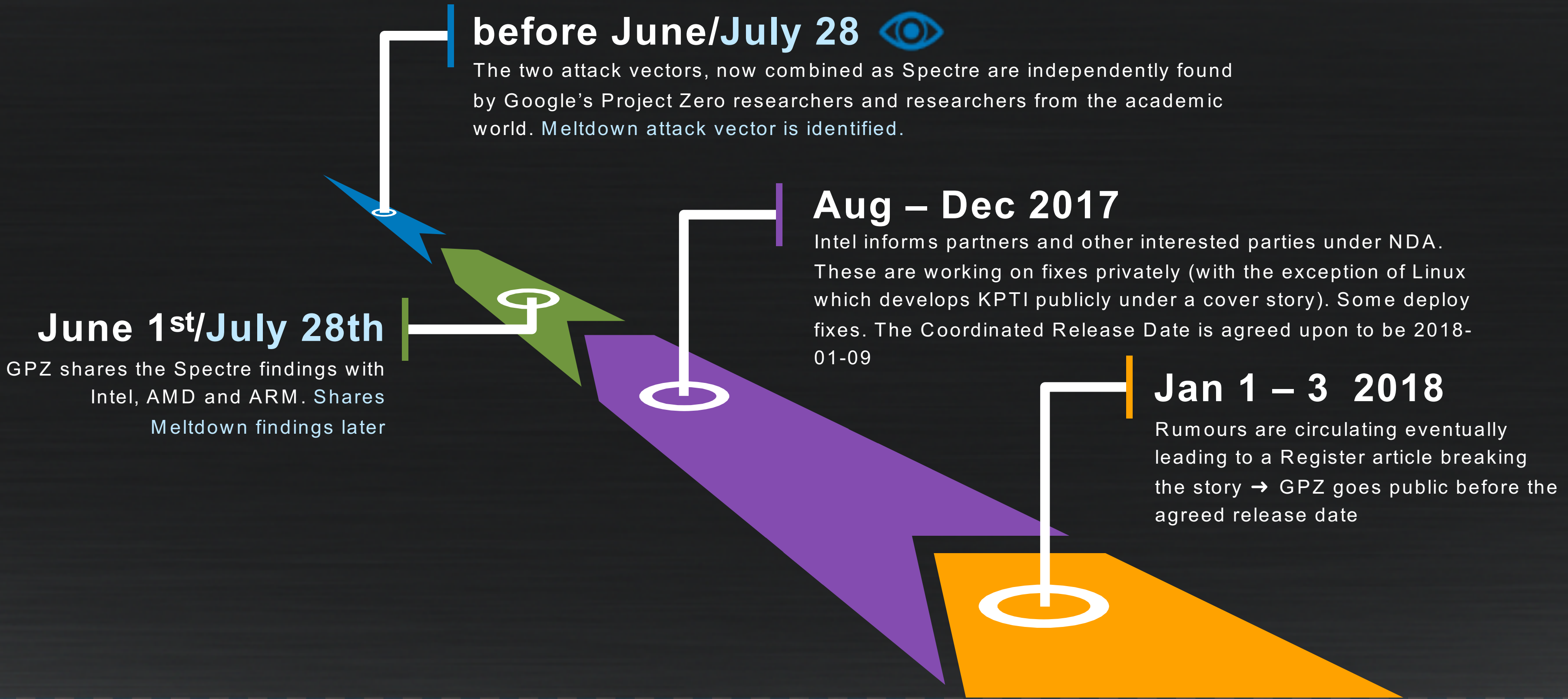
## April 7

OpenSSL core team: "*the coincidence of the two finds of the same issue at the same time increases the risk while this issue remained unpatched*. OpenSSL *therefore released updated packages [later] that day.*"

*Template adapted from one designed by PresentationGO.com*

# Spectre/Meltdown: Who knew what when

Based on www.theverge.com/2018/1/11/16878670/meltdown-spectre-disclosure-embargo-google-microsoft-linux & plus.google.com/+jwildeboer/posts/jj6a9JUaovP



### before June/July 28

The two attack vectors, now combined as Spectre are independently found by Google's Project Zero researchers and researchers from the academic world. Meltdown attack vector is identified.

### June 1st/July 28th

GPZ shares the Spectre findings with Intel, AMD and ARM. Shares Meltdown findings later

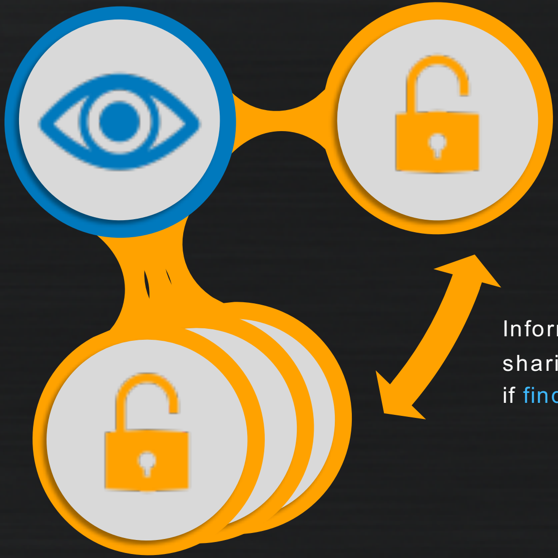
### Aug – Dec 2017

Intel informs partners and other interested parties under NDA. These are working on fixes privately (with the exception of Linux which develops KPTI publicly under a cover story). Some deploy fixes. The Coordinated Release Date is agreed upon to be 2018-01-09

### Jan 1 – 3 2018

Rumours are circulating eventually leading to a Register article breaking the story → GPZ goes public before the agreed release date

# Common Patterns



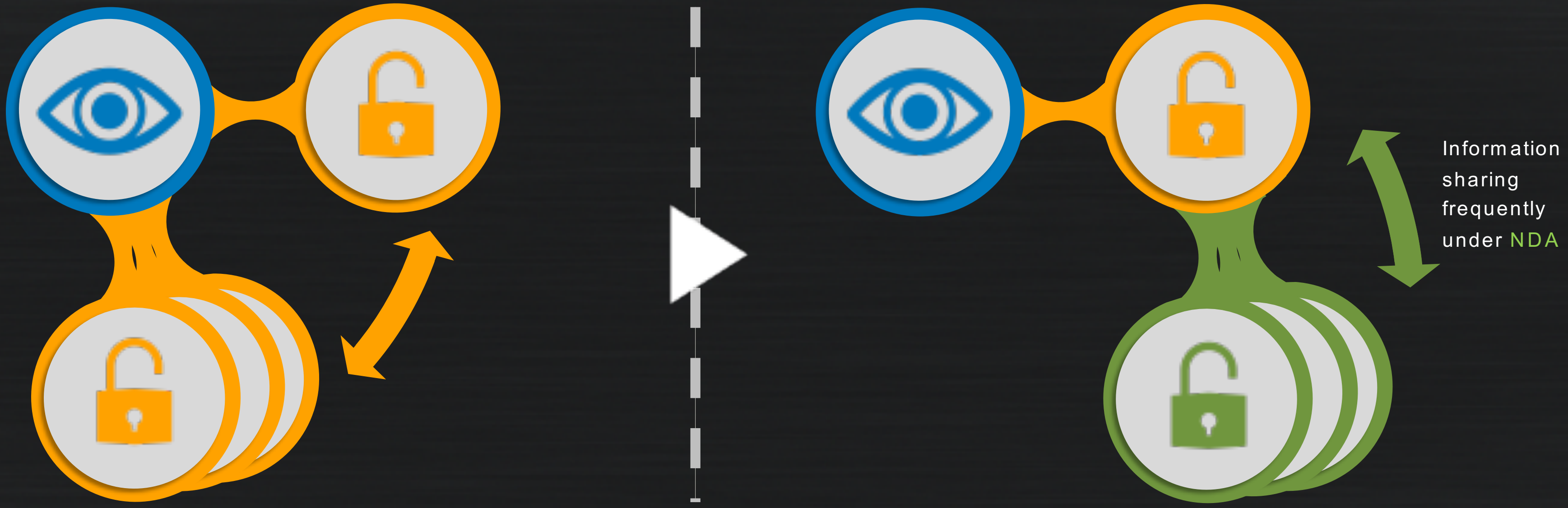
Typically, during initial discussion of the issue, it becomes apparent that this is a multi-vendor-issue

Information sharing by trust if finder permits

The finder of the issue stays in control over the time-table, reaching out to other impacted CSIRTS
The finder can insist on other constraints impacting CSIRTs

Unless the discovery comes through a bug bounty program
→ the discoverer essentially sells rights to the CSIRT managing the program

# Common Patterns



**Coordinator:** Frequently the discoverers **delegate managing the disclosure to a preferred CSIRT**, which then acts as a front-end for other CSIRTS

Typically coordinators (if commercial) will share information with other CSIRTS under NDA to cover themselves against litigation and prevent disclosure of issues

# Conclusion

# Lessons

Security Disclosure between FOSS and vendors can be complex

- Most project's have opaque practices: Linux Kernel, Qemu, Cloud Foundry, …
- There are exceptions: Linux Distros, Drupal, Mozilla, Node.js, OpenStack, Xen Project
- Some standardization in the LF eco-system would be helpful

  → across FOSS the range of different practices is even wider
- However this will be hard: it took the Xen Project 5 years of difficult iterative public negotiations to get to where we are today

Multi-party Disclosure is even harder, but becoming more common

- Emerging standards and Best Practices amongst CSIRTs
- These promote NDAs amongst participating CSIRTs

  (trust is insufficient if commercial stakes are high and there are legal risks)

  → a problem for FOSS projects which cannot normally sign NDAs

# References

**Open Source Policies:**
xenproject.org/security-policy.html
security.openstack.org/vmt-process.html
oss-security.openwall.org/wiki/mailing-lists/distros

**Best Practice and Standards:**
ISO/IEC29147 → ISO/IEC DIS 29147
www.first.org
www.first.org/global/sigs/
vulnerability-coordination/multiparty/guidelines-v1.0

# Questions

lars.kurth@xenproject.org

Picture by Lars Kurth