How Do Developers Act On Static Analysis Alerts? An Empirical Study of Coverity Usage

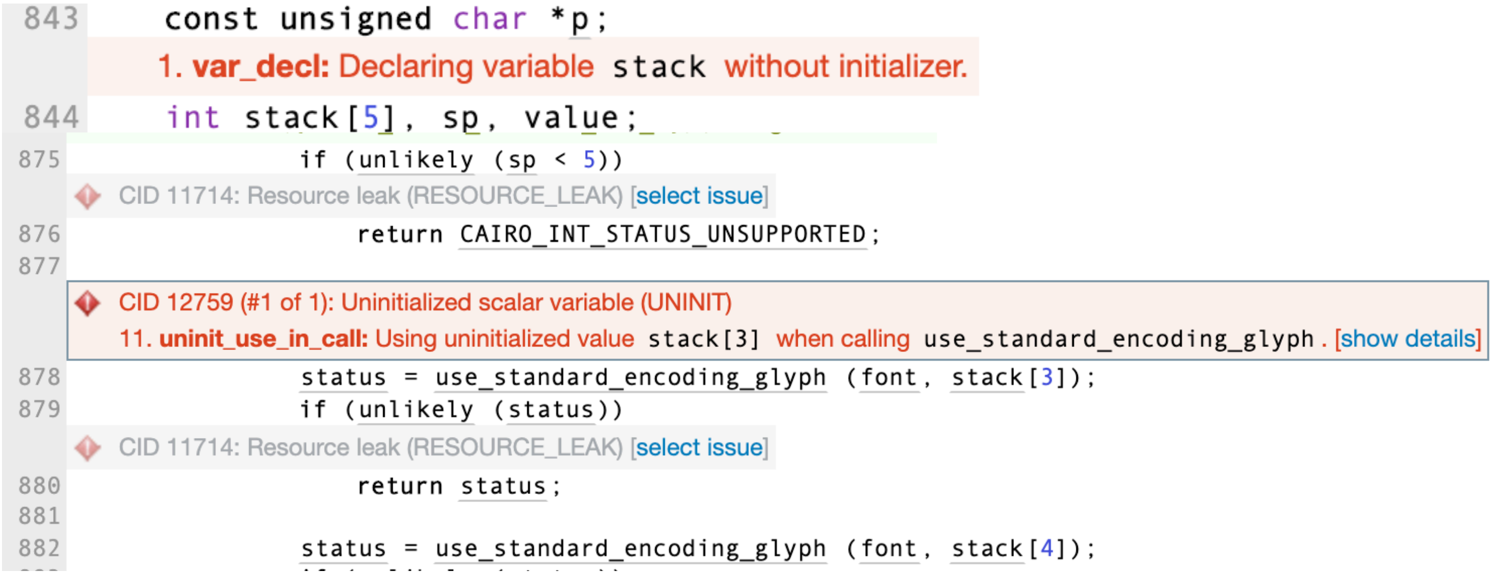
Presentation outline

- Problem background
- Research questions
- Related work
- Dataset and methodology
- Findings
- Threats to validity
- Key Takeaways

Static analysis tool

- Analyzes the code without *running* it
- Commonly includes data-flow, control-flow analysis, pattern matching etc.
- Finds potential security, performance, and reliability errors (*alerts*)

- Used alongside other testing methods (e.g. dynamic analysis, code review)
- Find defects early in the development process (a part of shift-left testing)



```
    const unsigned char *p;
    1. var_decl: Declaring variable stack without initializer.
    int stack[5], sp, value;
            if (unlikely (sp < 5))
CID 11714: Resource leak (RESOURCE_LEAK) [select issue]
                return CAIRO_INT_STATUS_UNSUPPORTED;

CID 12759 (#1 of 1): Uninitialized scalar variable (UNINIT)
11. uninit_use_in_call: Using uninitialized value stack[3] when calling use_standard_encoding_glyph. [show details]
            status = use_standard_encoding_glyph (font, stack[3]);
            if (unlikely (status))
CID 11714: Resource leak (RESOURCE_LEAK) [select issue]
                return status;

            status = use_standard_encoding_glyph (font, stack[4]);
```

Downsides of static analysis tools

- False positives
- Trivial issues
- Large volume of alerts

How do developers act on static analysis alerts?

- Do developers *care to fix*?
- *Which alerts* do they fix?
- How much *time* they take to fix?
- How *complex* are the fixes?
- How do the developers *prioritize* the alerts?

The goal of this paper *is to aid researchers and tool makers in improving the utility of static analysis tools through an empirical study of developer action on the static analysis alerts.*

What data do we need?

- Historical data
  - Developers working in the *real-world*
  - Developers *actively using* a static analysis tool

Our dataset

- Coverity
  - State-of-the-art static analysis tool
  - Maintains alert history in a database for individual projects

- Five open source projects
  - Linux, Firefox, Samba, Kodi, and Ovirt-engine
  - Four C/C++, one Java project
  - Maintainers *confirmed they monitor* Coverity alerts

Research questions

- (*Actionability*) Which Coverity alerts are fixed by the developers through code change?
- (*Lifespan*) How long do Coverity alerts remain in the code before being fixed?
- (*Fix complexity*) What is the complexity of the code changes that fix a Coverity alert?
- (*Prioritization*)
  - Do Coverity alerts with higher severity have shorter lifespan?
  - Do Coverity alerts with lower fix complexity have shorter lifespan?

- **Actionable alert**: An alert that developers *fix* through code changes
- **Actionability**: The rate of actionable alerts over total alerts

Related work

- Researchers run static analysis tools on historical code versions of a project and measure *actionability* of alerts:
  - – Liu et al. find actionability to be < 1% [TSE2018]
    - FindBugs on 730 Java projects
  - – Kim et al. find actionability to be 6% to 9% [FSE2007]
    - 3 tools on 3 programs
- When *developers are using* a static analysis tool:
  - – Marcillo et al. studied 248 Java projects from SonarQube database and find actionability to be 13% and lifespan 19 days [ICPC2019]

What do we add?

- Current *actionability* scenario
  - Historical data from large-scale projects when developers are actively using the tool
  - C/C++ projects
  - **Map alert history to code change history**
- An analysis of ***fix complexity***

Our dataset

- Coverity
  - State-of-the-art static analysis tool
  - Maintains alert history in a database for individual projects

- Five open source projects
  - Linux, Firefox, Samba, Kodi, and Ovirt-engine
  - Four C/C++, one Java project
  - Maintainers *confirmed they monitor* Coverity alerts

- **Data from at least past five years:**
  - **Alert history**
  - **Code history**

Open source projects using Coverity

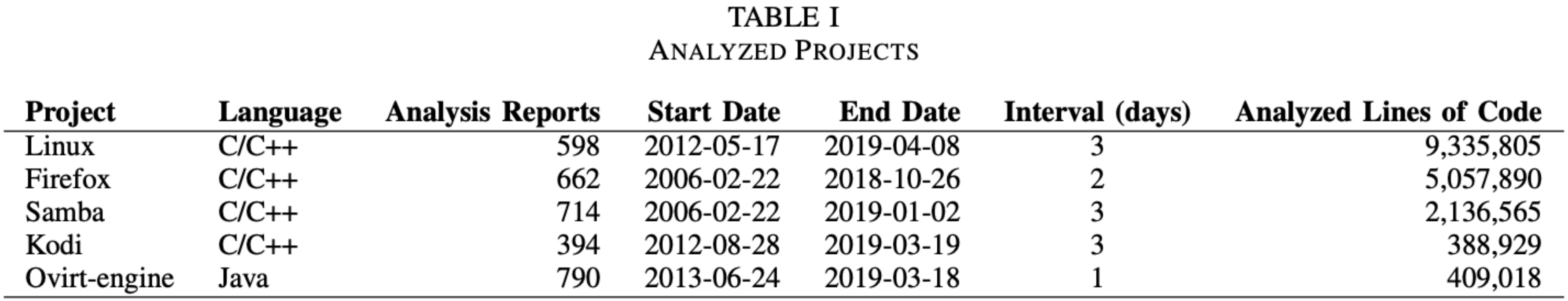
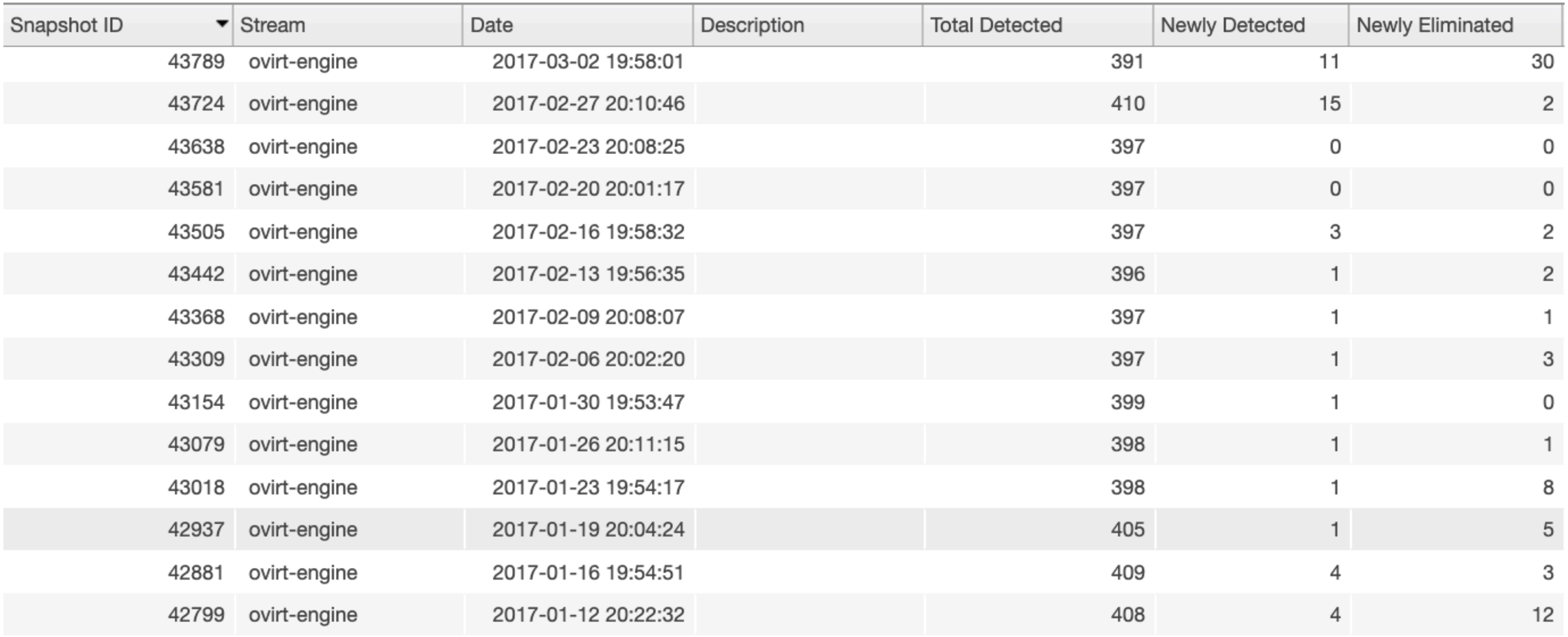


TABLE I
ANALYZED PROJECTS

Project	Language	Analysis Reports	Start Date	End Date	Interval (days)	Analyzed Lines of Code
Linux	C/C++	598	2012-05-17	2019-04-08	3	9,335,805
Firefox	C/C++	662	2006-02-22	2018-10-26	2	5,057,890
Samba	C/C++	714	2006-02-22	2019-01-02	3	2,136,565
Kodi	C/C++	394	2012-08-28	2019-03-19	3	388,929
Ovirt-engine	Java	790	2013-06-24	2019-03-18	1	409,018

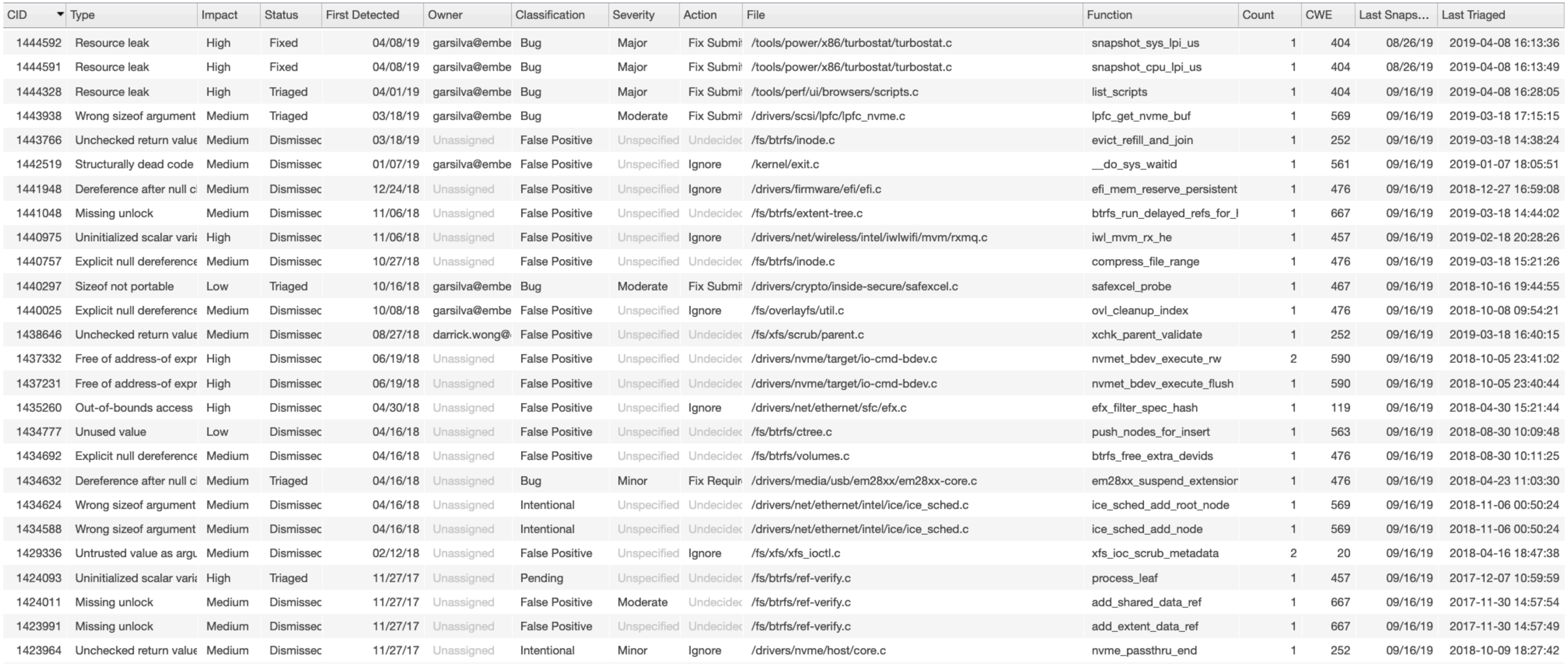
Analysis history on Coverity database



Snapshot ID	Stream	Date	Description	Total Detected	Newly Detected	Newly Eliminated
43789	ovirt-engine	2017-03-02 19:58:01		391	11	30
43724	ovirt-engine	2017-02-27 20:10:46		410	15	2
43638	ovirt-engine	2017-02-23 20:08:25		397	0	0
43581	ovirt-engine	2017-02-20 20:01:17		397	0	0
43505	ovirt-engine	2017-02-16 19:58:32		397	3	2
43442	ovirt-engine	2017-02-13 19:56:35		396	1	2
43368	ovirt-engine	2017-02-09 20:08:07		397	1	1
43309	ovirt-engine	2017-02-06 20:02:20		397	1	3
43154	ovirt-engine	2017-01-30 19:53:47		399	1	0
43079	ovirt-engine	2017-01-26 20:11:15		398	1	1
43018	ovirt-engine	2017-01-23 19:54:17		398	1	8
42937	ovirt-engine	2017-01-19 20:04:24		405	1	5
42881	ovirt-engine	2017-01-16 19:54:51		409	4	3
42799	ovirt-engine	2017-01-12 20:22:32		408	4	12

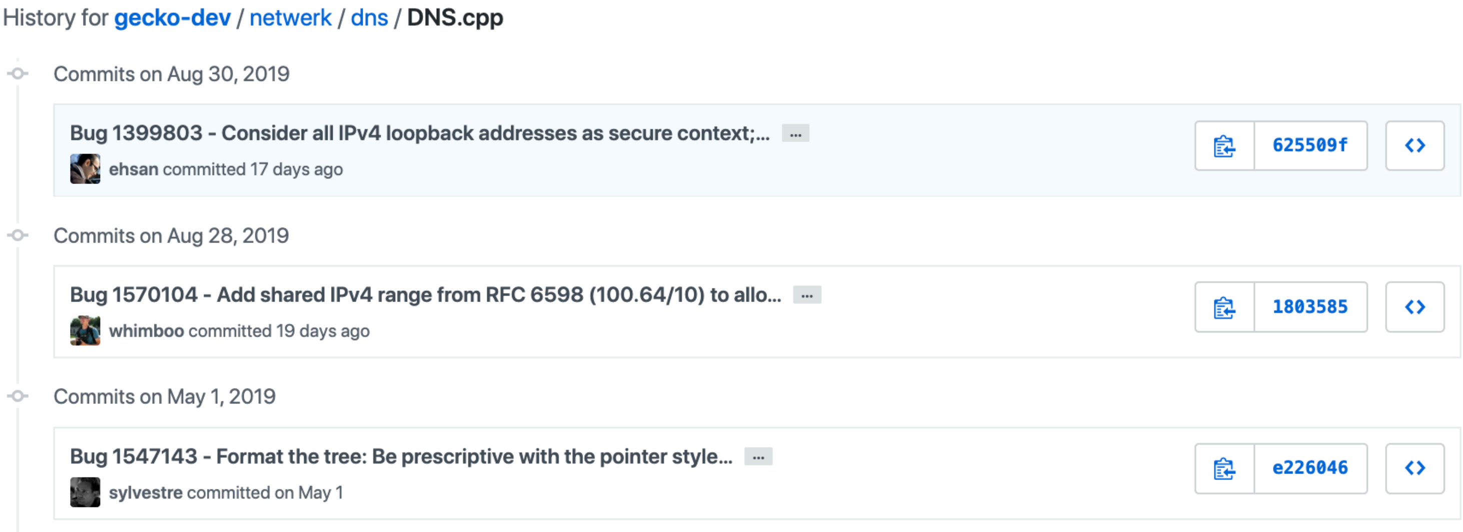
Alert history on Coverity database

- Unique alert id, alert type, severity, status (fixed/dismissed/new), date when first detected, a classification given by the developers (e.g. bug/false positive or unclassified by default), file location, date when last detected



CID	Type	Impact	Status	First Detected	Owner	Classification	Severity	Action	File	Function	Count	CWE	Last Snaps…	Last Triaged
1444592	Resource leak	High	Fixed	04/08/19	garsilva@embe	Bug	Major	Fix Submit	/tools/power/x86/turbostat/turbostat.c	snapshot_sys_lpi_us	1	404	08/26/19	2019-04-08 16:13:36
1444591	Resource leak	High	Fixed	04/08/19	garsilva@embe	Bug	Major	Fix Submit	/tools/power/x86/turbostat/turbostat.c	snapshot_cpu_lpi_us	1	404	08/26/19	2019-04-08 16:13:49
1444328	Resource leak	High	Triaged	04/01/19	garsilva@embe	Bug	Major	Fix Submit	/tools/perf/ui/browsers/scripts.c	list_scripts	1	404	09/16/19	2019-04-08 16:28:05
1443938	Wrong sizeof argument	Medium	Triaged	03/18/19	garsilva@embe	Bug	Moderate	Fix Submit	/drivers/scsi/lpfc/lpfc_nvme.c	lpfc_get_nvme_buf	1	569	09/16/19	2019-03-18 17:15:15
1443766	Unchecked return value	Medium	Dismissec	03/18/19	Unassigned	False Positive	Unspecified	Undecidec	/fs/btrfs/inode.c	evict_refill_and_join	1	252	09/16/19	2019-03-18 14:38:24
1442519	Structurally dead code	Medium	Dismissec	01/07/19	garsilva@embe	False Positive	Unspecified	Ignore	/kernel/exit.c	__do_sys_waitid	1	561	09/16/19	2019-01-07 18:05:51
1441948	Dereference after null cl	Medium	Dismissec	12/24/18	Unassigned	False Positive	Unspecified	Ignore	/drivers/firmware/efi/efi.c	efi_mem_reserve_persistent	1	476	09/16/19	2018-12-27 16:59:08
1441048	Missing unlock	Medium	Dismissec	11/06/18	Unassigned	False Positive	Unspecified	Undecidec	/fs/btrfs/extent-tree.c	btrfs_run_delayed_refs_for_l	1	667	09/16/19	2019-03-18 14:44:02
1440975	Uninitialized scalar varia	High	Dismissec	11/06/18	Unassigned	False Positive	Unspecified	Ignore	/drivers/net/wireless/intel/iwlwifi/mvm/rxmq.c	iwl_mvm_rx_he	1	457	09/16/19	2019-02-18 20:28:26
1440757	Explicit null dereference	Medium	Dismissec	10/27/18	Unassigned	False Positive	Unspecified	Undecidec	/fs/btrfs/inode.c	compress_file_range	1	476	09/16/19	2019-03-18 15:21:26
1440297	Sizeof not portable	Low	Triaged	10/16/18	garsilva@embe	Bug	Moderate	Fix Submit	/drivers/crypto/inside-secure/safexcel.c	safexcel_probe	1	467	09/16/19	2018-10-16 19:44:55
1440025	Explicit null dereference	Medium	Dismissec	10/08/18	garsilva@embe	False Positive	Unspecified	Ignore	/fs/overlayfs/util.c	ovl_cleanup_index	1	476	09/16/19	2018-10-08 09:54:21
1438646	Unchecked return value	Medium	Dismissec	08/27/18	darrick.wong@	False Positive	Unspecified	Undecidec	/fs/xfs/scrub/parent.c	xchk_parent_validate	1	252	09/16/19	2019-03-18 16:40:15
1437332	Free of address-of expr	High	Dismissec	06/19/18	Unassigned	False Positive	Unspecified	Undecidec	/drivers/nvme/target/io-cmd-bdev.c	nvmet_bdev_execute_rw	2	590	09/16/19	2018-10-05 23:41:02
1437231	Free of address-of expr	High	Dismissec	06/19/18	Unassigned	False Positive	Unspecified	Undecidec	/drivers/nvme/target/io-cmd-bdev.c	nvmet_bdev_execute_flush	1	590	09/16/19	2018-10-05 23:40:44
1435260	Out-of-bounds access	High	Dismissec	04/30/18	Unassigned	False Positive	Unspecified	Ignore	/drivers/net/ethernet/sfc/efx.c	efx_filter_spec_hash	1	119	09/16/19	2018-04-30 15:21:44
1434777	Unused value	Low	Dismissec	04/16/18	Unassigned	False Positive	Unspecified	Undecidec	/fs/btrfs/ctree.c	push_nodes_for_insert	1	563	09/16/19	2018-08-30 10:09:48
1434692	Explicit null dereference	Medium	Dismissec	04/16/18	Unassigned	False Positive	Unspecified	Undecidec	/fs/btrfs/volumes.c	btrfs_free_extra_devids	1	476	09/16/19	2018-08-30 10:11:25
1434632	Dereference after null cl	Medium	Triaged	04/16/18	Unassigned	Bug	Minor	Fix Requir	/drivers/media/usb/em28xx/em28xx-core.c	em28xx_suspend_extensior	1	476	09/16/19	2018-04-23 11:03:30
1434624	Wrong sizeof argument	Medium	Dismissec	04/16/18	Unassigned	Intentional	Unspecified	Undecidec	/drivers/net/ethernet/intel/ice/ice_sched.c	ice_sched_add_root_node	1	569	09/16/19	2018-11-06 00:50:24
1434588	Wrong sizeof argument	Medium	Dismissec	04/16/18	Unassigned	Intentional	Unspecified	Undecidec	/drivers/net/ethernet/intel/ice/ice_sched.c	ice_sched_add_node	1	569	09/16/19	2018-11-06 00:50:24
1429336	Untrusted value as argu	Medium	Dismissec	02/12/18	Unassigned	False Positive	Unspecified	Ignore	/fs/xfs/xfs_ioctl.c	xfs_ioc_scrub_metadata	2	20	09/16/19	2018-04-16 18:47:38
1424093	Uninitialized scalar varia	High	Triaged	11/27/17	Unassigned	Pending	Unspecified	Undecidec	/fs/btrfs/ref-verify.c	process_leaf	1	457	09/16/19	2017-12-07 10:59:59
1424011	Missing unlock	Medium	Dismissec	11/27/17	Unassigned	False Positive	Moderate	Undecidec	/fs/btrfs/ref-verify.c	add_shared_data_ref	1	667	09/16/19	2017-11-30 14:57:54
1423991	Missing unlock	Medium	Dismissec	11/27/17	Unassigned	False Positive	Unspecified	Undecidec	/fs/btrfs/ref-verify.c	add_extent_data_ref	1	667	09/16/19	2017-11-30 14:57:49
1423964	Unchecked return value	Medium	Dismissec	11/27/17	Unassigned	Intentional	Minor	Ignore	/drivers/nvme/host/core.c	nvme_passthru_end	1	252	09/16/19	2018-10-09 18:27:42

Code change history of a file



History for gecko-dev / netwerk / dns / DNS.cpp

Commits on Aug 30, 2019

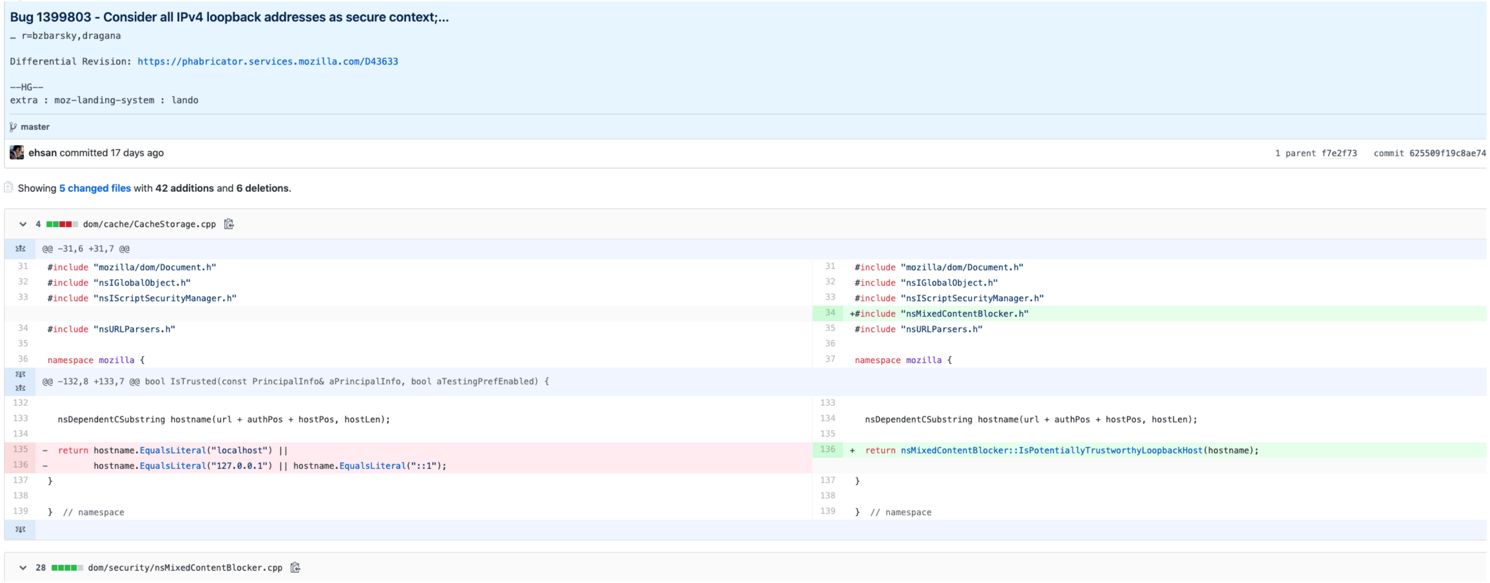
Bug 1399803 - Consider all IPv4 loopback addresses as secure context;... ...
ehsan committed 17 days ago
625509f

Commits on Aug 28, 2019

Bug 1570104 - Add shared IPv4 range from RFC 6598 (100.64/10) to allo... ...
whimboo committed 19 days ago
1803585

Commits on May 1, 2019

Bug 1547143 - Format the tree: Be prescriptive with the pointer style... ...
sylvestre committed on May 1
e226046



Bug 1399803 - Consider all IPv4 loopback addresses as secure context;...

```
… r=bzbarsky,dragana

Differential Revision: https://phabricator.services.mozilla.com/D43633

--HG--
extra : moz-landing-system : lando
```

master

ehsan committed 17 days ago — 1 parent f7e2f73 — commit 625509f19c8ae74

Showing 5 changed files with 42 additions and 6 deletions.

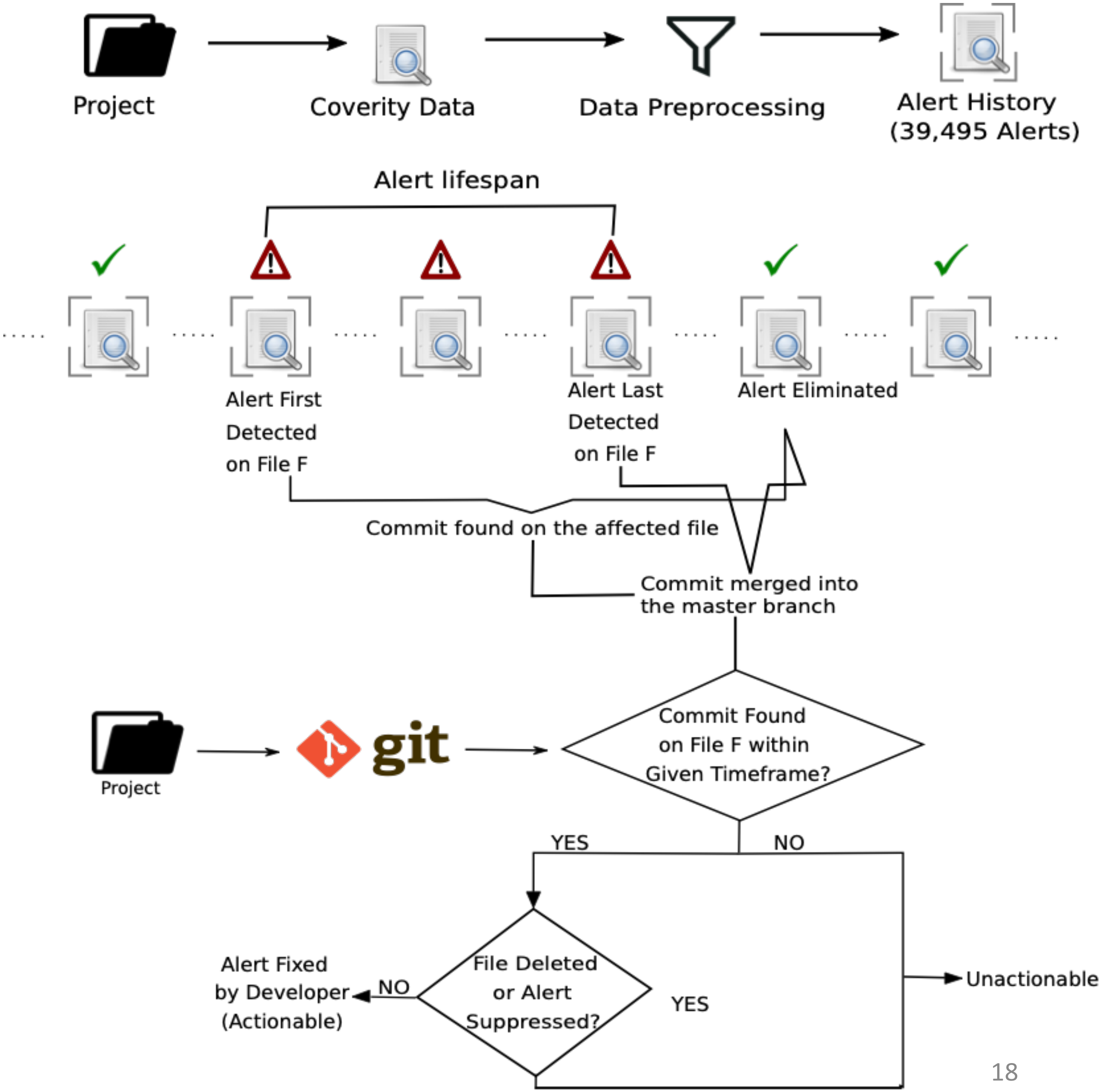
4 dom/cache/CacheStorage.cpp

```
@@ -31,6 +31,7 @@
 #include "mozilla/dom/Document.h"
 #include "nsIGlobalObject.h"
 #include "nsIScriptSecurityManager.h"

 #include "nsURLParsers.h"

 namespace mozilla {
@@ -132,8 +133,7 @@ bool IsTrusted(const PrincipalInfo& aPrincipalInfo, bool aTestingPrefEnabled) {

   nsDependentCSubstring hostname(url + authPos + hostPos, hostLen);

-  return hostname.EqualsLiteral("localhost") ||
-         hostname.EqualsLiteral("127.0.0.1") || hostname.EqualsLiteral("::1");
 }

 }  // namespace
```

```
 #include "mozilla/dom/Document.h"
 #include "nsIGlobalObject.h"
 #include "nsIScriptSecurityManager.h"
+#include "nsMixedContentBlocker.h"
 #include "nsURLParsers.h"

 namespace mozilla {


   nsDependentCSubstring hostname(url + authPos + hostPos, hostLen);

+  return nsMixedContentBlocker::IsPotentiallyTrustworthyLoopbackHost(hostname);

 }

 }  // namespace
```

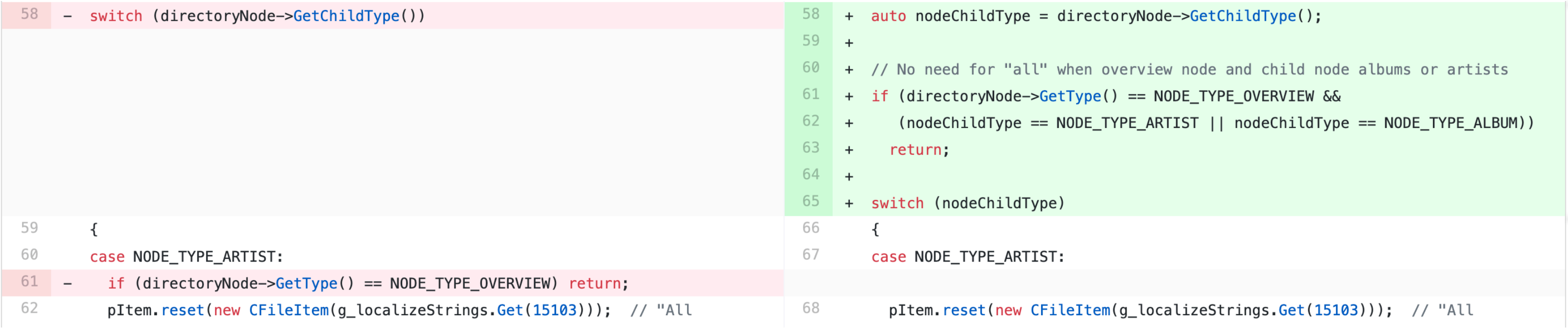
28 dom/security/nsMixedContentBlocker.cpp

How do we determine actionability and lifespan?


Project
Coverity Data
Data Preprocessing
Alert History (39,495 Alerts)
Alert lifespan
Alert First Detected on File F
Alert Last Detected on File F
Alert Eliminated
Commit found on the affected file
Commit merged into the master branch
Project
git
Commit Found on File F within Given Timeframe?
YES
NO
File Deleted or Alert Suppressed?
NO
YES
Alert Fixed by Developer (Actionable)
Unactionable

How do we determine fix complexity?

- Track *"fix commit"*
  - If *only one* commit found on the affected file when the alert gets fixed
- Metrics (adopted from Li et al. [CCS2017]):
  - No. of affected files
  - Total lines of code (LOC) change
  - Total logical change
  - In-file lines of code (LOC) change
  - In-file logical change



```
-  switch (directoryNode->GetChildType())
   {
   case NODE_TYPE_ARTIST:
-    if (directoryNode->GetType() == NODE_TYPE_OVERVIEW) return;
     pItem.reset(new CFileItem(g_localizeStrings.Get(15103)));  // "All
```

```
+  auto nodeChildType = directoryNode->GetChildType();
+
+  // No need for "all" when overview node and child node albums or artists
+  if (directoryNode->GetType() == NODE_TYPE_OVERVIEW &&
+     (nodeChildType == NODE_TYPE_ARTIST || nodeChildType == NODE_TYPE_ALBUM))
+    return;
+
+  switch (nodeChildType)
   {
   case NODE_TYPE_ARTIST:

     pItem.reset(new CFileItem(g_localizeStrings.Get(15103)));  // "All
```

One logical change

Findings

Findings: Actionability

Project	Total Alerts	Eliminated Alerts	Actionable Alerts	Triaged as Bug
Linux	17,133	60.3%	36.7%	3.6%
Firefox	12,945	73.6%	48.4%	8.2%
Samba	4,186	73.0%	27.4%	2.4%
Kodi	2,325	66.2%	49.5%	15.9%
Ovirt-engine	2,906	44.8%	31.3%	2.6%

Triaged as bug ⊂ Actionable alerts ⊂ Eliminated alerts ⊂ Total alerts

Unactionable alerts

Project	Total Alerts	Unactionable Alerts	Alive alerts	Triaged as false positive	Triaged as intentional	File deleted	Suppressed in code	Eliminated through undetermined ways
Linux	17,133	61.0%	26.4%	4.3%	4.6%	2.0%	0.9%	22.8%
Firefox	12,945	50.9%	9.7%	7.4%	7.6%	1.7%	0.7%	23.8%
Samba	4,186	72.6%	21.6%	2.5%	1.7%	1.9%	0.1%	44.7%
Kodi	2,325	50.2%	16.0%	3.6%	13.7%	4.1%	0.0%	12.8%
Ovirt-engine	2,906	68.4%	3.7%	44.0%	7.0%	0.9%	0.3%	12.5%

- For many unactionable alerts that were eliminated, we do not know how exactly they were eliminated
- Recent alerts which are alive for less time than the median lifespan of the project, we do not classify them as either actionable or unactionable

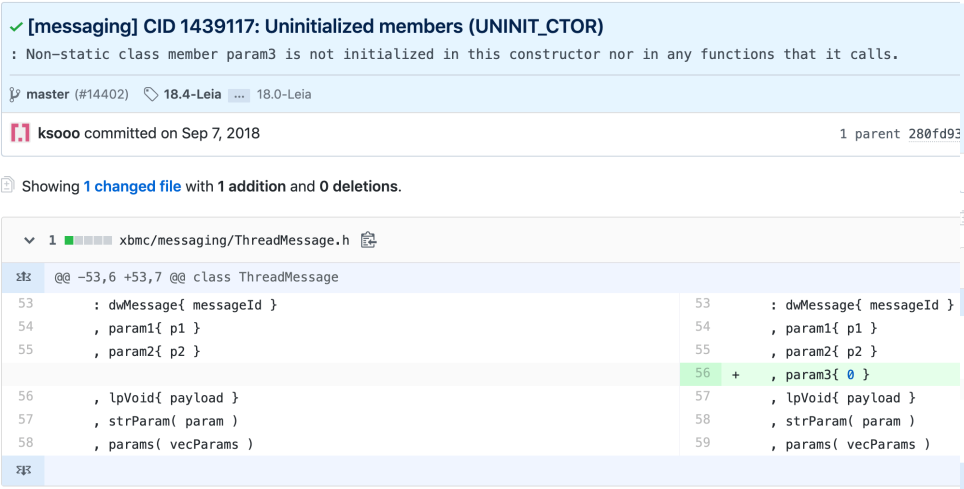
Findings: Lifespan (days)

Project	Actionable Alerts	Alerts Marked as Bug	Unactionable Eliminated
Linux	245.0	184.0	231.0
Firefox	124.0	64.0	174.0
Samba	39.5	200.0	46.0
Kodi	36.0	2.0	56.0
Ovirt-engine	96.0	43.0	152.0

For Linux, Firefox, and Kodi, alerts marked as bug get fixed *significantly* faster than other alerts

Findings: Fix complexity

Project	Fix commit tracked	Affected Files	Net LOC change	Net logical change	In-file LOC change	In-file logical change
All Alerts						
Linux	2299	1.0	4.0	1.5	3.0	1.0
Firefox	1835	3.0	40.7	11.0	5.0	1.7
Samba	639	1.0	3.0	1.0	2.0	1.0
Kodi	469	1.0	6.2	2.0	4.0	1.0
Ovirt-engine	666	1.5	24.9	5.5	7.0	2.0
Alerts Marked as Bug						
Linux	294	1.0	2.0	1.0	2.0	1.0
Firefox	345	1.0	9.5	3.0	4.0	1.5
Samba	27	1.0	5.0	1.0	4.0	1.0
Kodi	46	1.5	6.7	3.0	2.0	1.0
Ovirt-engine	68	1.0	4.0	2.0	4.0	1.0



✓ [messaging] CID 1439117: Uninitialized members (UNINIT_CTOR)

```
: Non-static class member param3 is not initialized in this constructor nor in any functions that it calls.
```

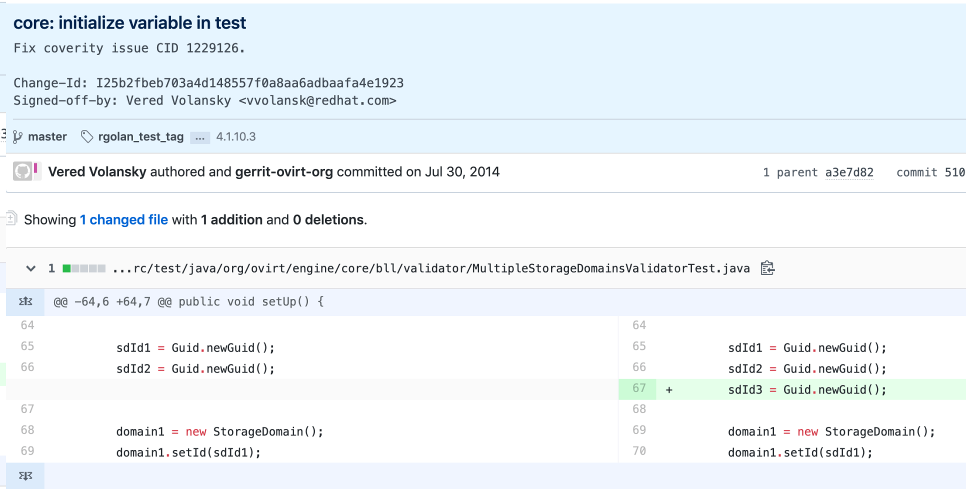
master (#14402) 18.4-Leia ... 18.0-Leia

ksooo committed on Sep 7, 2018 1 parent 280fd93

Showing 1 changed file with 1 addition and 0 deletions.

1 xbmc/messaging/ThreadMessage.h

```
@@ -53,6 +53,7 @@ class ThreadMessage
      : dwMessage{ messageId }             |       : dwMessage{ messageId }
      , param1{ p1 }                       |       , param1{ p1 }
      , param2{ p2 }                       |       , param2{ p2 }
                                           |   +   , param3{ 0 }
      , lpVoid{ payload }                  |       , lpVoid{ payload }
      , strParam( param )                  |       , strParam( param )
      , params( vecParams )                |       , params( vecParams )
```



core: initialize variable in test

```
Fix coverity issue CID 1229126.

Change-Id: I25b2fbeb703a4d148557f0a8aa6adbaafa4e1923
Signed-off-by: Vered Volansky <vvolansk@redhat.com>
```

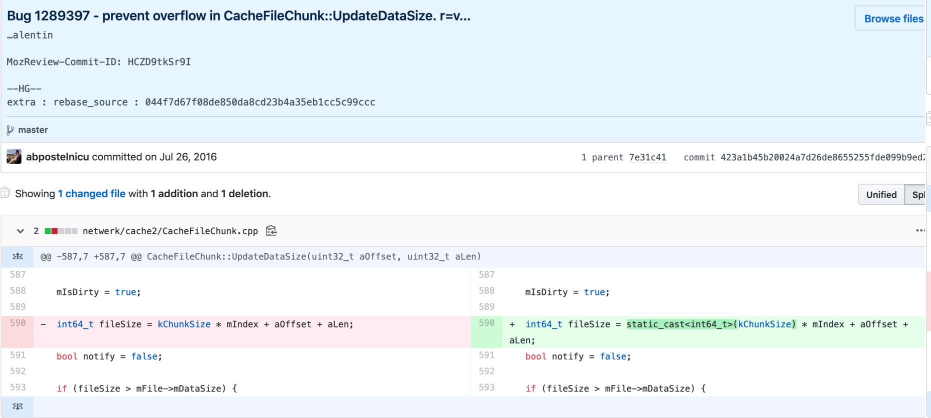
master rgolan_test_tag ... 4.1.10.3

Vered Volansky authored and gerrit-ovirt-org committed on Jul 30, 2014 1 parent a3e7d82 commit 510

Showing 1 changed file with 1 addition and 0 deletions.

1 ...rc/test/java/org/ovirt/engine/core/bll/validator/MultipleStorageDomainsValidatorTest.java

```
@@ -64,6 +64,7 @@ public void setUp() {

        sdId1 = Guid.newGuid();                  |         sdId1 = Guid.newGuid();
        sdId2 = Guid.newGuid();                  |         sdId2 = Guid.newGuid();
                                                 |   +     sdId3 = Guid.newGuid();

        domain1 = new StorageDomain();           |         domain1 = new StorageDomain();
        domain1.setId(sdId1);                    |         domain1.setId(sdId1);
```



Bug 1289397 - prevent overflow in CacheFileChunk::UpdateDataSize. r=v...

Browse files

```
…alentin

MozReview-Commit-ID: HCZD9tkSr9I

--HG--
extra : rebase_source : 044f7d67f08de850da8cd23b4a35eb1cc5c99ccc
```

master

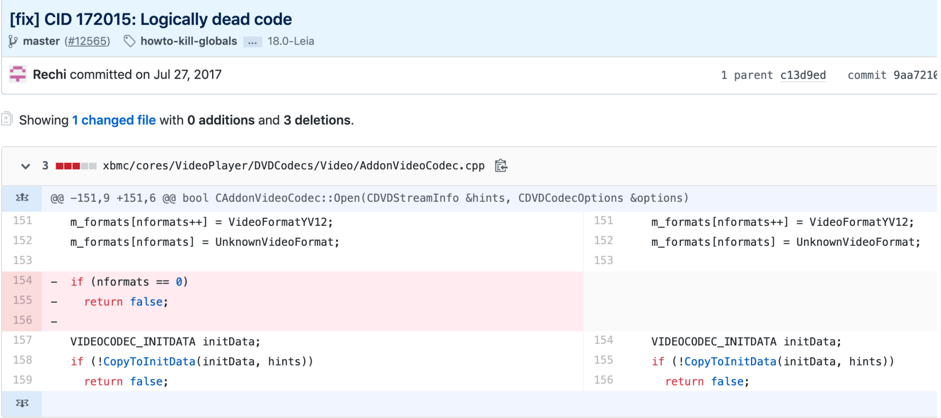
abpostelnicu committed on Jul 26, 2016 1 parent 7e31c41 commit 423a1b45b20024a7d26de8655255fde099b9ed2

Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

2 netwerk/cache2/CacheFileChunk.cpp

```
@@ -587,7 +587,7 @@ CacheFileChunk::UpdateDataSize(uint32_t aOffset, uint32_t aLen)

  mIsDirty = true;                                     |    mIsDirty = true;

- int64_t fileSize = kChunkSize * mIndex + aOffset + aLen;   | + int64_t fileSize = static_cast<int64_t>(kChunkSize) * mIndex + aOffset +
                                                       |   aLen;
  bool notify = false;                                 |    bool notify = false;

  if (fileSize > mFile->mDataSize) {                   |    if (fileSize > mFile->mDataSize) {
```



[fix] CID 172015: Logically dead code

master (#12565) howto-kill-globals ... 18.0-Leia

Rechi committed on Jul 27, 2017 1 parent c13d9ed commit 9aa721

Showing 1 changed file with 0 additions and 3 deletions.

3 xbmc/cores/VideoPlayer/DVDCodecs/Video/AddonVideoCodec.cpp

```
@@ -151,9 +151,6 @@ bool CAddonVideoCodec::Open(CDVDStreamInfo &hints, CDVDCodecOptions &options)
    m_formats[nformats++] = VideoFormatYV12;      |     m_formats[nformats++] = VideoFormatYV12;
    m_formats[nformats] = UnknownVideoFormat;     |     m_formats[nformats] = UnknownVideoFormat;
                                                  |
-   if (nformats == 0)                            |
-     return false;                               |
-                                                 |
    VIDEOCODEC_INITDATA initData;                 |     VIDEOCODEC_INITDATA initData;
    if (!CopyToInitData(initData, hints))         |     if (!CopyToInitData(initData, hints))
      return false;                               |       return false;
```

Example Coverity fixes from the developers

Findings: Do developers prioritize based on severity?

(Spearman's rank correlation for severity with lifespan)

Project	Correlation Coefficient (*r*)	High Alert Lifespan	Medium Alert Lifespan	Low Alert Lifespan
Linux	-0.02	217.0	248.0	206.0
Firefox	0.0	154.0	89.0	105.5
Samba	-0.01	36.0	56.0	18.0
✓ Kodi*	-0.29	2.0	26.5	416.0
Ovirt-engine	-0.03	89.0	77.0	129.0

* denotes statistically significant correlation

Findings: Do developers prioritize based on fix complexity?

(Spearman's rank correlation for fix complexity with lifespan)

	Project	Affected files	Net LOC change	Net logical change	In-file LOC change	In-file logical change
	Linux	0.05	0.09	x	0.06	x
✓	Firefox	0.17	0.20	0.20	0.12	0.07
	Samba	x	0.16	0.12	0.11	x
	Kodi	x	0.15	0.10	0.22	0.20
✓	Ovirt-engine	0.40	0.44	0.44	0.18	0.20

x denotes statistically insignificant correlation,
The rest of the correlations are statistically significant

Threats to validity

- We assume that if there is a code change in the affected file when an alert gets eliminated, developers have *fixed* the alert:
  - Code change can be unrelated to the alert
  - Code change can be made elsewhere to fix the alert
- Accurately tracking fix commit for an alert
  - We manually investigate 25 instances from each project to validate our method
- Generalizability threat
  - Coverity applies common static analysis techniques and catches a broad range of security and reliability alerts
  - Coverity is widely used in the industry

Feedback from an involved developer

your number match my gut feelings :)

you should be also aware that only two people were looking at coverity results (Andi and myself). It creates a strong bias as we aren't expert of all aspects of the code.

an interesting research area is "how many security issues found in the wild (CVE) were discovered by static analysis but not addressed?"

- Senior Engineering Manager, Mozilla

Key Takeaways:

The actionability rate of static analysis tool in the real word is around 36.7%, better than previously reported, but still lags behind the ideal.

Fixes for static analysis alerts are low in complexity: 1 to 2 units of logical changes in the affected file.

- Future research can be on estimating fix effort, automated program repair for static analysis alerts

Developers take ~3 months to fix the alerts.

- Future research can be on prioritizing security critical alerts, left-shifting static testing