Tool Support for Continuous Model-Based Verification of the Linux Kernel

Srinivas Dhanwada, Collin McIntyre, Ben Weno, Matt Wall
Client/Advisor: Dr. Suraj C. Kothari, Payas Awadhutkar

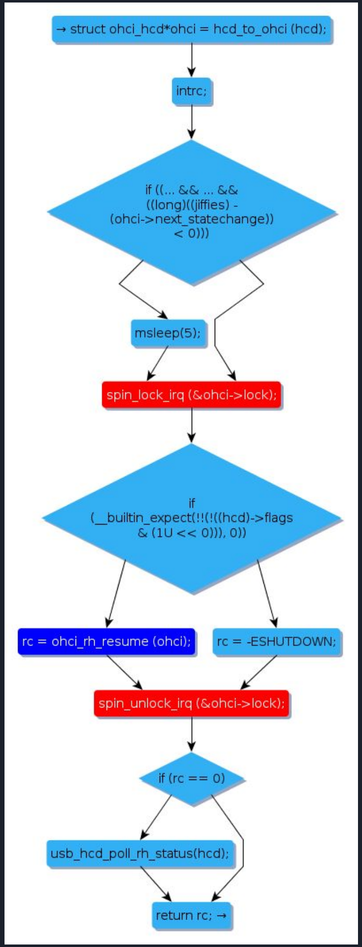
Overview

- L-SAP / Lock-Unlock Verification
- Our Solution
  - Automation Design
  - Differencing
  - Patching
  - Website
- Project Management
- Testing
- Results
- Summary

L-SAP / Lock-Unlock Verification

Lock-Unlock Verification Problem

- Problem: Verify each locking instance is followed by an unlocking instance on every *feasible* execution path
  - This is a specific case of the 2-event problem
  - Paths that miss the 2nd event pose a vulnerability problem


→ struct ohci_hcd*ohci = hcd_to_ohci (hcd);
intrc;
if ((... && ... && ((long)((jiffies) - (ohci->next_statechange)) < 0)))
msleep(5);
spin_lock_irq (&ohci->lock);
if (__builtin_expect(!!(!((hcd)->flags & (1U << 0))), 0))
rc = ohci_rh_resume (ohci);
rc = -ESHUTDOWN;
spin_unlock_irq (&ohci->lock);
if (rc == 0)
usb_hcd_poll_rh_status(hcd);
return rc; →

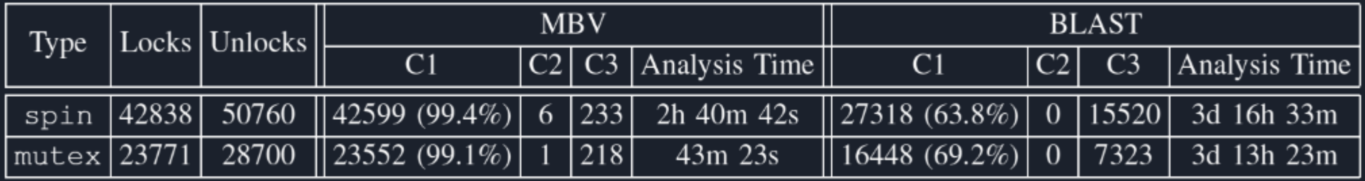
L-SAP

- Scalable and Accurate Lock/Unlock Pairing for the Linux Kernel
- L-SAP verifies the kernel by translating the code path into models and graphs
    - For each locking instance:
        - A graph is created tracking all the possible methods it can reach
        - A graph is created for each method tracking the status of the locking instance
    - L-SAP provides a conclusion, but also provides these graphs as evidence
    - Humans can look at this evidence and analyze inconclusive cases faster than looking at the codebase directly

Comparison of L-SAP to BLAST

TABLE III

SPIN AND MUTEX LOCK/UNLOCK PAIRING RESULTS ON LINUX KERNEL VERSIONS (3.17-RC1, 3.18-RC1 AND 3.19-RC1)



Type	Locks	Unlocks	MBV				BLAST			
			C1	C2	C3	Analysis Time	C1	C2	C3	Analysis Time
spin	42838	50760	42599 (99.4%)	6	233	2h 40m 42s	27318 (63.8%)	0	15520	3d 16h 33m
mutex	23771	28700	23552 (99.1%)	1	218	43m 23s	16448 (69.2%)	0	7323	3d 13h 23m

L-SAP is both faster, and is able to verify more instances than BLAST

Manually Running L-SAP

- Download newest release of linux kernel
    - Must be done manually once a new release is noticed
- Manually create a patch so the new release will run on L-SAP
- Run L-SAP
- Generate a basic webpage to display results

This process is something which can be automated, and a more useful representation of the results is needed

Requirements For Our Solution

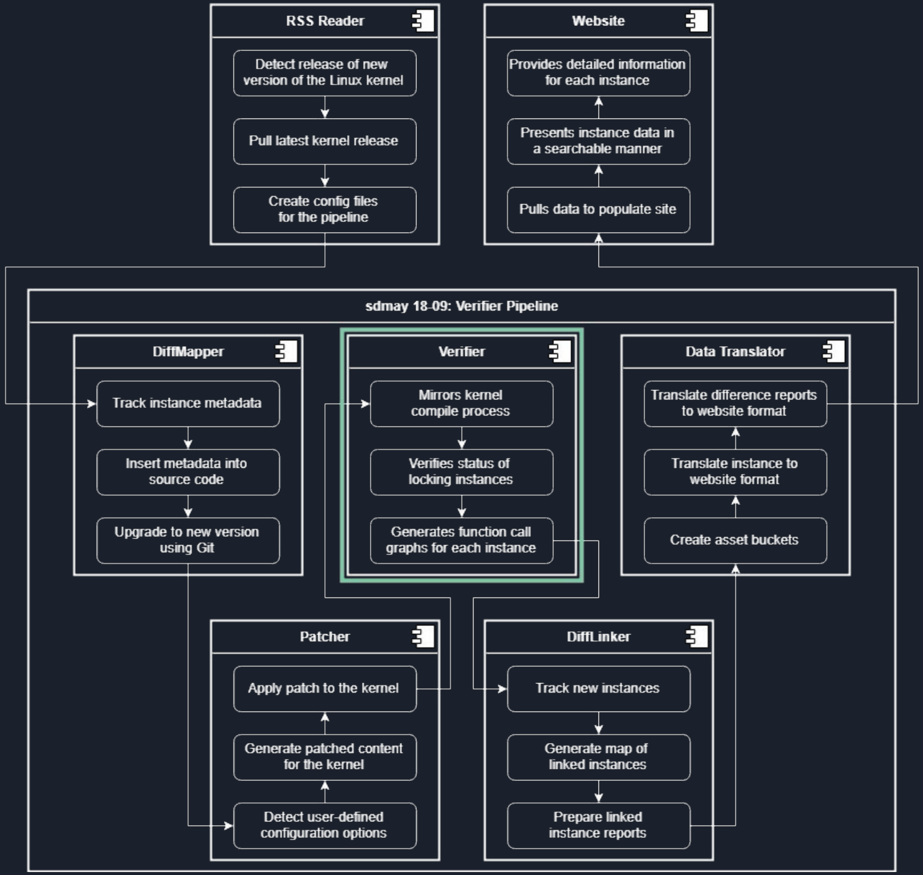
- Functional:
  - Automatically recognize kernel updates and start the tool
  - Create and apply a patch for the tool for each new version of the kernel
  - Run the tool with the new patch
  - Create a difference mapping for each locking instance
  - Post the results to the website
    - The website must properly display the new results in an organized manner
- Non Functional
  - The tool must run in a reasonable amount of time
  - The website must have a high level of usability, and scale to it’s demands

Our Solution

Our Solution

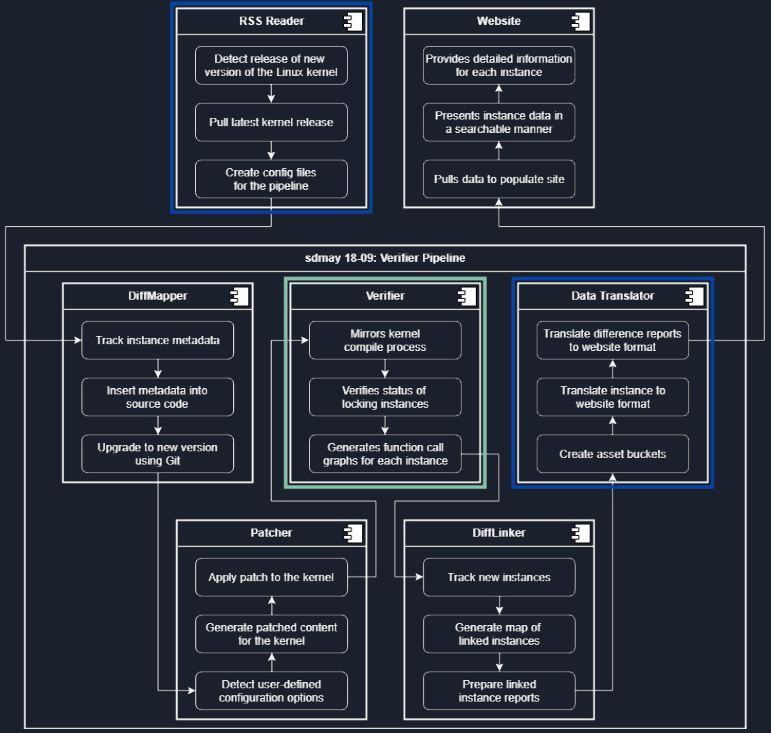
- Modularize and automate the pipeline
  - Instance Mapping
  - Patch Creation and Application
  - Running L-SAP
  - Differencing Results
- Redesign website to make tracking results easier
  - Provide search criteria
    - Text search
    - By driver search
  - Clearly show which instances are mismatched

Our Solution

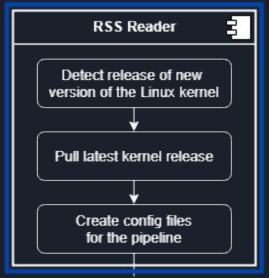


RSS Reader
Detect release of new version of the Linux kernel
Pull latest kernel release
Create config files for the pipeline
Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site
sdmay 18-09: Verifier Pipeline
DiffMapper
Track instance metadata
Insert metadata into source code
Upgrade to new version using Git
Verifier
Mirrors kernel compile process
Verifies status of locking instances
Generates function call graphs for each instance
Data Translator
Translate difference reports to website format
Translate instance to website format
Create asset buckets
Patcher
Apply patch to the kernel
Generate patched content for the kernel
Detect user-defined configuration options
DiffLinker
Track new instances
Generate map of linked instances
Prepare linked instance reports

Automation Design

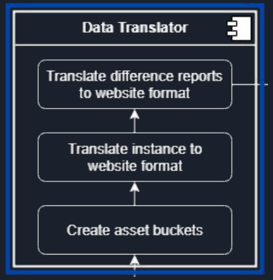

RSS Reader
Detect release of new version of the Linux kernel
Pull latest kernel release
Create config files for the pipeline
Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site
sdmay 18-09: Verifier Pipeline
DiffMapper
Track instance metadata
Insert metadata into source code
Upgrade to new version using Git
Verifier
Mirrors kernel compile process
Verifies status of locking instances
Generates function call graphs for each instance
Data Translator
Translate difference reports to website format
Translate instance to website format
Create asset buckets
Patcher
Apply patch to the kernel
Generate patched content for the kernel
Detect user-defined configuration options
DiffLinker
Track new instances
Generate map of linked instances
Prepare linked instance reports

Automation Design


RSS Reader
Detect release of new version of the Linux kernel
Pull latest kernel release
Create config files for the pipeline

- RSS Reader:
  - Polls the linux release rss feed and detects when a new release becomes available
  - Automatically pulls the newest release from github
  - Generates a config file that is used by other modules within the pipeline

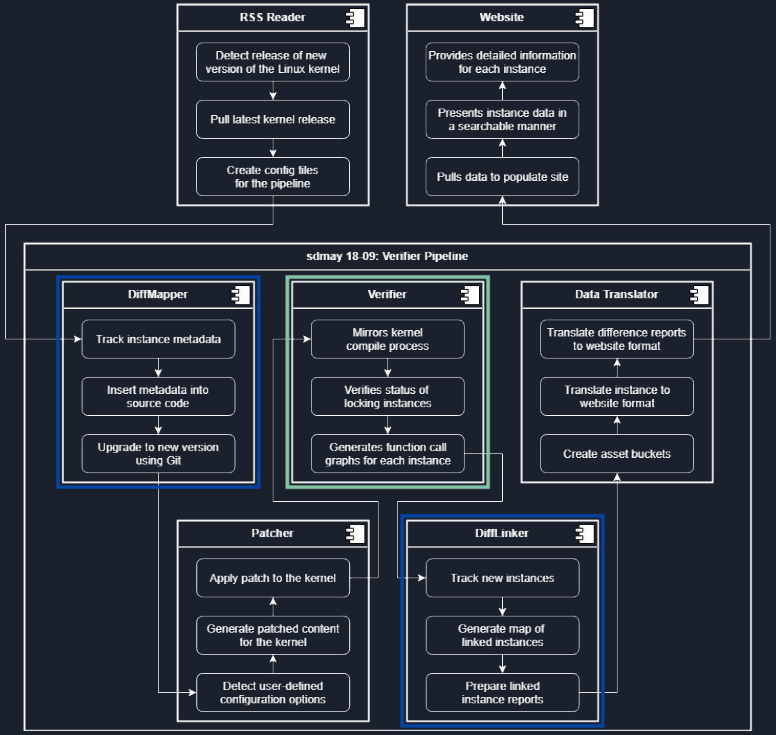
Automation Design



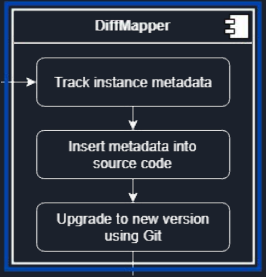
Data Translator
Translate difference reports to website format
Translate instance to website format
Create asset buckets

- Data Translator:
    - Scans through the output files from L-SAP and the DiffLinker to create files to upload to the database which include:
        - JSON file containing instance data for the new version
        - JSON file containing links information between the new version and the previous
        - An asset bucket containing the images of graphs within the proper directory structure

Differencing

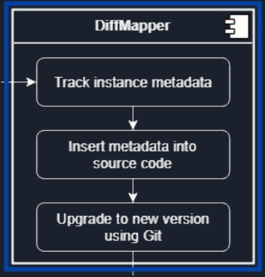

RSS Reader
Detect release of new version of the Linux kernel
Pull latest kernel release
Create config files for the pipeline
Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site
sdmay 18-09: Verifier Pipeline
DiffMapper
Track instance metadata
Insert metadata into source code
Upgrade to new version using Git
Verifier
Mirrors kernel compile process
Verifies status of locking instances
Generates function call graphs for each instance
Data Translator
Translate difference reports to website format
Translate instance to website format
Create asset buckets
Patcher
Apply patch to the kernel
Generate patched content for the kernel
Detect user-defined configuration options
DiffLinker
Track new instances
Generate map of linked instances
Prepare linked instance reports

Differencing

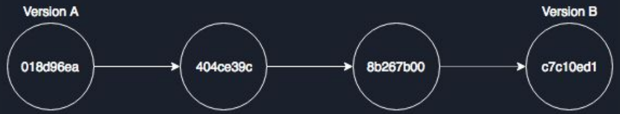

DiffMapper
Track instance metadata
Insert metadata into source code
Upgrade to new version using Git

- Problem: Current Implementation of L-SAP assigns a random ID to each instance
  - No mapping between versions is possible, which means no comparison is possible
  - Looking at the same point in source code does not work either
    - Additions/Deletions could happen nearby causing a shift in the location of the instance
    - Changes to the locking variable name could also make it hard to find between two versions
  - Need some way to have data move as the source code moves.
- Solution: Leverage the Linux Kernel's use of Git!
  - Insert a comment tag at the location of the instance -- commit this as a new branch
  - Use Git to rebase the differences between versions onto the new branch
    - Comments move with the source code changes!

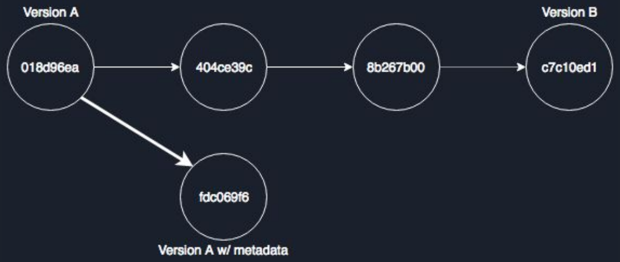
Differencing


DiffMapper
Track instance metadata
Insert metadata into source code
Upgrade to new version using Git

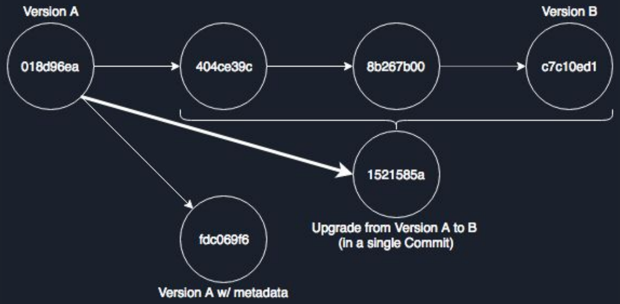
Step 1



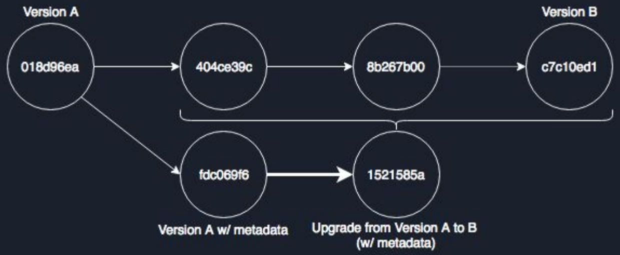
Step 2



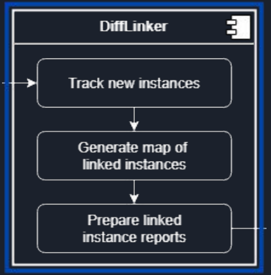
Step 3



Step 4

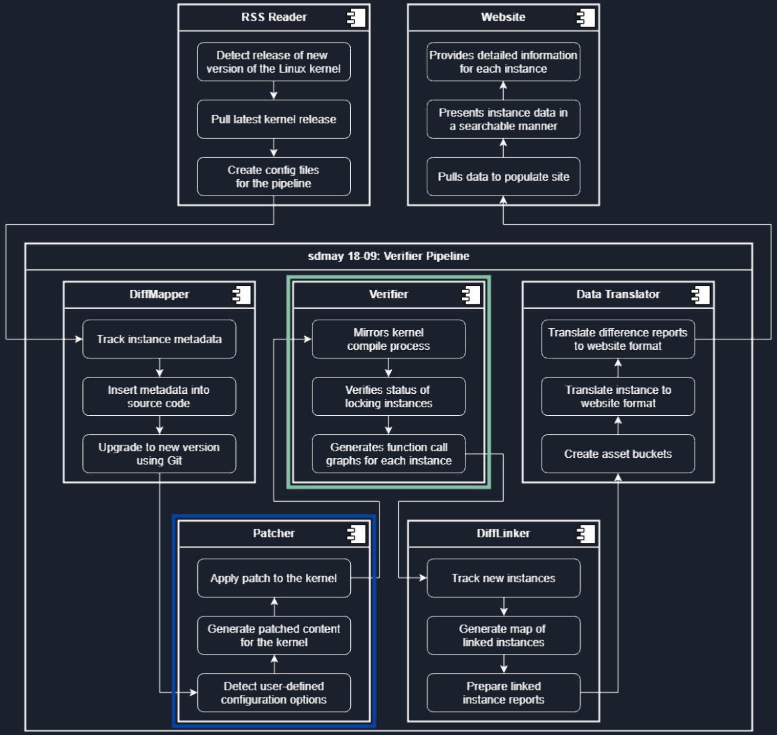


Differencing

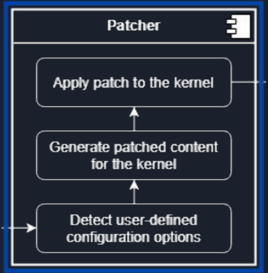

DiffLinker
Track new instances
Generate map of linked instances
Prepare linked instance reports

- Fetch results and look for changes between results
    - L-SAP is run with our modified kernel
    - Look at each instance for the metadata tag
    - Export all linked instances for the website
    - Analyze linked instances for changes and generate spreadsheet of changed instances
- At this point, the only metric of detecting changed instances is the status
    - L-SAP does generate data about the number of edges/nodes in a graph, which can be used to detect differences as well. This update is planned to help detect more changed instances

Patching

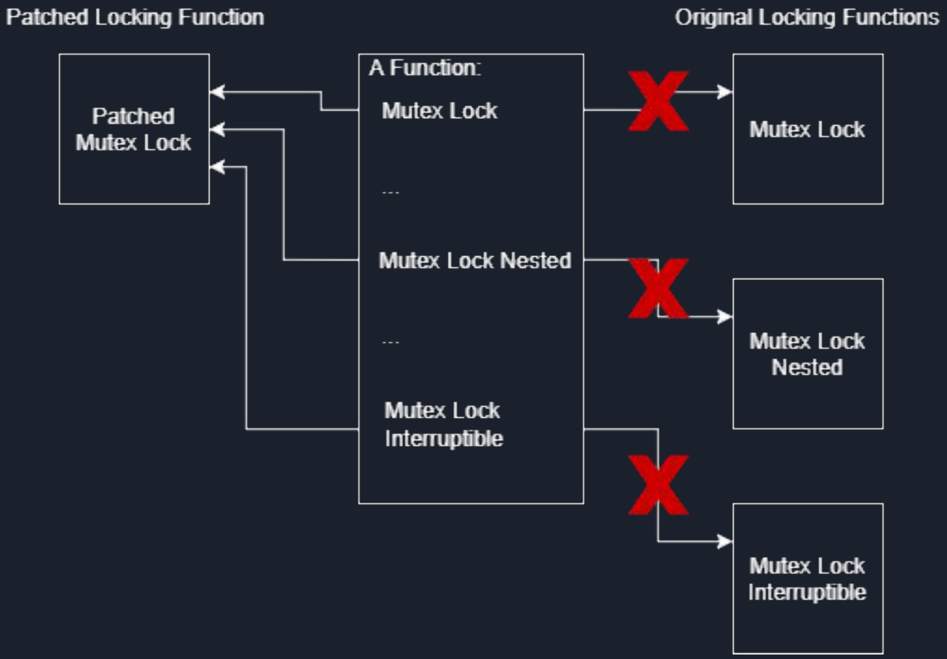

RSS Reader
Detect release of new version of the Linux kernel
Pull latest kernel release
Create config files for the pipeline
Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site
sdmay 18-09: Verifier Pipeline
DiffMapper
Track instance metadata
Insert metadata into source code
Upgrade to new version using Git
Verifier
Mirrors kernel compile process
Verifies status of locking instances
Generates function call graphs for each instance
Data Translator
Translate difference reports to website format
Translate instance to website format
Create asset buckets
Patcher
Apply patch to the kernel
Generate patched content for the kernel
Detect user-defined configuration options
DiffLinker
Track new instances
Generate map of linked instances
Prepare linked instance reports

Patcher

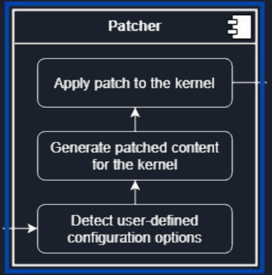

Patcher
Apply patch to the kernel
Generate patched content for the kernel
Detect user-defined configuration options

- What is the patch?
  - Redirection of all locking function calls to a single function per lock type

- Why is it needed?
  - L-SAP matches locks and unlocks based on function calls
  - Redirecting locking calls to a single function reduces computation time
  - Since kernel structure is all we care about, we can redirect to an empty function

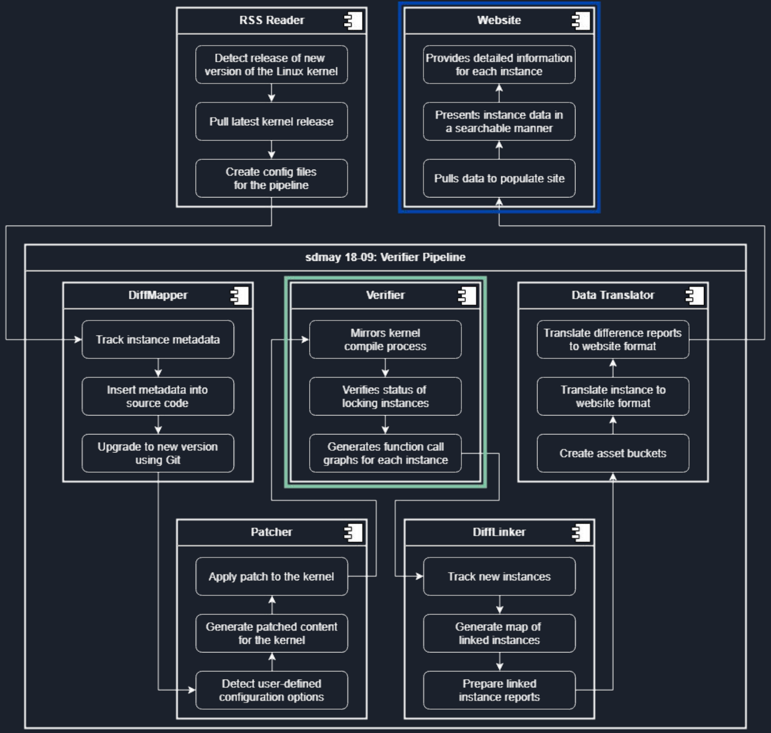


Patching Algorithm

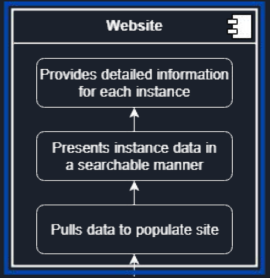

Patcher
Apply patch to the kernel
Generate patched content for the kernel
Detect user-defined configuration options

- Before the algorithm runs
  - Set various options in the Patcher’s config file
    - Locking function/macro criteria
    - Files with locking implementations
    - Predetermined patch content
  - Use command line arguments to specify kernel directory, output directory, and level of detail of logged messages.
- Overview
  - Get function/macro information from existing headers
  - Generate patched header files
  - Using function/macro information…
    - Remove existing function/macro declarations
    - Remove existing function/macro implementations
- Process is the same for mutex locks and spinlocks

Website Redesign

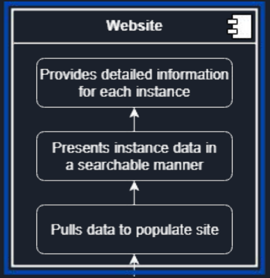

RSS Reader
Detect release of new version of the Linux kernel
Pull latest kernel release
Create config files for the pipeline
Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site
sdmay 18-09: Verifier Pipeline
DiffMapper
Track instance metadata
Insert metadata into source code
Upgrade to new version using Git
Verifier
Mirrors kernel compile process
Verifies status of locking instances
Generates function call graphs for each instance
Data Translator
Translate difference reports to website format
Translate instance to website format
Create asset buckets
Patcher
Apply patch to the kernel
Generate patched content for the kernel
Detect user-defined configuration options
DiffLinker
Track new instances
Generate map of linked instances
Prepare linked instance reports

Angular + Typescript

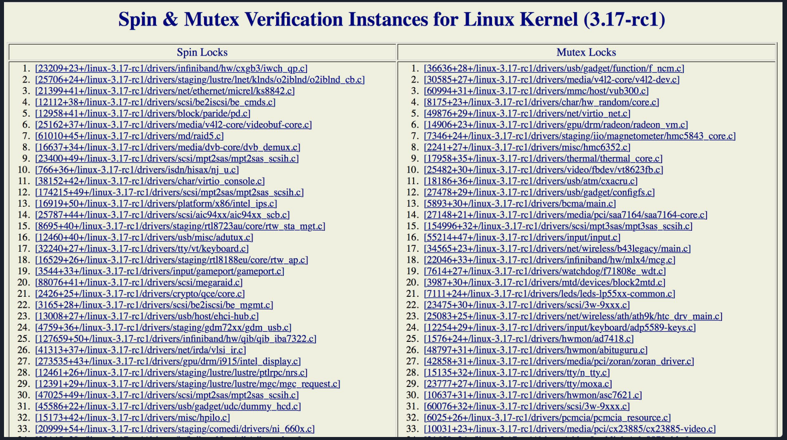

Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site

PROS	CONS
● Modularity	● Complex data structures
● Scalability	● Some server side implementation required
● Maintenance	● Must have javascript enabled
● Optimization	

Homepage Design

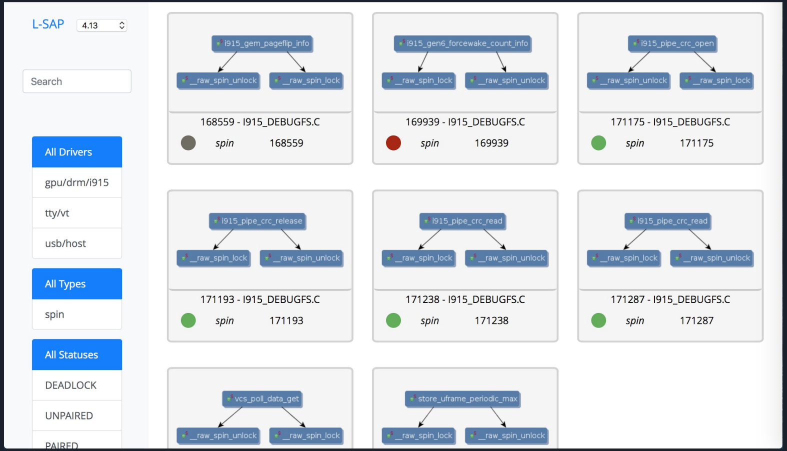


Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site



Spin & Mutex Verification Instances for Linux Kernel (3.17-rc1)

	Spin Locks		Mutex Locks
1.	[23209+23+/linux-3.17-rc1/drivers/infiniband/hw/cxgb3/iwch_qp.c]	1.	[36636+28+/linux-3.17-rc1/drivers/usb/gadget/function/f_ncm.c]
2.	[25706+24+/linux-3.17-rc1/drivers/staging/lustre/lnet/klnds/o2iblnd/o2iblnd_cb.c]	2.	[30585+27+/linux-3.17-rc1/drivers/media/v4l2-core/v4l2-dev.c]
3.	[21399+41+/linux-3.17-rc1/drivers/net/ethernet/micrel/ks8842.c]	3.	[60994+31+/linux-3.17-rc1/drivers/mmc/host/vub300.c]
4.	[12112+38+/linux-3.17-rc1/drivers/scsi/be2iscsi/be_cmds.c]	4.	[8175+23+/linux-3.17-rc1/drivers/char/hw_random/core.c]
5.	[12958+41+/linux-3.17-rc1/drivers/block/paride/pd.c]	5.	[49876+29+/linux-3.17-rc1/drivers/net/virtio_net.c]
6.	[25162+37+/linux-3.17-rc1/drivers/media/v4l2-core/videobuf-core.c]	6.	[14906+23+/linux-3.17-rc1/drivers/gpu/drm/radeon/radeon_vm.c]
7.	[61010+45+/linux-3.17-rc1/drivers/md/raid5.c]	7.	[7346+24+/linux-3.17-rc1/drivers/staging/iio/magnetometer/hmc5843_core.c]
8.	[16637+34+/linux-3.17-rc1/drivers/media/dvb-core/dvb_demux.c]	8.	[2241+27+/linux-3.17-rc1/drivers/misc/hmc6352.c]
9.	[23400+49+/linux-3.17-rc1/drivers/scsi/mpt2sas/mpt2sas_scsih.c]	9.	[17958+35+/linux-3.17-rc1/drivers/thermal/thermal_core.c]
10.	[766+36+/linux-3.17-rc1/drivers/isdn/hisax/nj_u.c]	10.	[25482+30+/linux-3.17-rc1/drivers/video/fbdev/vt8623fb.c]
11.	[38152+42+/linux-3.17-rc1/drivers/char/virtio_console.c]	11.	[18186+36+/linux-3.17-rc1/drivers/usb/atm/cxacru.c]
12.	[174215+49+/linux-3.17-rc1/drivers/scsi/mpt2sas/mpt2sas_scsih.c]	12.	[27478+29+/linux-3.17-rc1/drivers/usb/gadget/configfs.c]
13.	[16919+50+/linux-3.17-rc1/drivers/platform/x86/intel_ips.c]	13.	[5893+30+/linux-3.17-rc1/drivers/bcma/main.c]
14.	[25787+44+/linux-3.17-rc1/drivers/scsi/aic94xx/aic94xx_scb.c]	14.	[27148+21+/linux-3.17-rc1/drivers/media/pci/saa7164/saa7164-core.c]
15.	[8695+40+/linux-3.17-rc1/drivers/staging/rtl8723au/core/rtw_sta_mgt.c]	15.	[154996+32+/linux-3.17-rc1/drivers/scsi/mpt3sas/mpt3sas_scsih.c]
16.	[12460+40+/linux-3.17-rc1/drivers/usb/misc/adutux.c]	16.	[55214+47+/linux-3.17-rc1/drivers/input/input.c]
17.	[32240+27+/linux-3.17-rc1/drivers/tty/vt/keyboard.c]	17.	[34565+23+/linux-3.17-rc1/drivers/net/wireless/b43legacy/main.c]
18.	[16529+26+/linux-3.17-rc1/drivers/staging/rtl8188eu/core/rtw_ap.c]	18.	[22046+33+/linux-3.17-rc1/drivers/infiniband/hw/mlx4/mcg.c]
19.	[3544+33+/linux-3.17-rc1/drivers/input/gameport/gameport.c]	19.	[7614+27+/linux-3.17-rc1/drivers/watchdog/f71808e_wdt.c]
20.	[88076+41+/linux-3.17-rc1/drivers/scsi/megaraid.c]	20.	[3987+30+/linux-3.17-rc1/drivers/mtd/devices/block2mtd.c]
21.	[2426+25+/linux-3.17-rc1/drivers/crypto/qce/core.c]	21.	[7111+24+/linux-3.17-rc1/drivers/leds/leds-lp55xx-common.c]
22.	[3165+28+/linux-3.17-rc1/drivers/scsi/be2iscsi/be_mgmt.c]	22.	[23475+30+/linux-3.17-rc1/drivers/scsi/3w-9xxx.c]
23.	[13008+27+/linux-3.17-rc1/drivers/usb/host/ehci-hub.c]	23.	[25083+25+/linux-3.17-rc1/drivers/net/wireless/ath/ath9k/htc_drv_main.c]
24.	[4759+36+/linux-3.17-rc1/drivers/staging/gdm72xx/gdm_usb.c]	24.	[12254+29+/linux-3.17-rc1/drivers/input/keyboard/adp5589-keys.c]
25.	[127659+50+/linux-3.17-rc1/drivers/infiniband/hw/qib/qib_iba7322.c]	25.	[1576+24+/linux-3.17-rc1/drivers/hwmon/ad7418.c]
26.	[41313+37+/linux-3.17-rc1/drivers/net/irda/vlsi_ir.c]	26.	[48797+31+/linux-3.17-rc1/drivers/hwmon/abituguru.c]
27.	[273535+43+/linux-3.17-rc1/drivers/gpu/drm/i915/intel_display.c]	27.	[42858+31+/linux-3.17-rc1/drivers/media/pci/zoran/zoran_driver.c]
28.	[12461+26+/linux-3.17-rc1/drivers/staging/lustre/lustre/ptlrpc/nrs.c]	28.	[15135+32+/linux-3.17-rc1/drivers/tty/n_tty.c]
29.	[12391+29+/linux-3.17-rc1/drivers/staging/lustre/lustre/mgc/mgc_request.c]	29.	[23777+27+/linux-3.17-rc1/drivers/tty/moxa.c]
30.	[47025+49+/linux-3.17-rc1/drivers/scsi/mpt2sas/mpt2sas_scsih.c]	30.	[10637+31+/linux-3.17-rc1/drivers/hwmon/asc7621.c]
31.	[45586+22+/linux-3.17-rc1/drivers/usb/gadget/udc/dummy_hcd.c]	31.	[60076+32+/linux-3.17-rc1/drivers/scsi/3w-9xxx.c]
32.	[15173+42+/linux-3.17-rc1/drivers/misc/hpilo.c]	32.	[6025+26+/linux-3.17-rc1/drivers/pcmcia/pcmcia_resource.c]
33.	[20999+54+/linux-3.17-rc1/drivers/staging/comedi/drivers/ni_660x.c]	33.	[10031+23+/linux-3.17-rc1/drivers/media/pci/cx23885/cx23885-video.c]



168559 - I915_DEBUGFS.C
● *spin* 168559

169939 - I915_DEBUGFS.C
🔴 *spin* 169939

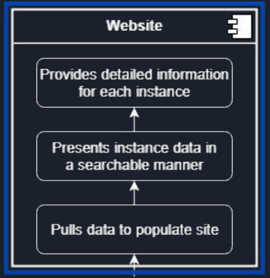
171175 - I915_DEBUGFS.C
🟢 *spin* 171175

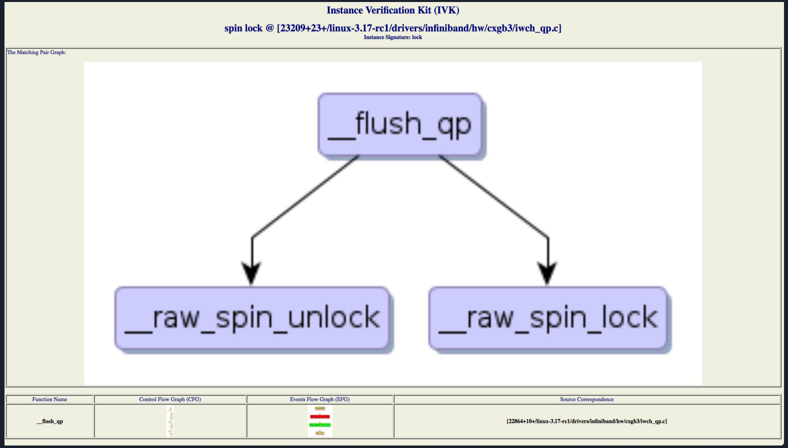
171193 - I915_DEBUGFS.C
🟢 *spin* 171193

171238 - I915_DEBUGFS.C
🟢 *spin* 171238

171287 - I915_DEBUGFS.C
🟢 *spin* 171287

Single Instance Design


Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site



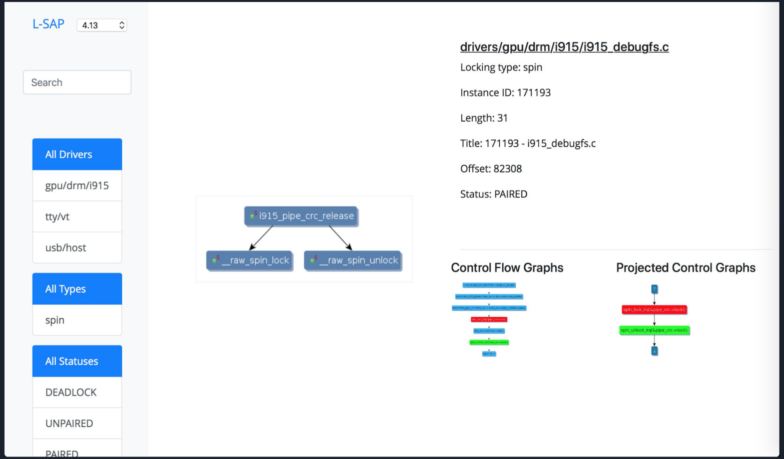
Instance Verification Kit (IVK)

spin lock @ [23209+23+/linux-3.17-rc1/drivers/infiniband/hw/cxgb3/iwch_qp.c]

Instance Signature: lock

The Matching Pair Graph:

Function Name	Control Flow Graph (CFG)	Events Flow Graph (EFG)	Source Correspondence
__flush_qp			[22864+10+/linux-3.17-rc1/drivers/infiniband/hw/cxgb3/iwch_qp.c]



drivers/gpu/drm/i915/i915_debugfs.c

Locking type: spin

Instance ID: 171193

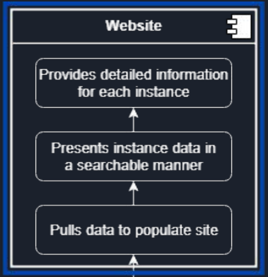
Length: 31

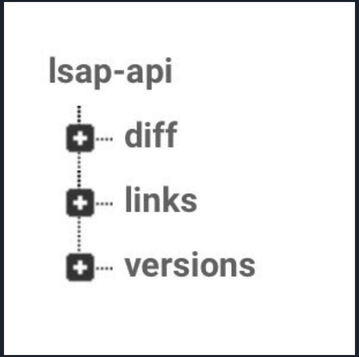
Title: 171193 - i915_debugfs.c

Offset: 82308

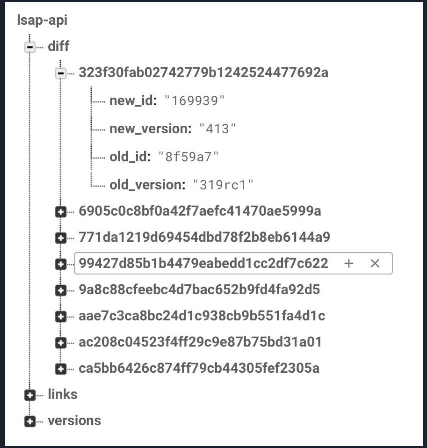
Status: PAIRED

Firebase (Backend Database)

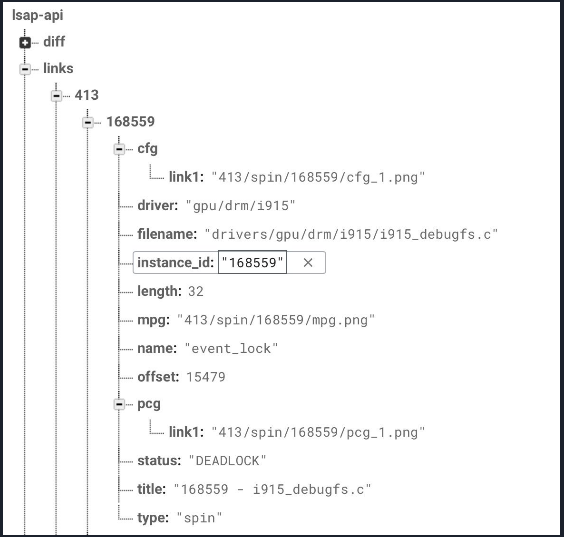

Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site



```
lsap-api
 ⊞ diff
 ⊞ links
 ⊞ versions
```

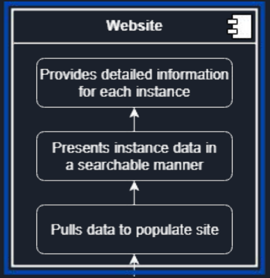


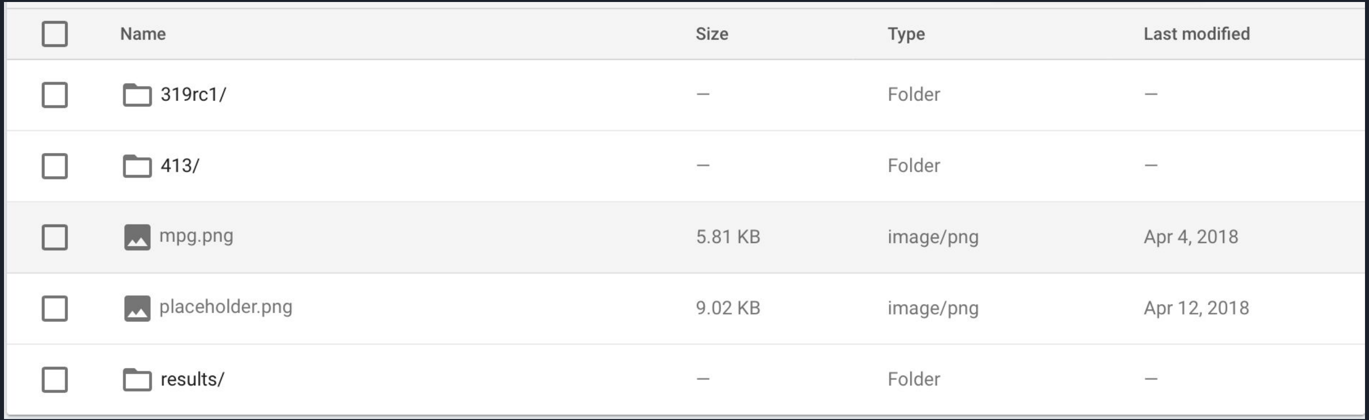
```
lsap-api
 ⊟ diff
    ⊟ 323f30fab02742779b1242524477692a
         new_id: "169939"
         new_version: "413"
         old_id: "8f59a7"
         old_version: "319rc1"
    ⊞ 6905c0c8bf0a42f7aefc41470ae5999a
    ⊞ 771da1219d69454dbd78f2b8eb6144a9
    ⊞ 99427d85b1b4479eabedd1cc2df7c622  + ×
    ⊞ 9a8c88cfeebc4d7bac652b9fd4fa92d5
    ⊞ aae7c3ca8bc24d1c938cb9b551fa4d1c
    ⊞ ac208c04523f4ff29c9e87b75bd31a01
    ⊞ ca5bb6426c874ff79cb44305fef2305a
 ⊞ links
 ⊞ versions
```



```
lsap-api
 ⊞ diff
 ⊟ links
    ⊟ 413
       ⊟ 168559
          ⊟ cfg
               link1: "413/spin/168559/cfg_1.png"
            driver: "gpu/drm/i915"
            filename: "drivers/gpu/drm/i915/i915_debugfs.c"
            instance_id: "168559"  ×
            length: 32
            mpg: "413/spin/168559/mpg.png"
            name: "event_lock"
            offset: 15479
          ⊟ pcg
               link1: "413/spin/168559/pcg_1.png"
            status: "DEADLOCK"
            title: "168559 - i915_debugfs.c"
            type: "spin"
```

Firebase (Backend Storage)


Website
Provides detailed information for each instance
Presents instance data in a searchable manner
Pulls data to populate site

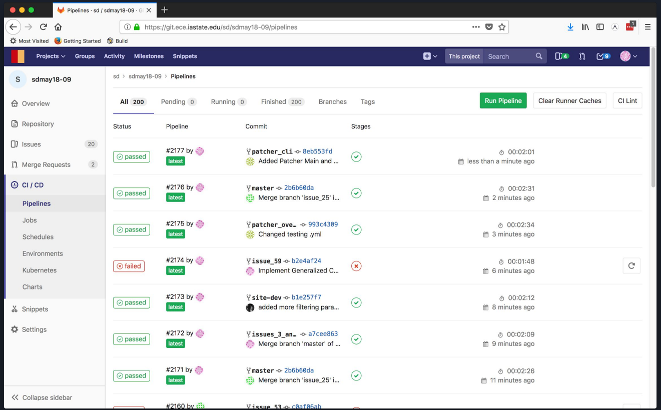


	Name	Size	Type	Last modified
☐	319rc1/	—	Folder	—
☐	413/	—	Folder	—
☐	mpg.png	5.81 KB	image/png	Apr 4, 2018
☐	placeholder.png	9.02 KB	image/png	Apr 12, 2018
☐	results/	—	Folder	—

Testing

Testing

- Unit Testing
  - Patching Algorithm, Differencing Algorithm, and The Automation Design are all written in Java
    - Unit Testing was done through JUnit testing framework
    - Tests were run on every commit pushed to our online repository
    - Achieved 92% Code Coverage
- Small Scale Testing (v3.19-rc1 to v4.13)
  - Pipeline was run using a subset of instances:
  - Diff Mapper mapped 83% of instances
  - Patcher generated correct patch
  - Diff Linker captured 700 of a 1200 subset
    - 50 instances were marked for analysis
  - Data Translator Generated Website Data
  - Website of instances has been deployed.

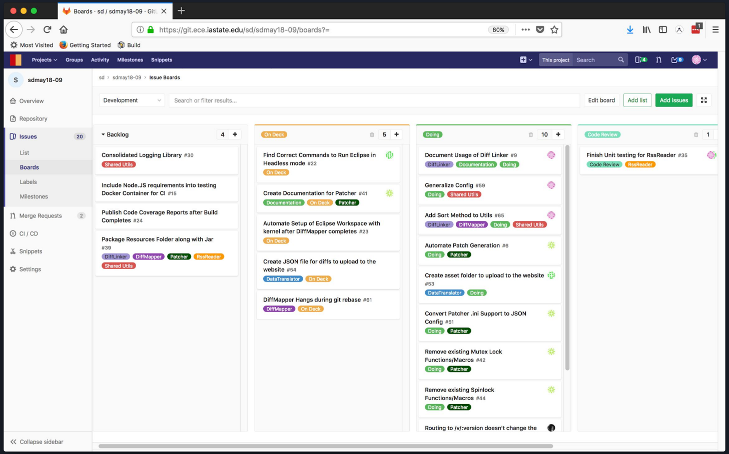


Status	Pipeline	Commit	Stages	
passed	#2177 by latest	patcher_cli 8eb553fd Added Patcher Main and ...	✓	00:02:01 less than a minute ago
passed	#2176 by latest	master 2b6b60da Merge branch 'issue_25' i...	✓	00:02:31 2 minutes ago
passed	#2175 by latest	patcher_ove... 993c4309 Changed testing .yml	✓	00:02:34 3 minutes ago
failed	#2174 by latest	issue_59 b2e4af24 Implement Generalized C...	✗	00:01:48 6 minutes ago
passed	#2173 by latest	site-dev b1e257f7 added more filtering para...	✓	00:02:12 8 minutes ago
passed	#2172 by latest	issues_3_an... a7cee863 Merge branch 'master' of ...	✓	00:02:09 9 minutes ago
passed	#2171 by latest	master 2b6b60da Merge branch 'issue_25' i...	✓	00:02:26 11 minutes ago
	#2160 by	issue_53 c0af06ab		

Project Management

Project Management

- GitLab Issues
  - Tags
  - Issue boards
- Merge Requests
  - Merge requirements
    - Approval
    - Pass CI pipeline
    - Doesn't decrease testing coverage
- Risks averted
  - Change in L-SAP's output structure
  - Memory footprint of modules kept low



Backlog 4

- Consolidated Logging Library #30 — Shared Utils
- Include Node.JS requirements into testing Docker Container for CI #15
- Publish Code Coverage Reports after Build Completes #24
- Package Resources Folder along with Jar #39 — DiffLinker, DiffMapper, Patcher, RssReader, Shared Utils

On Deck 5

- Find Correct Commands to Run Eclipse in Headless mode #22 — On Deck
- Create Documentation for Patcher #41 — Documentation, On Deck, Patcher
- Automate Setup of Eclipse Workspace with kernel after DiffMapper completes #23 — On Deck
- Create JSON file for diffs to upload to the website #54 — DataTranslator, On Deck
- DiffMapper Hangs during git rebase #61 — DiffMapper, On Deck

Doing 10

- Document Usage of Diff Linker #9 — DiffLinker, Documentation, Doing
- Generalize Config #59 — Doing, Shared Utils
- Add Sort Method to Utils #65 — DiffLinker, DiffMapper, Doing, Shared Utils
- Automate Patch Generation #6 — Doing, Patcher
- Create asset folder to upload to the website #53 — DataTranslator, Doing
- Convert Patcher .ini Support to JSON Config #51 — Doing, Patcher
- Remove existing Mutex Lock Functions/Macros #42 — Doing, Patcher
- Remove existing Spinlock Functions/Macros #44 — Doing, Patcher
- Routing to /v/:version doesn't change the

Code Review 1

- Finish Unit testing for RssReader #35 — Code Review, RssReader

Results

Time Comparison

	Manual Pipeline	Automated Pipeline
Detect and download new kernel release	~3 minutes + time to notice when a new release is available	~3 minutes
Creating patch for L-SAP	45-60 minutes	~ 1 second
Setup and run L-SAP	~18 hours	~18 hours
Generate difference reports	Infeasible	~10 minutes
Total time outside L-SAP	~1 hour + time between kernel release and starting	~13 minutes

Summary

Summary

- L-SAP: Verifies the Kernel and Generates Human Readable Evidence
    - Manual setup is tedious and provides no support for linking instances between versions
- Our Solution solves those problems and automates the process
    - Patch Creation and Application
    - Difference Mapping and Summarization
    - User-Friendly Website to aid in Understanding/Analyzing Results
    - Automation of Patching, Verification and Differencing
- Project Management
- Testing Framework is integrated into development cycle
    - Initial Tests with Kernel subset are promising

Questions?

Visit:
https://lsap.knowledgecentricsoftwarelab.com/