Cyber Grand Shellphish

SHELLPHISH

POC 2016

seclab

THE COMPUTER SECURITY GROUP AT UC SANTA BARBARA

SHELLPHISH

HEX on the beach

Shellphish?

- Founded in 2004
- Oldest? Capture the Flag team around
- Semi-successful
    - Won DEFCON CTF 2005
    - Qualified for DEFCON CTF every year but 2007 or so
- Hackademics at heart

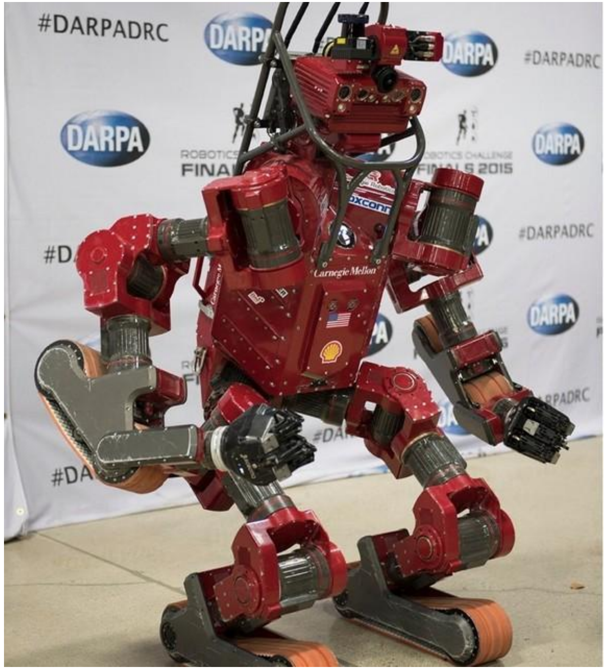
CYBER

GRAND_CHALLENGE

DARPA Grand Challenges

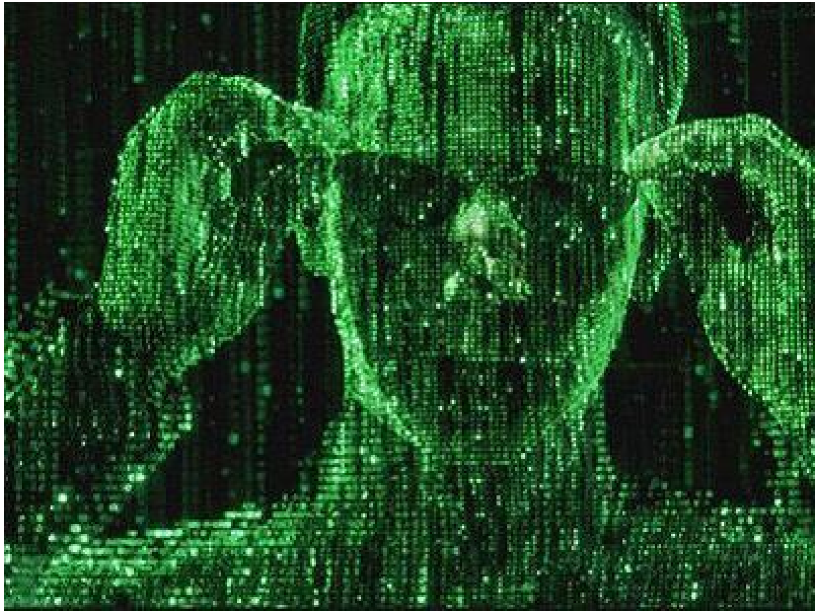
Self-driving Cars

Robots



DARPA Cyber Grand Challenge

Programs!



Past: Manual Vulnerability Analysis

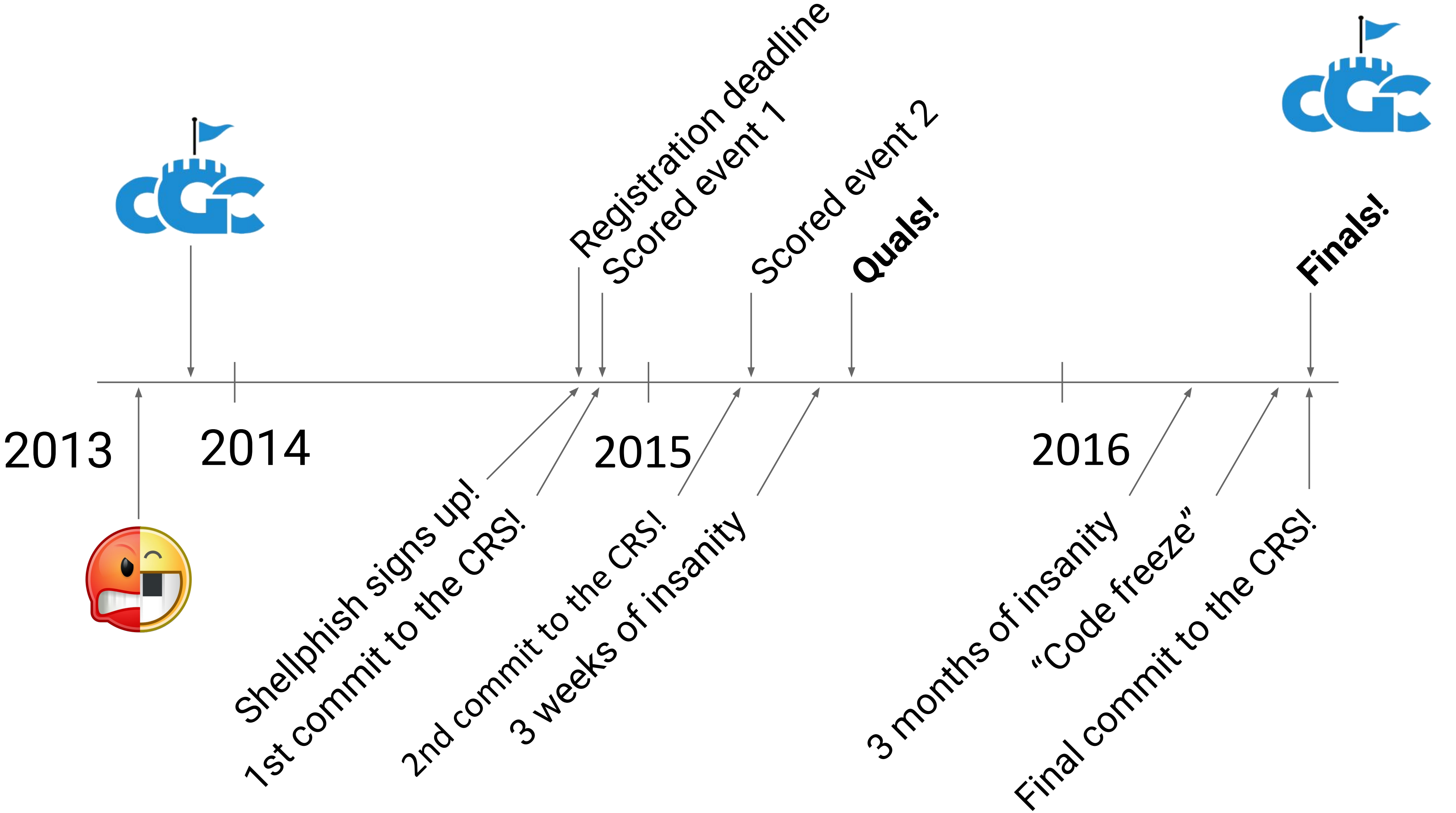
- "Look at the code and see what you can find"
- Requires substantial expertise
  - Analysis is as good as the person performing it
- Allows for the identification of complex vulnerabilities
- Expensive, does not scale

Current: Tool-assisted Vuln Analysis

- "Run these tools and verify/expand the results"
- Tools help in identifying areas of interest
  - Ruling out known code
  - Identifying potential vulnerabilities
  - Generate shellcode
- A human is involved, expertise and scale are still issues

Future: Automated Vuln Analysis

- "Run this tool and it will find the vulnerability"
  - … and generate an exploit
  - … and generate a patch
- Requires well-defined models for the vulnerabilities
- Which vulnerabilities? Must be modeled
- How to scale?
- The problem with halting…



2013

2014

2015

2016

Shellphish signs up!

1st commit to the CRS!

Registration deadline

Scored event 1

2nd commit to the CRS!

Scored event 2

3 weeks of insanity

Quals!

3 months of insanity

“Code freeze”

Final commit to the CRS!

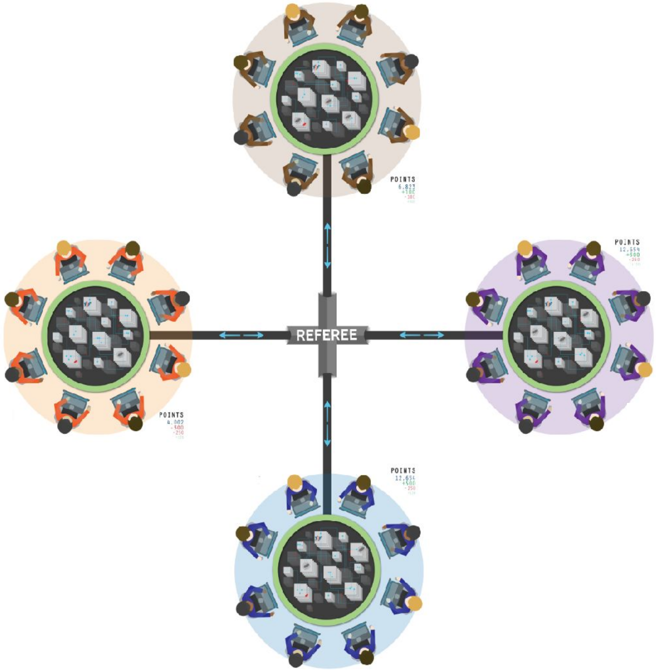
Finals!

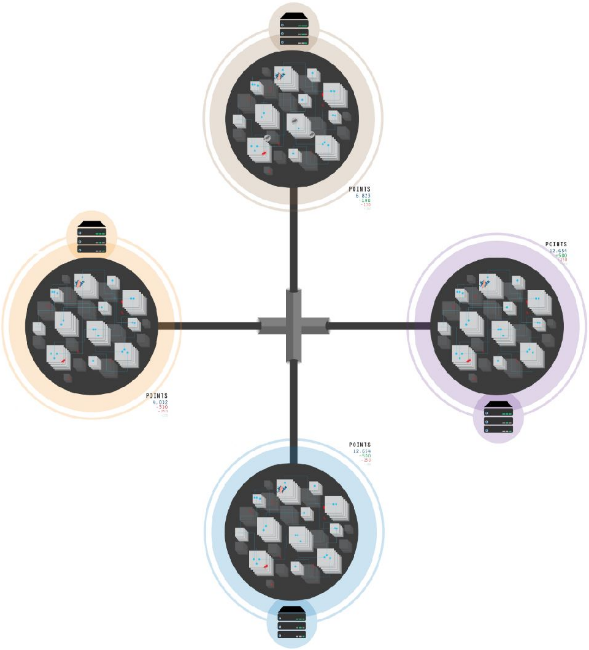
SHELLPHISH

🇮🇹 🇩🇪 🇮🇹 🇮🇹 🇺🇸 🇮🇹 🇮🇳 🇺🇸 🇺🇸 🇺🇸 🇨🇳

CRSPY
disekt
CYBER
GALACTICA
CodeJitsu
JIMA
CSDS
RUBEUS
CRSPY
SPARRING
ZANDRA
MECHANICAL PHISH
MAYHEM
SECURITY
DARPA

DARPA Cyber Grand Challenge





Organized as a Attack/Defense CTF

- Leverage CTF style to advance science
- Completely autonomous
- No team to team traffic
- Patches and exploits fielded through API (TI)
- Network traffic available via tap
- Big $$$ on the line (3.75 million USD)
  - Lots of clarifications needed (68 pages of FAQ)
  - Roboter over air-gap to transfer data one-way (out)

CGC
CYBER
GRAND CHALLENGE
SECURITY
MIND THE
AIR GAP
DATA OUT
DATA OUT
DARPA
POWER
POWER

analyze

pwn

patch

Analyze

- DECREE is Linux-inspired environment, only 7 syscalls
  - `transmit / receive / fdwait (≈select)`
  - `allocate / deallocate`
  - `random`
  - `terminate`
- No need to model the POSIX API
- Otherwise real(istic) programs!

Pwn

- No filesystem → No flag?
- CGC Quals: Crash = Exploit
- CGC Finals: Two types of exploits
  - "flag overwrite": set a register to X, crash at Y
  - "flag read": leak the "secret flag" from memory

Patch

~~int main() { return 0; }~~

fails functionality checks...

~~signal(SIGSEGV, exit)~~

no signal handling!

~~inline QEMU-based CFI?~~

performance penalties...

CGC Final Event (CFE)

- Divided in 96 rounds, with short breaks between rounds
- API access to Challenge Binaries (CBs) for teams’ CRSs
    - Each CB provides a service (e.g., an HTTP server)
    - Initially, all teams run the same binaries for each service
- Each round: score for each unique (team, service) tuple

CGC Final Event (CFE)

$$\sum_{i=0}^{\#CB} Availability \times Security \times Evaluation$$

- Availability: how badly did you f*ck up the binary?
- Security: did you defend against *all* exploits?
- Evaluation: how many teams did you pwn?

Mechanical Phish (CFE)

Our Cyber Reasoning System (CRS)

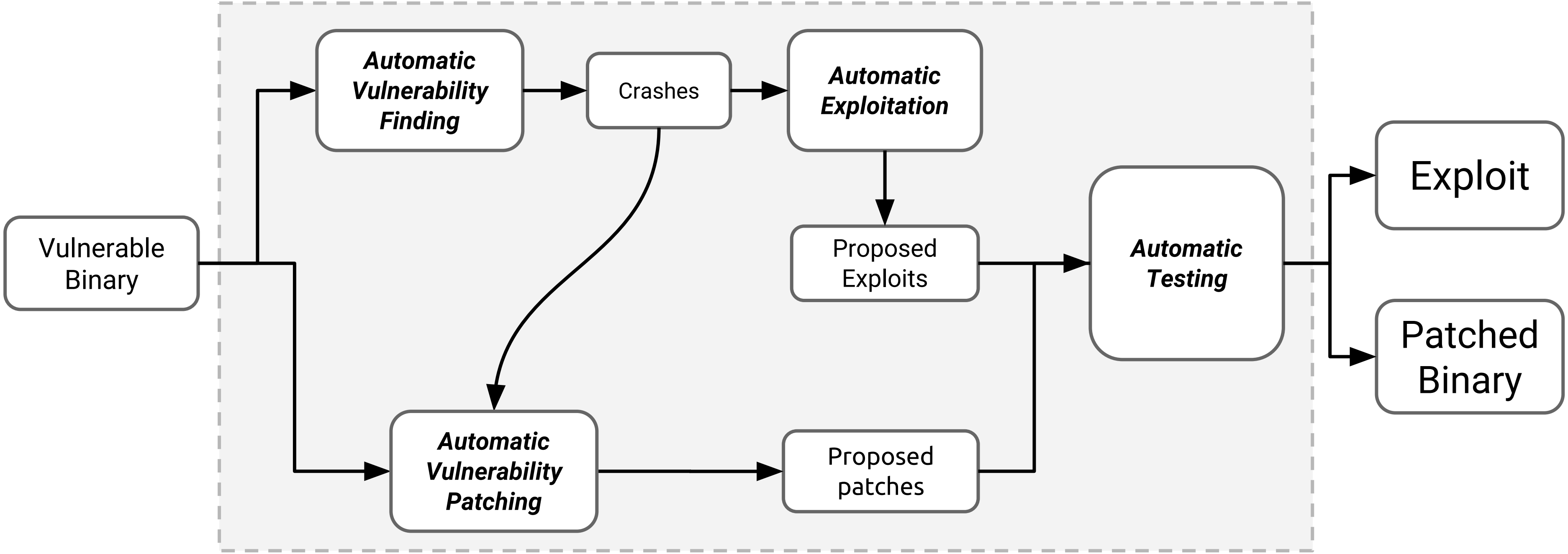
Completely autonomous system

- Patch
- Crash
- Exploit

Computational resources provided:

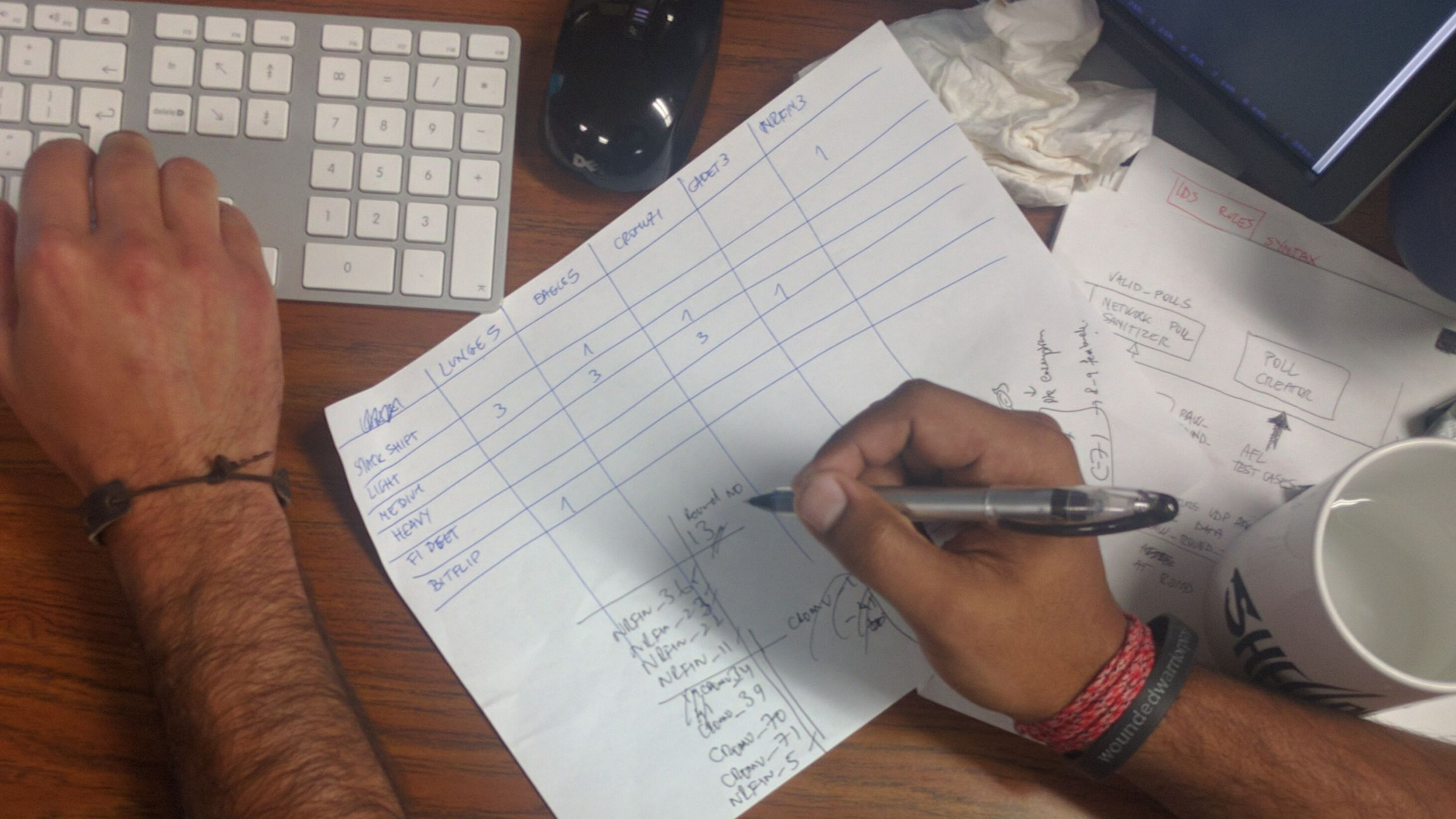
- 1,280 cores; 16TB memory

Mechanical Phish (CFE)



Challenges

- Infrastructure Availability
    - (Almost) No event can cause a catastrophic downtime
- Analysis Scalability
    - Efficiently and autonomously directing fuzzing and state exploration
- Performance/Security Trade-off
    - Many patches, many approaches: which patched binary to field?



	LUNGES	EAGLES	CROMU71	CADET3	NRFIN3
STACK SHIFT					1
LIGHT	3	1			
MEDIUM		3	1		
HEAVY			3	1	
FI DBET					
BITFLIP	1				

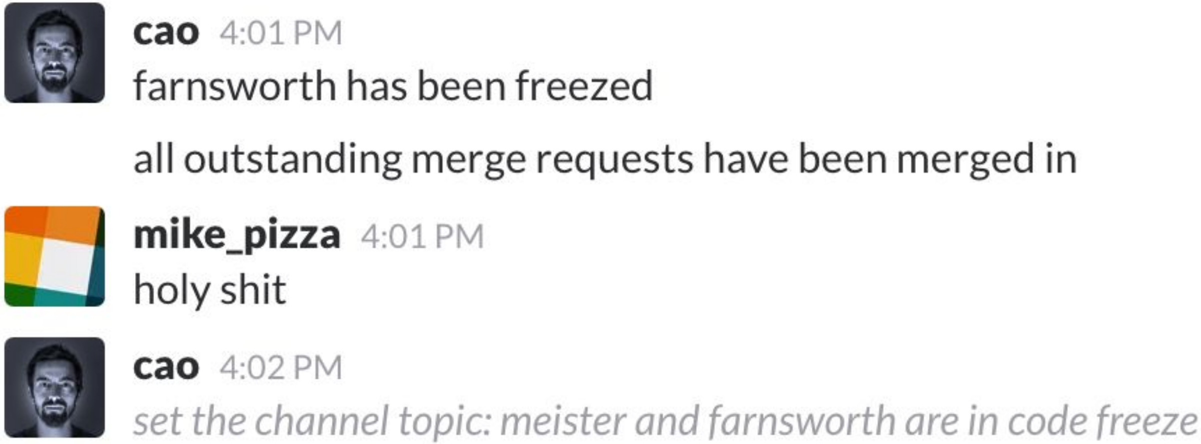
found NO
13

NRFIN_36
NRFIN_23
NRFIN_32
NRFIN_11

CROMU

CROMU_14
CROMU_39
CROMU_70
CROMU_71
NRFIN_5

Code Freeze

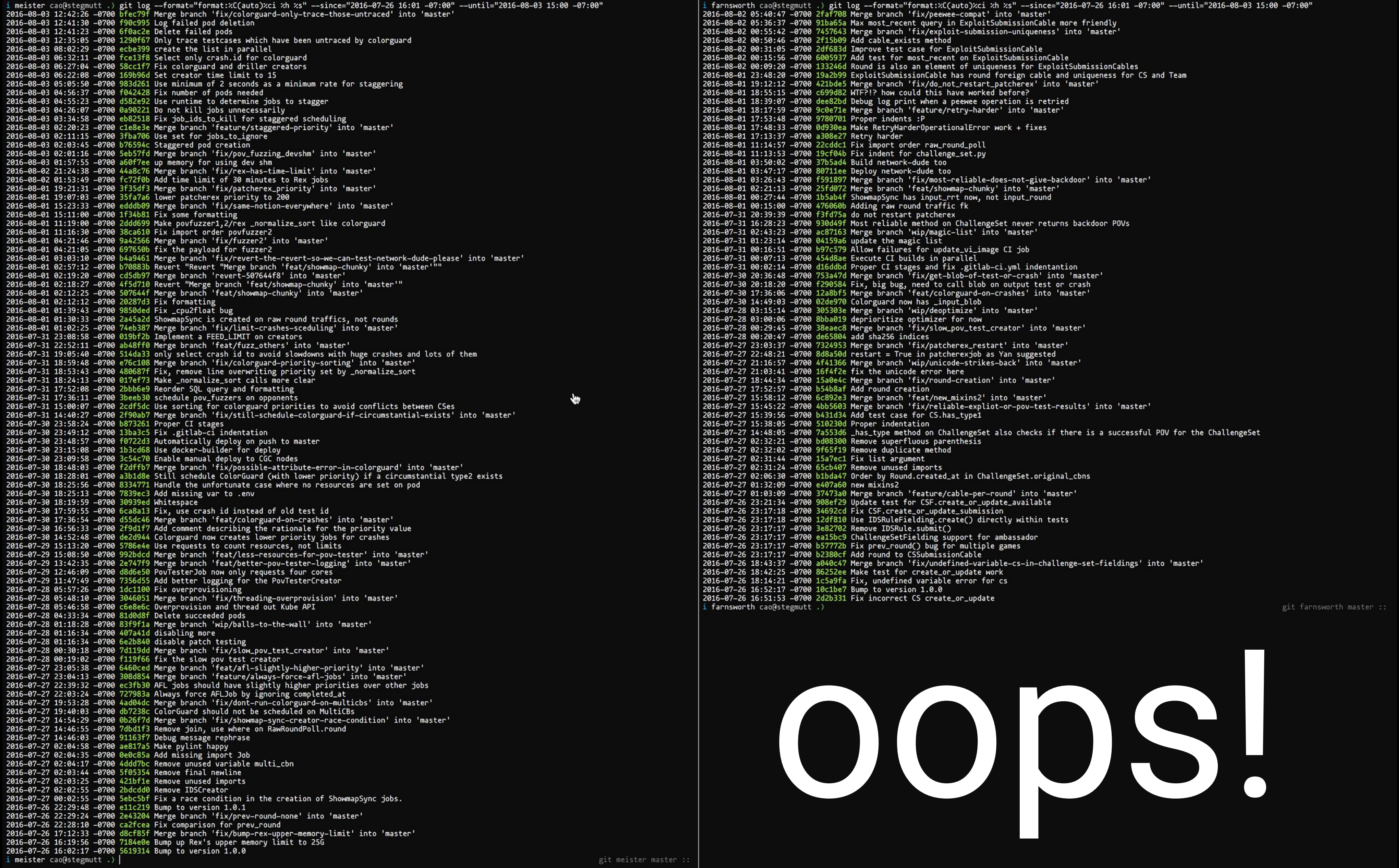


cao 4:01 PM
farnsworth has been freezed

all outstanding merge requests have been merged in

mike_pizza 4:01 PM
holy shit

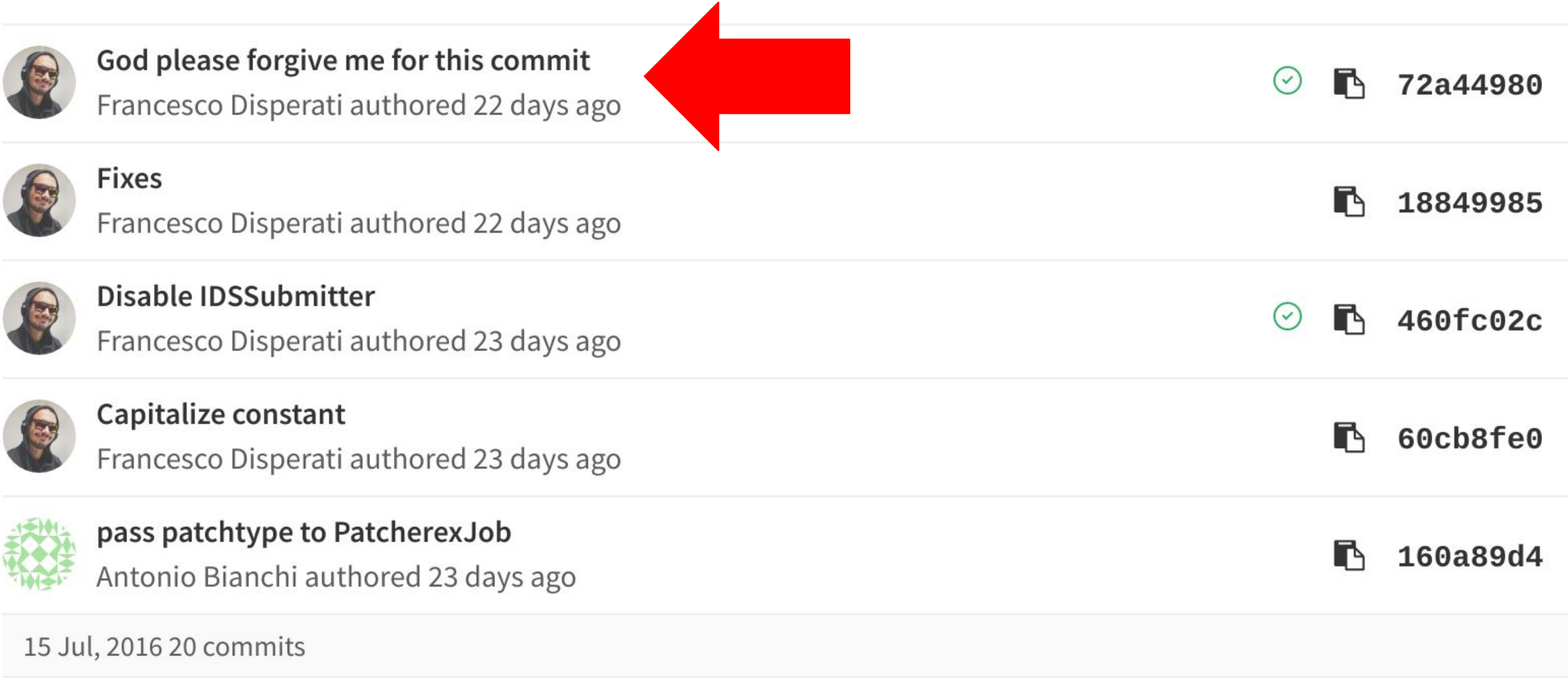
cao 4:02 PM
set the channel topic: meister and farnsworth are in code freeze



```
i meister cao@stegmutt .) git log --format="format:%C(auto)%ci %h %s" --since="2016-07-26 16:01 -07:00" --until="2016-08-03 15:00 -07:00"
2016-08-03 12:42:26 -0700 bfec79f Merge branch 'fix/colorguard-only-trace-those-untraced' into 'master'
2016-08-03 12:41:30 -0700 f90c995 Log failed pod deletion
2016-08-03 12:41:23 -0700 6f0ac2e Delete failed pods
2016-08-03 12:35:05 -0700 1290f67 Only trace testcases which have been untraced by colorguard
2016-08-03 08:02:29 -0700 ecbe399 create the list in parallel
2016-08-03 06:32:11 -0700 fce13f8 Select only crash.id for colorguard
2016-08-03 06:27:04 -0700 58cc1f7 Fix colorguard and driller creators
2016-08-03 06:22:08 -0700 169b96d Set creator time limit to 15
2016-08-03 05:05:50 -0700 983d261 Use minimum of 2 seconds as a minimum rate for staggering
2016-08-03 04:56:37 -0700 f042428 Fix number of pods needed
2016-08-03 04:55:23 -0700 d582e92 Use runtime to determine jobs to stagger
2016-08-03 04:26:07 -0700 0a90221 Do not kill jobs unnecessarily
2016-08-03 03:34:58 -0700 eb82518 Fix job_ids_to_kill for staggered scheduling
2016-08-03 02:20:23 -0700 c1e8e3e Merge branch 'feature/staggered-priority' into 'master'
2016-08-03 02:11:15 -0700 3fba706 Use set for jobs_to_ignore
2016-08-03 02:03:45 -0700 b76594c Staggered pod creation
2016-08-03 02:01:16 -0700 5eb57fd Merge branch 'fix/pov_fuzzing_devshm' into 'master'
2016-08-03 01:57:55 -0700 a60f7ee up memory for using dev shm
2016-08-02 21:24:38 -0700 44a8c76 Merge branch 'fix/rex-has-time-limit' into 'master'
2016-08-02 01:53:49 -0700 fc72f0b Add time limit of 30 minutes to Rex jobs
2016-08-01 19:21:31 -0700 3f35df3 Merge branch 'fix/patcherex_priority' into 'master'
2016-08-01 19:07:03 -0700 35fa7a6 lower patcherex priority to 200
2016-08-01 15:23:33 -0700 edddb09 Merge branch 'fix/same-notion-everywhere' into 'master'
2016-08-01 15:11:00 -0700 1f34b81 Fix some formatting
2016-08-01 11:19:00 -0700 2ddd699 Make povfuzzer1,2/rex _normalize_sort like colorguard
2016-08-01 11:16:30 -0700 38ca610 Fix import order povfuzzer2
2016-08-01 04:21:46 -0700 9a42566 Merge branch 'fix/fuzzer2' into 'master'
2016-08-01 04:21:05 -0700 697650b fix the payload for fuzzer2
2016-08-01 03:03:10 -0700 b4a9461 Merge branch 'fix/revert-the-revert-so-we-can-test-network-dude-please' into 'master'
2016-08-01 02:57:12 -0700 b70883b Revert "Revert "Merge branch 'feat/showmap-chunky' into 'master'""
2016-08-01 02:19:20 -0700 cd5db97 Merge branch 'revert-507644f8' into 'master'
2016-08-01 02:18:27 -0700 4f5d710 Revert "Merge branch 'feat/showmap-chunky' into 'master'"
2016-08-01 02:12:25 -0700 507644f Merge branch 'feat/showmap-chunky' into 'master'
2016-08-01 02:12:12 -0700 20287d3 Fix formatting
2016-08-01 01:39:43 -0700 9850ded Fix _cpu2float bug
2016-08-01 01:30:33 -0700 2a45a2d ShowmapSync is created on raw round traffics, not rounds
2016-08-01 01:02:25 -0700 74eb387 Merge branch 'fix/limit-crashes-sceduling' into 'master'
2016-07-31 23:08:58 -0700 019bf2b Implement a FEED_LIMIT on creators
2016-07-31 22:52:11 -0700 ab48ff0 Merge branch 'feat/fuzz_others' into 'master'
2016-07-31 19:05:40 -0700 514da33 only select crash id to avoid slowdowns with huge crashes and lots of them
2016-07-31 18:59:48 -0700 e76c108 Merge branch 'fix/colorguard-priority-sorting' into 'master'
2016-07-31 18:53:43 -0700 480687f Fix, remove line overwriting priority set by _normalize_sort
2016-07-31 18:24:13 -0700 017ef73 Make _normalize_sort calls more clear
2016-07-31 17:52:08 -0700 2bbb6e9 Reorder SQL query and formatting
2016-07-31 17:36:11 -0700 3beeb30 schedule pov_fuzzers on opponents
2016-07-31 15:00:07 -0700 2cdf5dc Use sorting for colorguard priorities to avoid conflicts between CSes
2016-07-31 14:40:27 -0700 2f90ab7 Merge branch 'fix/still-schedule-colorguard-if-circumstantial-exists' into 'master'
2016-07-30 23:58:24 -0700 b873261 Proper CI stages
2016-07-30 23:49:12 -0700 13ba3c5 Fix .gitlab-ci indentation
2016-07-30 23:48:57 -0700 f0722d3 Automatically deploy on push to master
2016-07-30 23:15:08 -0700 1b3cd68 Use docker-builder for deploy
2016-07-30 23:09:58 -0700 3c54c70 Enable manual deploy to CGC nodes
2016-07-30 18:48:03 -0700 f2dffb7 Merge branch 'fix/possible-attribute-error-in-colorguard' into 'master'
2016-07-30 18:28:01 -0700 a3b1d8e Still schedule ColorGuard (with lower priority) if a circumstantial type2 exists
2016-07-30 18:25:56 -0700 8334771 Handle the unfortunate case where no resources are set on pod
2016-07-30 18:25:13 -0700 7839ec3 Add missing var to .env
2016-07-30 18:19:59 -0700 30939ed Whitespace
2016-07-30 17:59:55 -0700 6ca8a13 Fix, use crash id instead of old test id
2016-07-30 17:36:54 -0700 d55dc46 Merge branch 'feat/colorguard-on-crashes' into 'master'
2016-07-30 16:56:33 -0700 2f9d1f7 Add comment describing the rationale for the priority value
2016-07-30 14:52:48 -0700 de2d944 Colorguard now creates lower priority jobs for crashes
2016-07-29 15:13:20 -0700 5786e4e Use requests to count resources, not limits
2016-07-29 15:08:50 -0700 992bdcd Merge branch 'feat/less-resources-for-pov-tester' into 'master'
2016-07-29 13:42:35 -0700 2e747f9 Merge branch 'feat/better-pov-tester-logging' into 'master'
2016-07-29 12:46:09 -0700 d8d6e50 PovTesterJob now only requests four cores
2016-07-29 11:47:49 -0700 7356d55 Add better logging for the PovTesterCreator
2016-07-28 05:57:26 -0700 1dc1100 Fix overprovisioning
2016-07-28 05:48:10 -0700 3046051 Merge branch 'fix/threading-overprovision' into 'master'
2016-07-28 05:46:58 -0700 c6e8e6c Overprovision and thread out Kube API
2016-07-28 04:33:34 -0700 81d0d8f Delete succeeded pods
2016-07-28 01:18:28 -0700 83f9f1a Merge branch 'wip/balls-to-the-wall' into 'master'
2016-07-28 01:16:34 -0700 407a41d disabling more
2016-07-28 01:16:34 -0700 6e2b840 disable patch testing
2016-07-28 00:30:18 -0700 7d119dd Merge branch 'fix/slow_pov_test_creator' into 'master'
2016-07-28 00:19:02 -0700 f119f66 fix the slow pov test creator
2016-07-27 23:05:38 -0700 6460ced Merge branch 'feat/afl-slightly-higher-priority' into 'master'
2016-07-27 23:04:13 -0700 308d854 Merge branch 'feature/always-force-afl-jobs' into 'master'
2016-07-27 22:39:32 -0700 ec3fb30 AFL jobs should have slightly higher priorities over other jobs
2016-07-27 22:03:24 -0700 727983a Always force AFLJob by ignoring completed_at
2016-07-27 19:53:28 -0700 4ad04dc Merge branch 'fix/dont-run-colorguard-on-multicbs' into 'master'
2016-07-27 19:40:03 -0700 db7238c ColorGuard should not be scheduled on MultiCBs
2016-07-27 14:54:29 -0700 0b26f7d Merge branch 'fix/showmap-sync-creator-race-condition' into 'master'
2016-07-27 14:46:55 -0700 7dbd1f3 Remove join, use where on RawRoundPoll.round
2016-07-27 14:46:03 -0700 91163f7 Debug message rephrase
2016-07-27 02:04:58 -0700 ae817a5 Make pylint happy
2016-07-27 02:04:35 -0700 0e0c85a Add missing import Job
2016-07-27 02:04:17 -0700 4ddd7bc Remove unused variable multi_cbn
2016-07-27 02:03:44 -0700 5f05354 Remove final newline
2016-07-27 02:03:25 -0700 421bf1e Remove unused imports
2016-07-27 02:02:55 -0700 2bdcdd0 Remove IDSCreator
2016-07-27 00:02:55 -0700 5ebc5bf Fix a race condition in the creation of ShowmapSync jobs.
2016-07-26 22:29:48 -0700 e11c219 Bump to version 1.0.1
2016-07-26 22:29:24 -0700 2e43204 Merge branch 'fix/prev-round-none' into 'master'
2016-07-26 22:28:10 -0700 ca2fcea Fix comparison for prev_round
2016-07-26 17:12:33 -0700 d8cf85f Merge branch 'fix/bump-rex-upper-memory-limit' into 'master'
2016-07-26 16:19:56 -0700 7184e0e Bump up Rex's upper memory limit to 25G
2016-07-26 16:02:17 -0700 5619314 Bump to version 1.0.0
i meister cao@stegmutt .) |                                                                          git meister master ::
```

```
i farnsworth cao@stegmutt .) git log --format="format:%C(auto)%ci %h %s" --since="2016-07-26 16:01 -07:00" --until="2016-08-03 15:00 -07:00"
2016-08-02 05:40:47 -0700 2faf708 Merge branch 'fix/peewee-compat' into 'master'
2016-08-02 05:36:37 -0700 91ba65a Max most_recent query in ExploitSubmissionCable more friendly
2016-08-02 00:55:42 -0700 7457643 Merge branch 'fix/exploit-submission-uniqueness' into 'master'
2016-08-02 00:50:46 -0700 2f15b09 Add cable_exists method
2016-08-02 00:31:05 -0700 2df683d Improve test case for ExploitSubmissionCable
2016-08-02 00:15:56 -0700 6005937 Add test for most_recent on ExploitSubmissionCable
2016-08-02 00:09:20 -0700 133246d Round is also an element of uniqueness for ExploitSubmissionCables
2016-08-01 23:48:20 -0700 19a2b99 ExploitSubmissionCable has round foreign cable and uniqueness for CS and Team
2016-08-01 19:12:12 -0700 421bde5 Merge branch 'fix/do_not_restart_patcherex' into 'master'
2016-08-01 18:55:15 -0700 c699d82 WTF?!? how could this have worked before?
2016-08-01 18:39:07 -0700 dee82bd Debug log print when a peewee operation is retried
2016-08-01 18:17:59 -0700 9c0e71e Merge branch 'feature/retry-harder' into 'master'
2016-08-01 17:53:48 -0700 9780701 Proper indents :P
2016-08-01 17:48:33 -0700 0d930ea Make RetryHarderOperationalError work + fixes
2016-08-01 17:13:37 -0700 a308e27 Retry harder
2016-08-01 11:14:57 -0700 22cddc1 Fix import order raw_round_poll
2016-08-01 11:13:53 -0700 19cf04b Fix indent for challenge_set.py
2016-08-01 03:50:02 -0700 37b5ad4 Build network-dude too
2016-08-01 03:47:17 -0700 80711ee Deploy network-dude too
2016-08-01 03:26:43 -0700 f591897 Merge branch 'fix/most-reliable-does-not-give-backdoor' into 'master'
2016-08-01 02:21:13 -0700 25fd072 Merge branch 'feat/showmap-chunky' into 'master'
2016-08-01 00:27:44 -0700 1b5ab4f ShowmapSync has input_rrt now, not input_round
2016-08-01 00:15:00 -0700 476060b Adding raw round traffic fk
2016-07-31 20:39:39 -0700 f3fd75a do not restart patcherex
2016-07-31 16:28:23 -0700 930d49f Most reliable method on ChallengeSet never returns backdoor POVs
2016-07-31 02:43:23 -0700 ac87163 Merge branch 'wip/magic-list' into 'master'
2016-07-31 01:23:14 -0700 04159a6 update the magic list
2016-07-31 00:16:51 -0700 b97c579 Allow failures for update_vi_image CI job
2016-07-31 00:07:13 -0700 454d8ae Execute CI builds in parallel
2016-07-31 00:02:14 -0700 d16ddbd Proper CI stages and fix .gitlab-ci.yml indentation
2016-07-30 20:36:48 -0700 753a47d Merge branch 'fix/get-blob-of-test-or-crash' into 'master'
2016-07-30 20:18:20 -0700 f290584 Fix, big bug, need to call blob on output test or crash
2016-07-30 17:36:06 -0700 12a8bf5 Merge branch 'feat/colorguard-on-crashes' into 'master'
2016-07-30 14:49:03 -0700 02de970 Colorguard now has _input_blob
2016-07-28 03:15:14 -0700 305303e Merge branch 'wip/deoptimize' into 'master'
2016-07-28 03:00:06 -0700 8bba019 deprioritize optimizer for now
2016-07-28 00:29:45 -0700 38eaec8 Merge branch 'fix/slow_pov_test_creator' into 'master'
2016-07-28 00:20:47 -0700 de65804 add sha256 indices
2016-07-27 23:03:37 -0700 7324953 Merge branch 'fix/patcherex_restart' into 'master'
2016-07-27 22:48:21 -0700 8d8a50d restart = True in patcherexjob as Yan suggested
2016-07-27 21:16:57 -0700 4f41366 Merge branch 'wip/unicode-strikes-back' into 'master'
2016-07-27 21:03:41 -0700 16f4f2e fix the unicode error here
2016-07-27 18:44:34 -0700 15a0e4c Merge branch 'fix/round-creation' into 'master'
2016-07-27 17:52:57 -0700 b54b8af Add round creation
2016-07-27 15:58:12 -0700 6c892e3 Merge branch 'feat/new_mixins2' into 'master'
2016-07-27 15:45:22 -0700 4bb5603 Merge branch 'fix/reliable-expliot-or-pov-test-results' into 'master'
2016-07-27 15:39:56 -0700 b431d34 Add test case for CS.has_type1
2016-07-27 15:38:05 -0700 510230d Proper indentation
2016-07-27 14:48:05 -0700 7a553d6 _has_type method on ChallengeSet also checks if there is a successful POV for the ChallengeSet
2016-07-27 02:32:21 -0700 bd08300 Remove superfluous parenthesis
2016-07-27 02:32:02 -0700 9f65f19 Remove duplicate method
2016-07-27 02:31:44 -0700 15a7ec1 Fix list argument
2016-07-27 02:31:24 -0700 65cb407 Remove unused imports
2016-07-27 02:06:30 -0700 b1bda47 Order by Round.created_at in ChallengeSet.original_cbns
2016-07-27 01:32:09 -0700 e407a60 new mixins2
2016-07-27 01:03:09 -0700 37473a0 Merge branch 'feature/cable-per-round' into 'master'
2016-07-26 23:21:34 -0700 908ef29 Update test for CSF.create_or_update_available
2016-07-26 23:17:18 -0700 34692cd Fix CSF.create_or_update_submission
2016-07-26 23:17:18 -0700 12df810 Use IDSRuleFielding.create() directly within tests
2016-07-26 23:17:17 -0700 3e82702 Remove IDSRule.submit()
2016-07-26 23:17:17 -0700 ea15bc9 ChallengeSetFielding support for ambassador
2016-07-26 23:17:17 -0700 b57772b Fix prev_round() bug for multiple games
2016-07-26 23:17:17 -0700 b2380cf Add round to CSSubmissionCable
2016-07-26 18:43:37 -0700 a040c47 Merge branch 'fix/undefined-variable-cs-in-challenge-set-fieldings' into 'master'
2016-07-26 18:42:25 -0700 86252ee Make test for create_or_update work
2016-07-26 18:14:21 -0700 1c5a9fa Fix, undefined variable error for cs
2016-07-26 16:52:17 -0700 10c1be7 Bump to version 1.0.0
2016-07-26 16:51:53 -0700 2d2b331 Fix incorrect CS create_or_update
i farnsworth cao@stegmutt .)                                                                             git farnsworth master ::
```

oops!



God please forgive me for this commit
Francesco Disperati authored 22 days ago ✓ 72a44980

Fixes
Francesco Disperati authored 22 days ago 18849985

Disable IDSSubmitter
Francesco Disperati authored 23 days ago ✓ 460fc02c

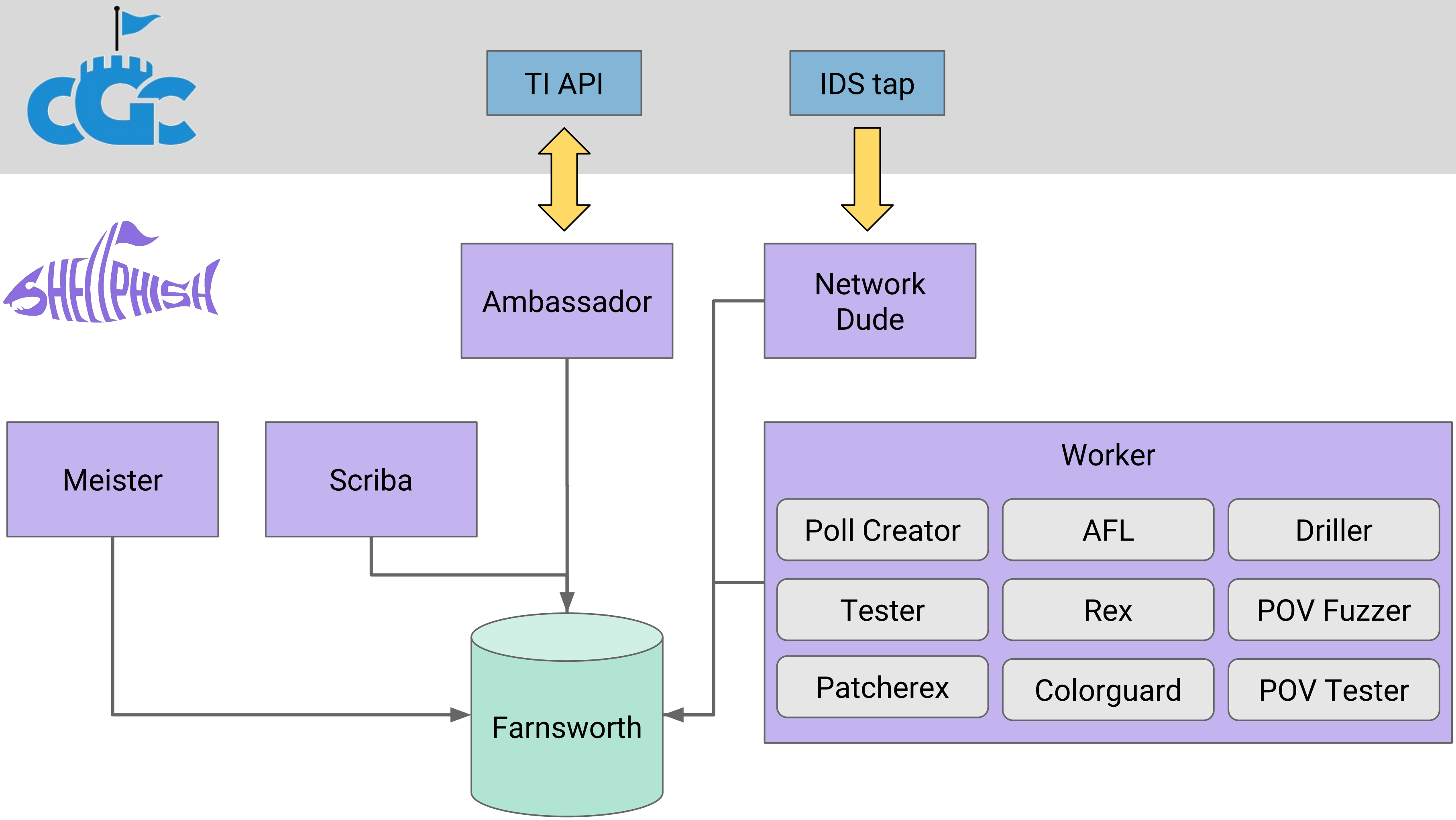
Capitalize constant
Francesco Disperati authored 23 days ago 60cb8fe0

pass patchtype to PatcherexJob
Antonio Bianchi authored 23 days ago 160a89d4

15 Jul, 2016 20 commits

Tue 2 Aug, 23:54

~15 hours before access shutdown


CGC
SHELLPHISH
TI API
IDS tap
Ambassador
Network Dude
Meister
Scriba
Worker
Poll Creator
AFL
Driller
Tester
Rex
POV Fuzzer
Patcherex
Colorguard
POV Tester
Farnsworth

Ambassador and Scriba

Ambassador:

Talk with Team Interface API

- Update Farnsworth

Scriba:

Submission Decision Maker

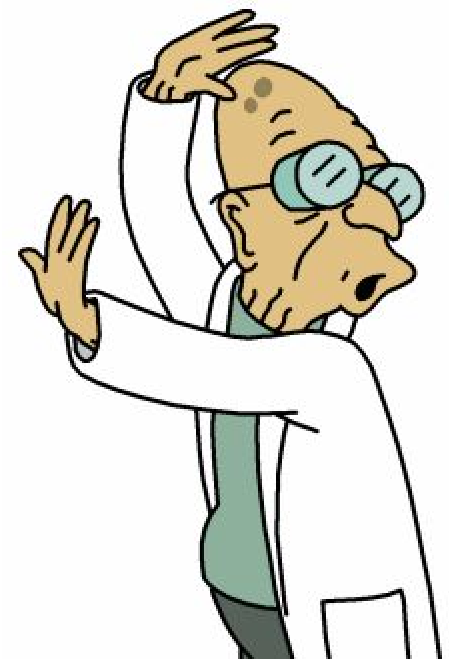
- Which exploit to launch against which team?
- Which patch to field?

Farnsworth

Object-relational model for database:

- What CS are fielded this round?
- Do we have crashes?
- Do we have a good patch?
- ...

Our ground truth and the only component reasonably well tested*

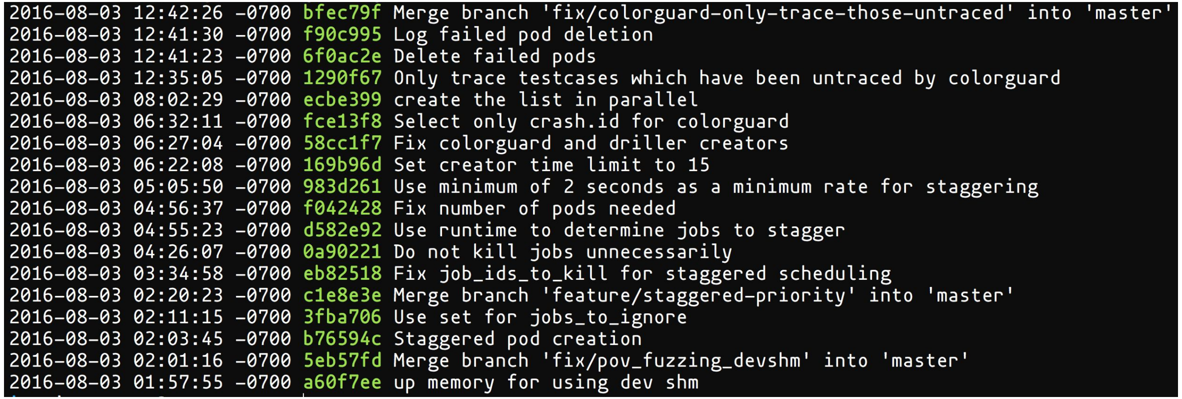


* 69% coverage

Meister

Job scheduler:

- Looks at game state
- Asks creators for jobs
- Schedules them based on priority



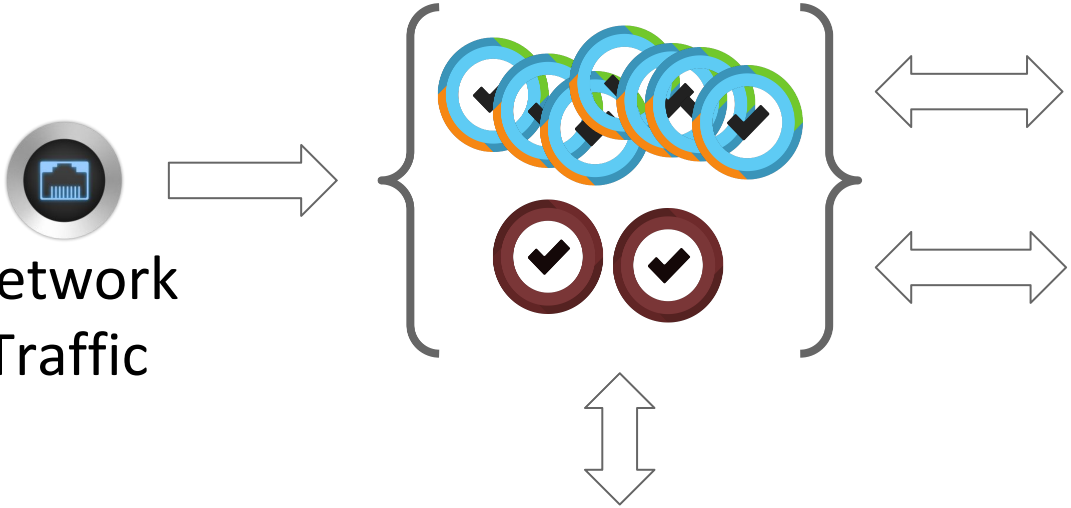
```
2016-08-03 12:42:26 -0700 bfec79f Merge branch 'fix/colorguard-only-trace-those-untraced' into 'master'
2016-08-03 12:41:30 -0700 f90c995 Log failed pod deletion
2016-08-03 12:41:23 -0700 6f0ac2e Delete failed pods
2016-08-03 12:35:05 -0700 1290f67 Only trace testcases which have been untraced by colorguard
2016-08-03 08:02:29 -0700 ecbe399 create the list in parallel
2016-08-03 06:32:11 -0700 fce13f8 Select only crash.id for colorguard
2016-08-03 06:27:04 -0700 58cc1f7 Fix colorguard and driller creators
2016-08-03 06:22:08 -0700 169b96d Set creator time limit to 15
2016-08-03 05:05:50 -0700 983d261 Use minimum of 2 seconds as a minimum rate for staggering
2016-08-03 04:56:37 -0700 f042428 Fix number of pods needed
2016-08-03 04:55:23 -0700 d582e92 Use runtime to determine jobs to stagger
2016-08-03 04:26:07 -0700 0a90221 Do not kill jobs unnecessarily
2016-08-03 03:34:58 -0700 eb82518 Fix job_ids_to_kill for staggered scheduling
2016-08-03 02:20:23 -0700 c1e8e3e Merge branch 'feature/staggered-priority' into 'master'
2016-08-03 02:11:15 -0700 3fba706 Use set for jobs_to_ignore
2016-08-03 02:03:45 -0700 b76594c Staggered pod creation
2016-08-03 02:01:16 -0700 5eb57fd Merge branch 'fix/pov_fuzzing_devshm' into 'master'
2016-08-03 01:57:55 -0700 a60f7ee up memory for using dev shm
```

angr

- Binary analysis framework developed at UC Santa Barbara
- Supports variety of architectures
  - x86, MIPS, ARM, PPC, etc. (all 32 and 64 bit)
- Open-source, free for commercial use (!)
  - http://angr.io
  - https://github.com/angr
  - angr@lists.cs.ucsb.edu

M 9686

How Do We Find Crashes?



Network Traffic

Symbolic Execution

Fuzzing

"Uncrasher"

Fuzzing

- Automated procedure to send inputs and record safety condition violations as crashes
  - Assumption: crashes are potentially exploitable
- Several dimensions in the fuzzing space
  - How to supply inputs to the program under test?
  - How to generate inputs?
  - How to find more “relevant” crashes?
  - How to change inputs between runs?
- Goal: Maximized effectiveness of the process

Fuzzing

```
x = int(input())
if x >= 10:
    if x < 100:
        print "You win!"
    else:
        print "You lose!"
else:
    print "You lose!"
```

Let's fuzz it!

1 ⇒ "You lose!"

593 ⇒ "You lose!"

4 ⇒ "You lose!"

498 ⇒ "You lose!"

42 ⇒ "You win!"

Fuzzing

```
x = int(input())
if x >= 10:
    if x^2 == 152399025:
        print "You win!"
    else:
        print "You lose!"
else:
    print "You lose!"
```

Let's fuzz it!

1 ⇒ "You lose!"

593 ⇒ "You lose!"

4 ⇒ "You lose!"

498 ⇒ "You lose!"

42 ⇒ "You lose!"

3 ⇒ "You lose!"

....

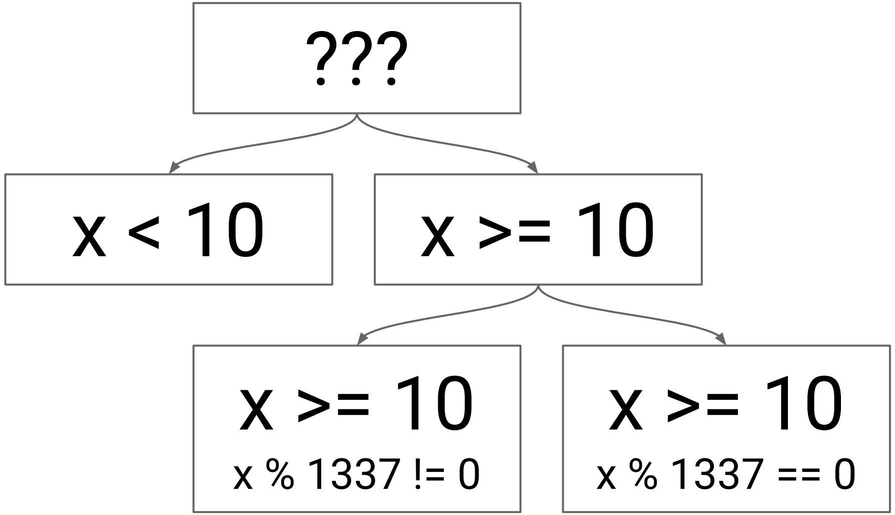
57 ⇒ "You lose!"

AFL

- Very fast!
- Very effective!
- Unable to deal with certain situations:
    - Magic numbers
    - Hashes
    - Specific identifiers

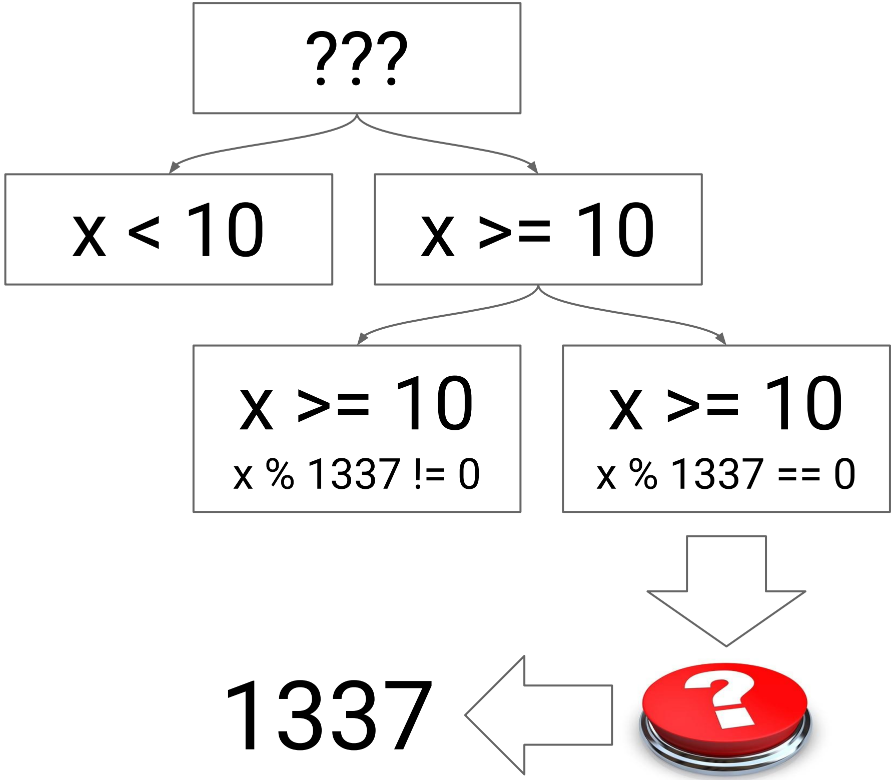
angr

```
x = input()
if x >= 10:
  if x % 1337 == 0:
    print "You win!"
  else:
    print "You lose!"
else:
  print "You lose!"
```

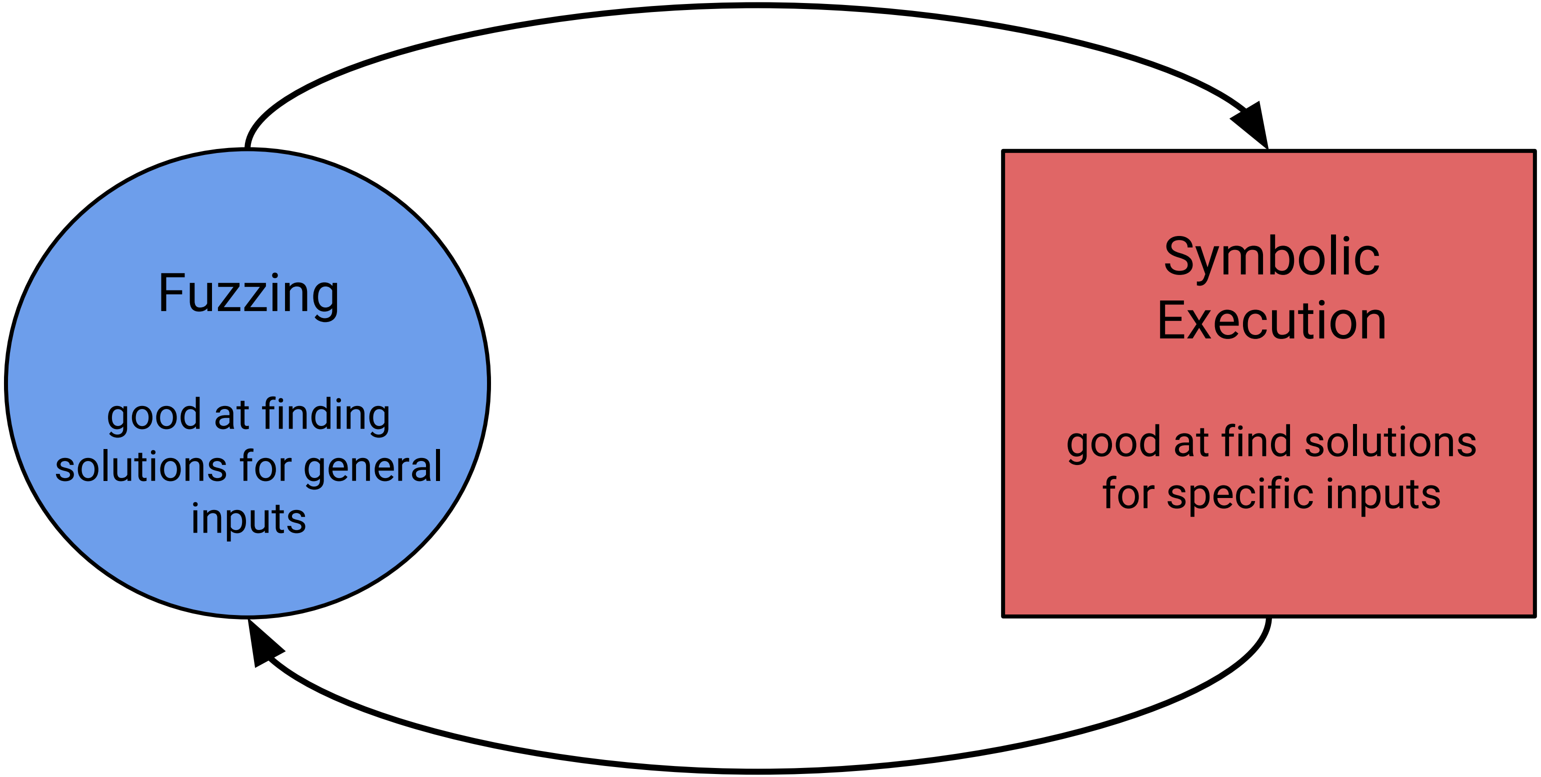


angr

```
x = input()
if x >= 10:
    if x % 1337 == 0:
        print "You win!"
    else:
        print "You lose!"
else:
    print "You lose!"
```



Driller = angr + AFL



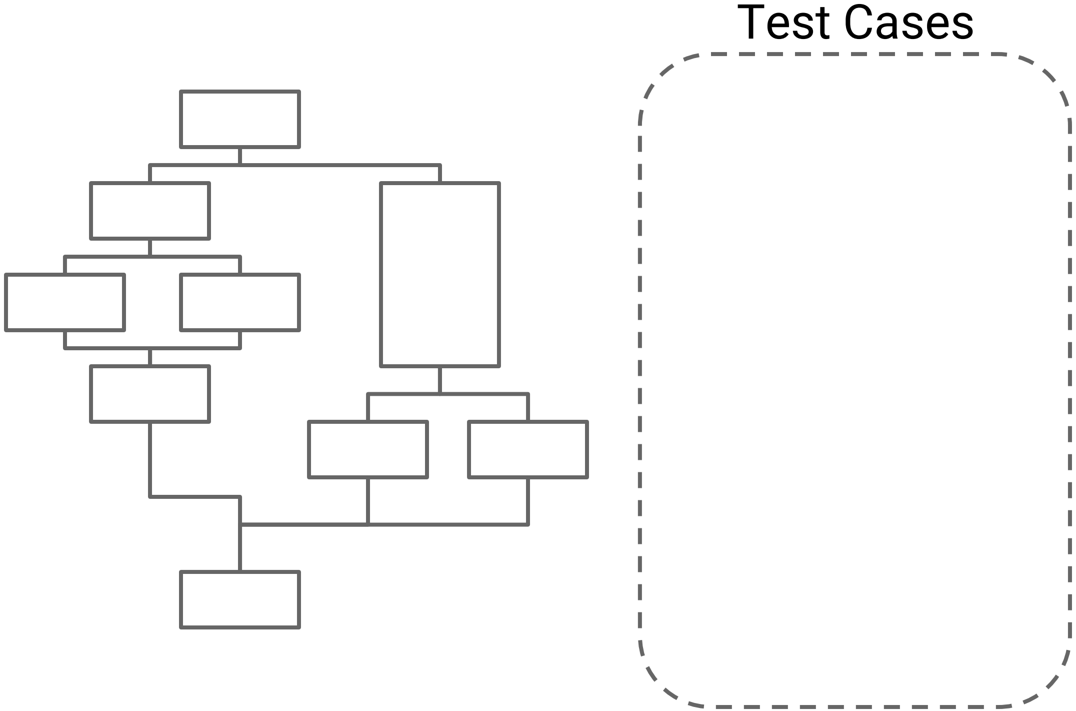
Fuzzing

good at finding solutions for general inputs

Symbolic Execution

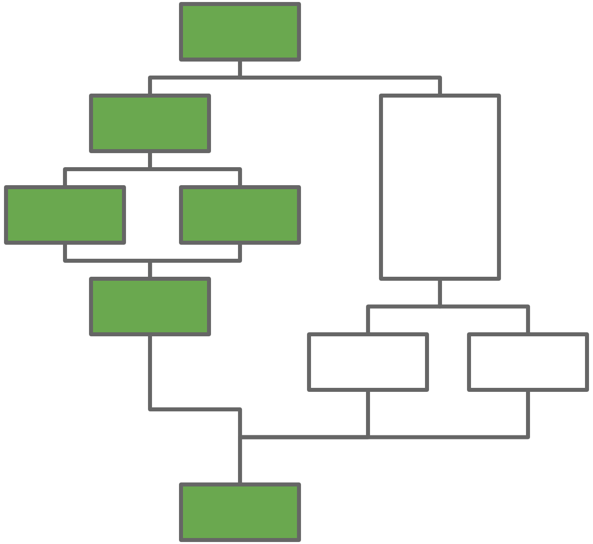
good at find solutions for specific inputs

Driller



Driller

"Cheap" fuzzing coverage

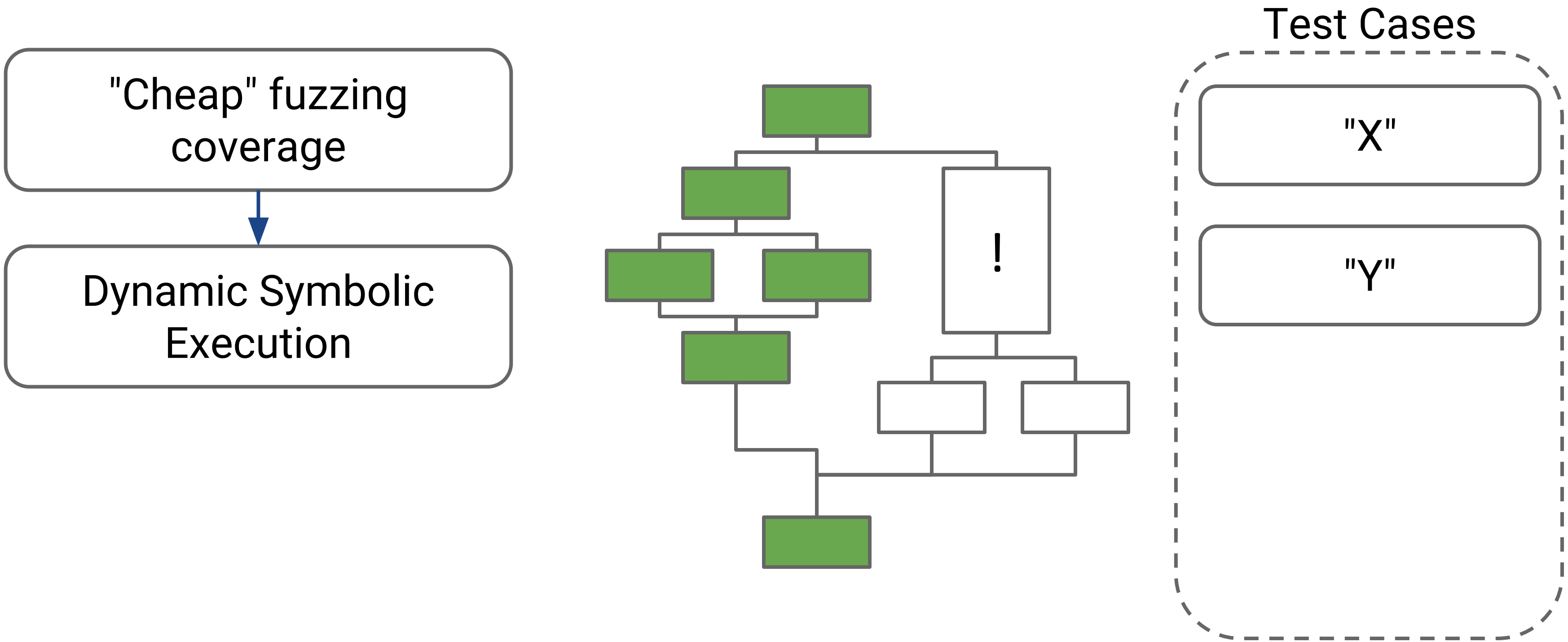


Test Cases

“X”

“Y”

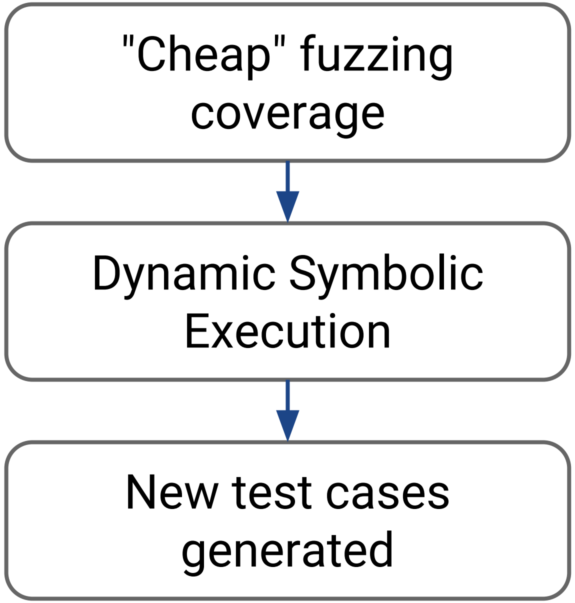
Driller



"Cheap" fuzzing coverage

Dynamic Symbolic Execution

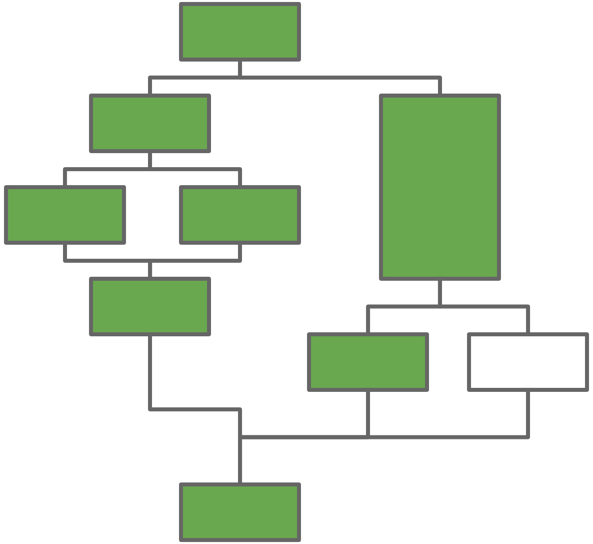
Driller



"Cheap" fuzzing coverage

Dynamic Symbolic Execution

New test cases generated



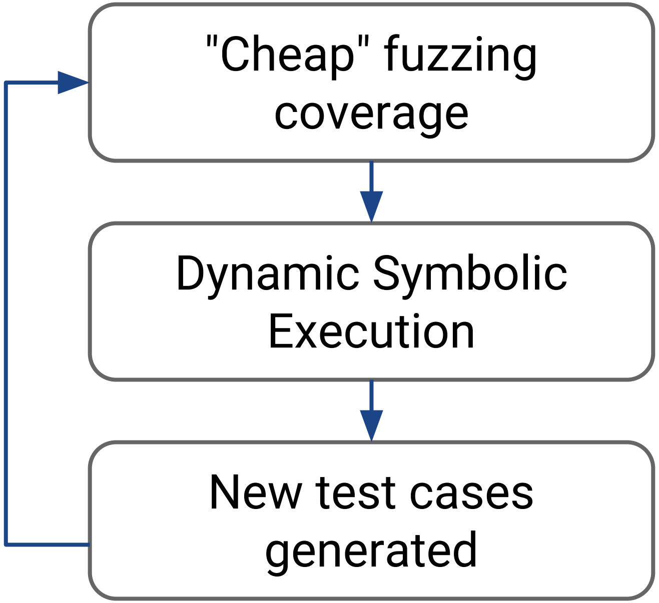
Test Cases

"X"

"Y"

"CGC_MAGIC"

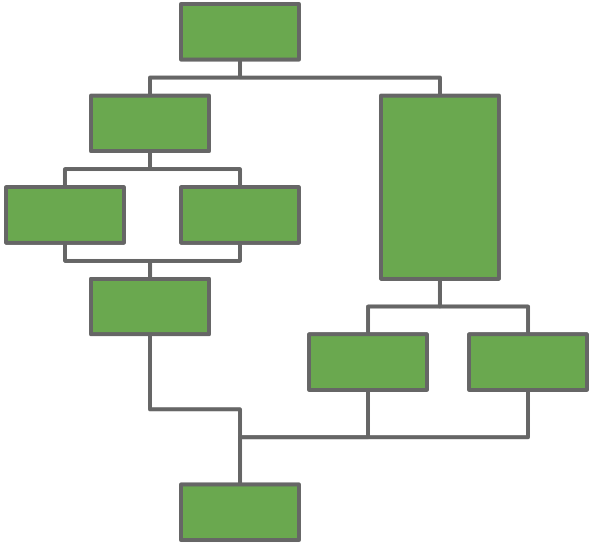
Driller



"Cheap" fuzzing coverage

Dynamic Symbolic Execution

New test cases generated



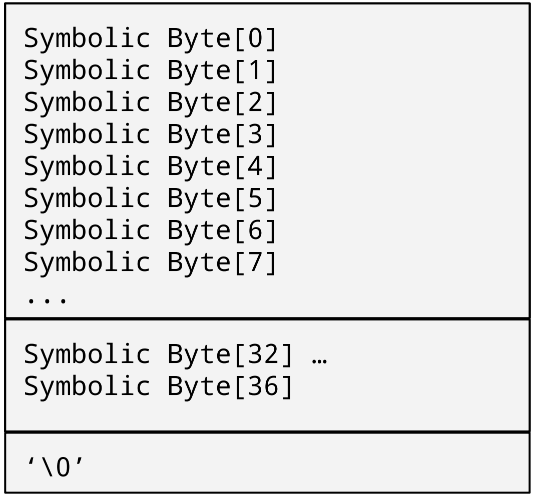
Test Cases

- "X"
- "Y"
- "CGC_MAGIC"
- "CGC_MAGICY"

Automatic Exploitation (Simplified)

```
typedef struct component {
     char name[32];
     int (*do_something)(int arg);
} comp_t;

comp_t *initialize_component(char *cmp_name) {
     int i = 0;
     struct component *cmp;

     cmp = malloc(sizeof(struct component));
     cmp->do_something = sample_func;

     while (*cmp_name)
          cmp->name[i++] = *cmp_name++;

     cmp->name[i] = ‘\0’;
     return cmp;
}
x = get_input();
cmp = initialize_component(x);
cmp->do_something(1);
```

HEAP



Symbolic Byte[0]
Symbolic Byte[1]
Symbolic Byte[2]
Symbolic Byte[3]
Symbolic Byte[4]
Symbolic Byte[5]
Symbolic Byte[6]
Symbolic Byte[7]
...

Symbolic Byte[32] …
Symbolic Byte[36]

‘\0’

call <symbolic byte[36:32]>

Automatic Exploitation (Simplified)

1. Turning the state into an **exploited** state

angr

```
assert state.se.symbolic(state.regs.pc)
```

2. Constrain **buffer** to contain our shellcode

angr

```
buf_addr = find_symbolic_buffer(state, len(shellcode))
mem = state.memory.load(buf_addr, len(shellcode))
state.add_constraints(mem == state.se.bvv(shellcode))
```

Automatic Exploitation (Simplified)

3. Constrain **PC** to point to the buffer

angr

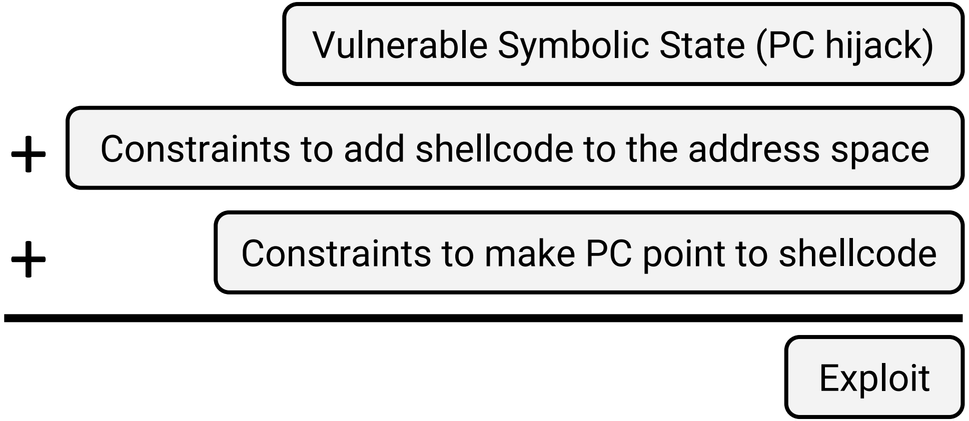
```
state.se.add_constraints(state.regs.pc == buf_addr)
```

4. **Synthesize**!

angr

```
exploit = state.posix.dumps(0)
```

Automatic Exploitation (Simplified)



Vulnerable Symbolic State (PC hijack)

+ Constraints to add shellcode to the address space

+ Constraints to make PC point to shellcode

Exploit

Exploit Techniques

- Circumstantial
- Shellcode
- ROP
- Arbitrary Read - Point to Flag
- Arbitrary Read/Write - Exploration
- Write-What-Where

Colorguard: Flag Page Leaks

- Make only the flag page symbolic
- Everything else is completely concrete
  - Significantly faster
  - Can execute most basic block with Unicorn
- When cores are idle on the CRS, trace all our test cases
- Solved DEFCON CTF LEGIT_00009 challenge

Patcherex

Patching Techniques:

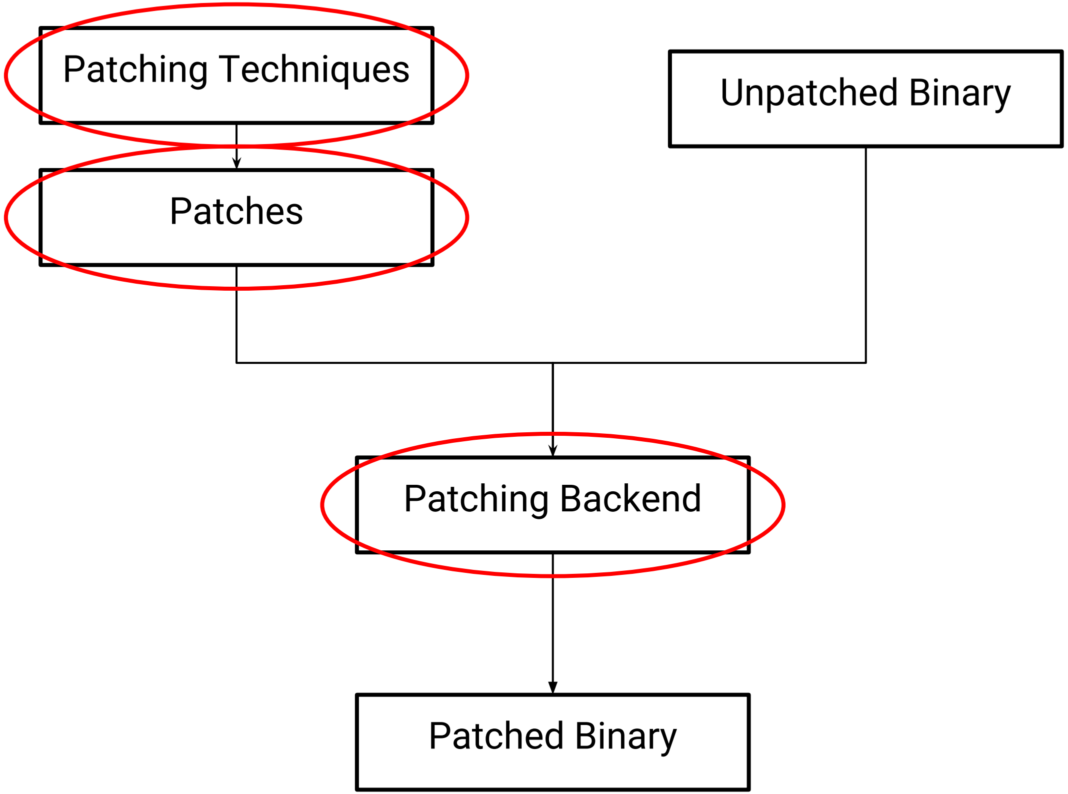
- Stack randomization
- Return pointer encryption
- ...

Patches:

- Insert code
- Insert data
- …

Patching Backend:

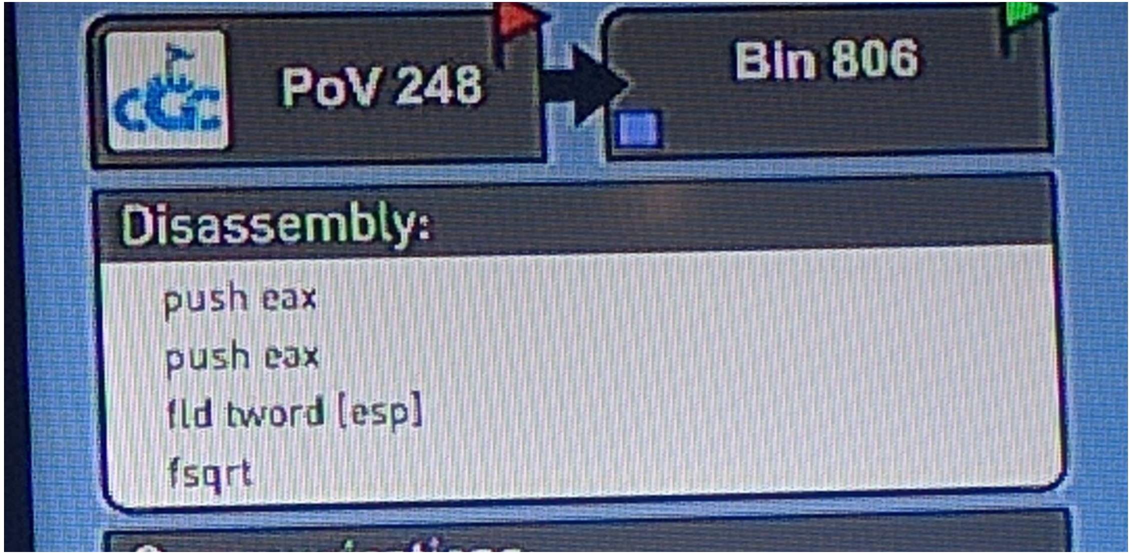
- Detour
- Reassembler
- Reassembler Optimized



Adversarial Patches 1/2

Detect QEMU

```
xor eax, eax
inc eax
push eax
push eax
push eax
fld TBYTE PTR [esp]
fsqrt
```



```
push eax
push eax
fld tword [esp]
fsqrt
```

Adversarial Patches 2/2

Transmit the flag

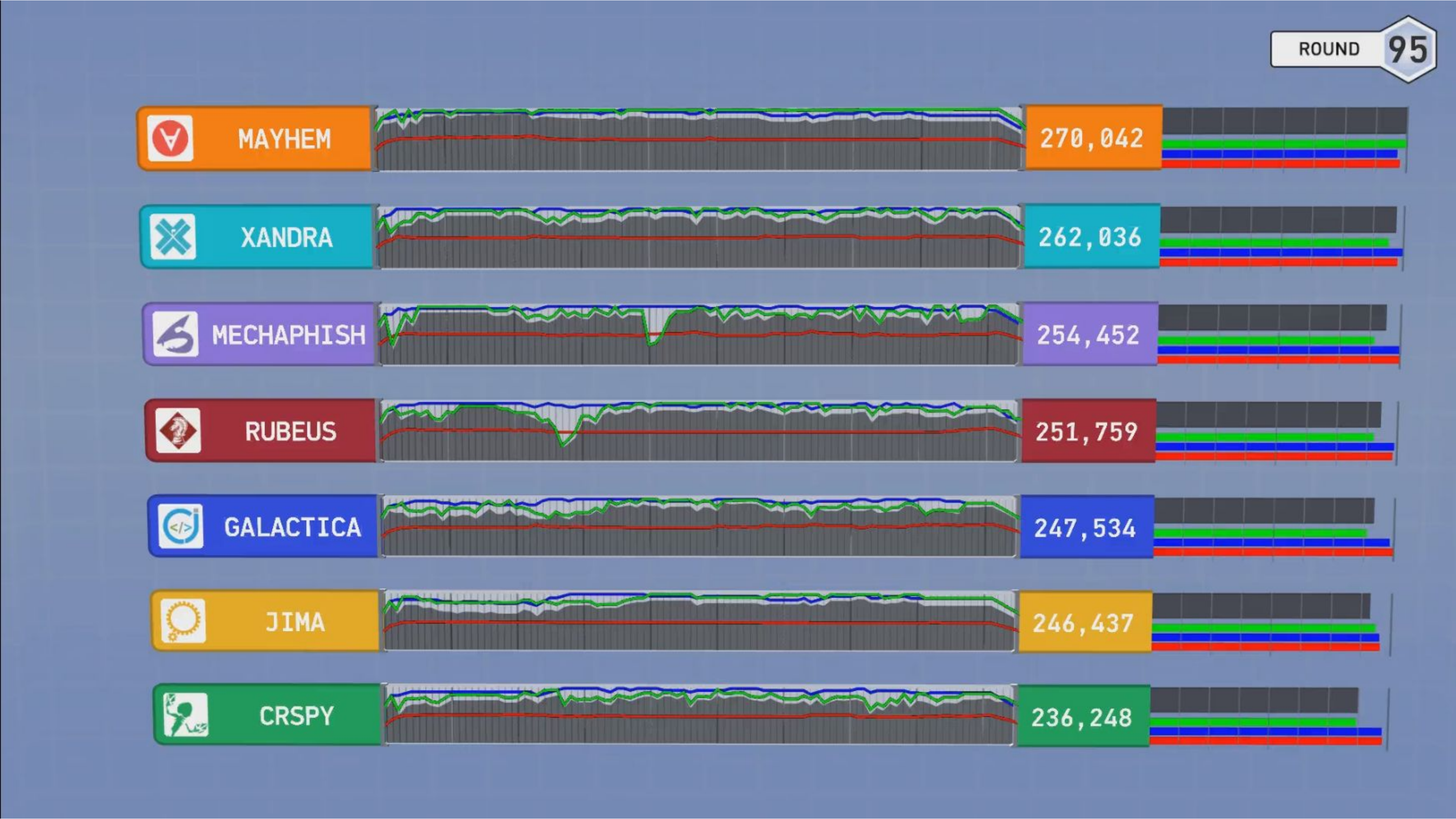
- To **stderr**!

Backdoor

- Hash-based challenge-response backdoor
- Not cryptographically secure (can be pre-computed)
- Good enough to defeat automatic systems (online > exec timeout)

Generic Patches

- Return pointer encryption
- Protect indirect calls/jmps
- Extended Malloc allocations
- Randomly shift the stack (ASLR)
- Clean uninitialized stack space
- ...



ROUND 95

MAYHEM	270,042
XANDRA	262,036
MECHAPHISH	254,452
RUBEUS	251,759
GALACTICA	247,534
JIMA	246,437
CRSPY	236,248

CFE Strategies / Techniques

Defense:

- Do not evaluate patches locally, too unreliable
- Do not deploy IDS rules, too dangerous
- Only briefly analyze patches from other teams
- Deploy patches immediately

Offense:

- Pwn as much as possible

CFE Statistics 1/3

- 82 Challenge Sets fielded
- 2442 Exploits generated
- 1709 Exploits for 14/82 CS with 100% Reliability
- Longest exploit: 3791 lines of C code
- Shortest exploit: 226 lines of C code
- crackaddr: 517 lines of C code

CFE Statistics 2/3

100% reliable exploits generated for:

- CROMU_000{46,51,55,65,94,98}
- KPRCA_00{065,094,112}
- NRFIN_000{52,59,63}
- YAN01_000{15,16}

Rematch Challenges:

- SQLSlammer (CROMU_00094)
- crackaddr (CROMU_00098)

CFE Statistics 3/3

Vulnerabilities in CS we exploited:

- CWE-20 Improper Input Validation
- CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer
- CWE-121: Stack-based Buffer Overflow
- CWE-122: Heap-based Buffer Overflow
- CWE-126: Buffer Over-read
- CWE-131: Incorrect Calculation of Buffer Size
- CWE-190: Integer Overflow or Wraparound
- CWE-193 Off-by-one Error
- CWE-201: Information Exposure Through Sent Data
- CWE-202: Exposure of Sensitive Data Through Data Queries
- CWE-681: Incorrect Conversion between Numeric Types
- CWE-787: Out-of-bounds Write
- CWE-788: Access of Memory Location After End of Buffer

CFE Pwning Statistics

<table>
<tr><th>Team</th><th colspan="2">Flags Captured (49 rounds | all)</th><th colspan="2">CSes Pwned (49 rounds | all)</th></tr>
<tr><td>Shellphish</td><td><b>206</b></td><td><b>402</b></td><td>6</td><td><b>15</b></td></tr>
<tr><td>CodeJitsu</td><td>59</td><td>392</td><td>3</td><td>9</td></tr>
<tr><td>DeepRed</td><td>154</td><td>265</td><td>3</td><td>6</td></tr>
<tr><td>TECHx</td><td>66</td><td>214</td><td>2</td><td>4</td></tr>
<tr><td>Disekt</td><td>101</td><td>210</td><td>5</td><td>6</td></tr>
<tr><td>ForAllSecure</td><td>185</td><td>187</td><td><b>10</b></td><td>11</td></tr>
<tr><td>CSDS</td><td>20</td><td>22</td><td>1</td><td>2</td></tr>
</table>

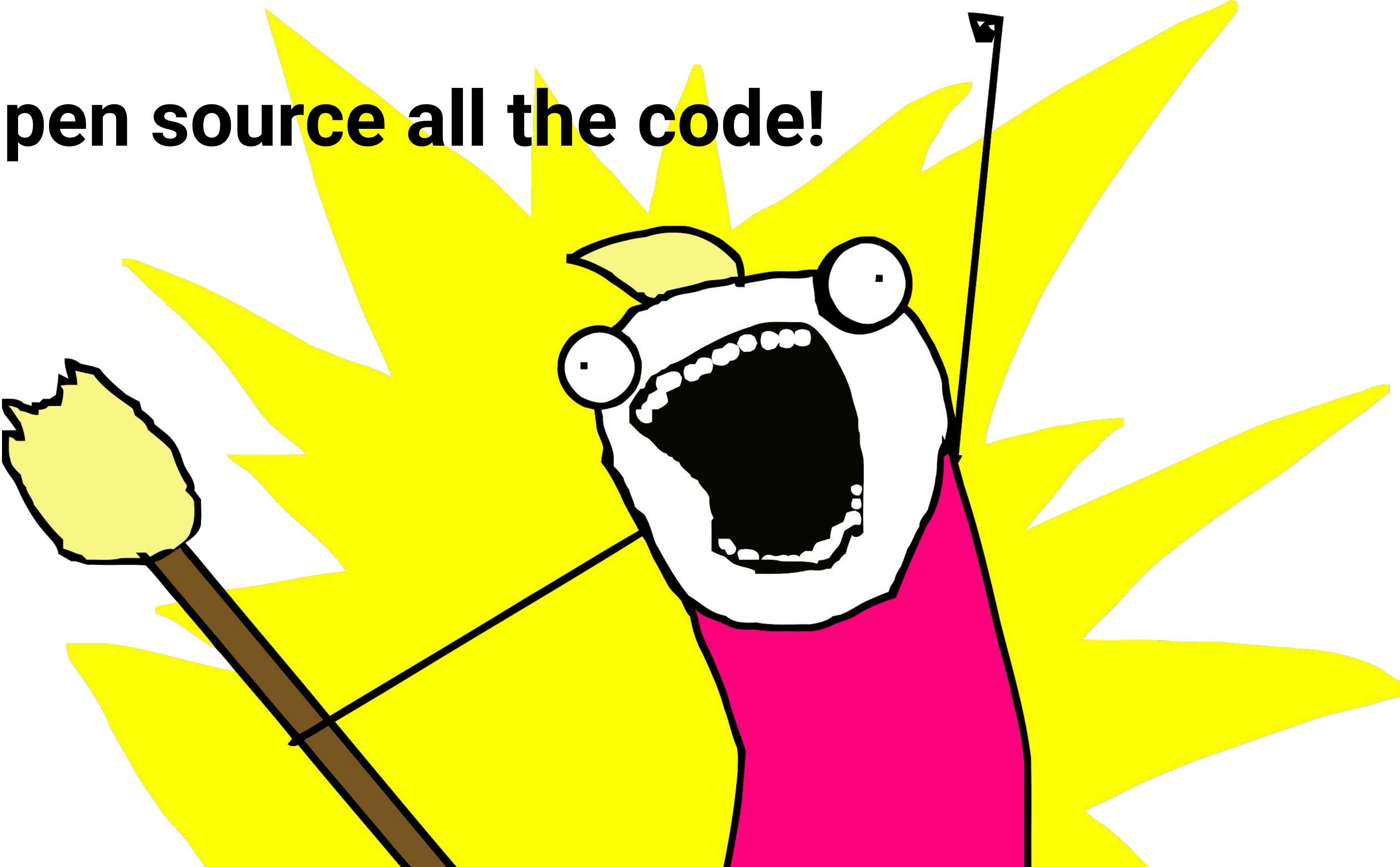
CFE Patching Statistics

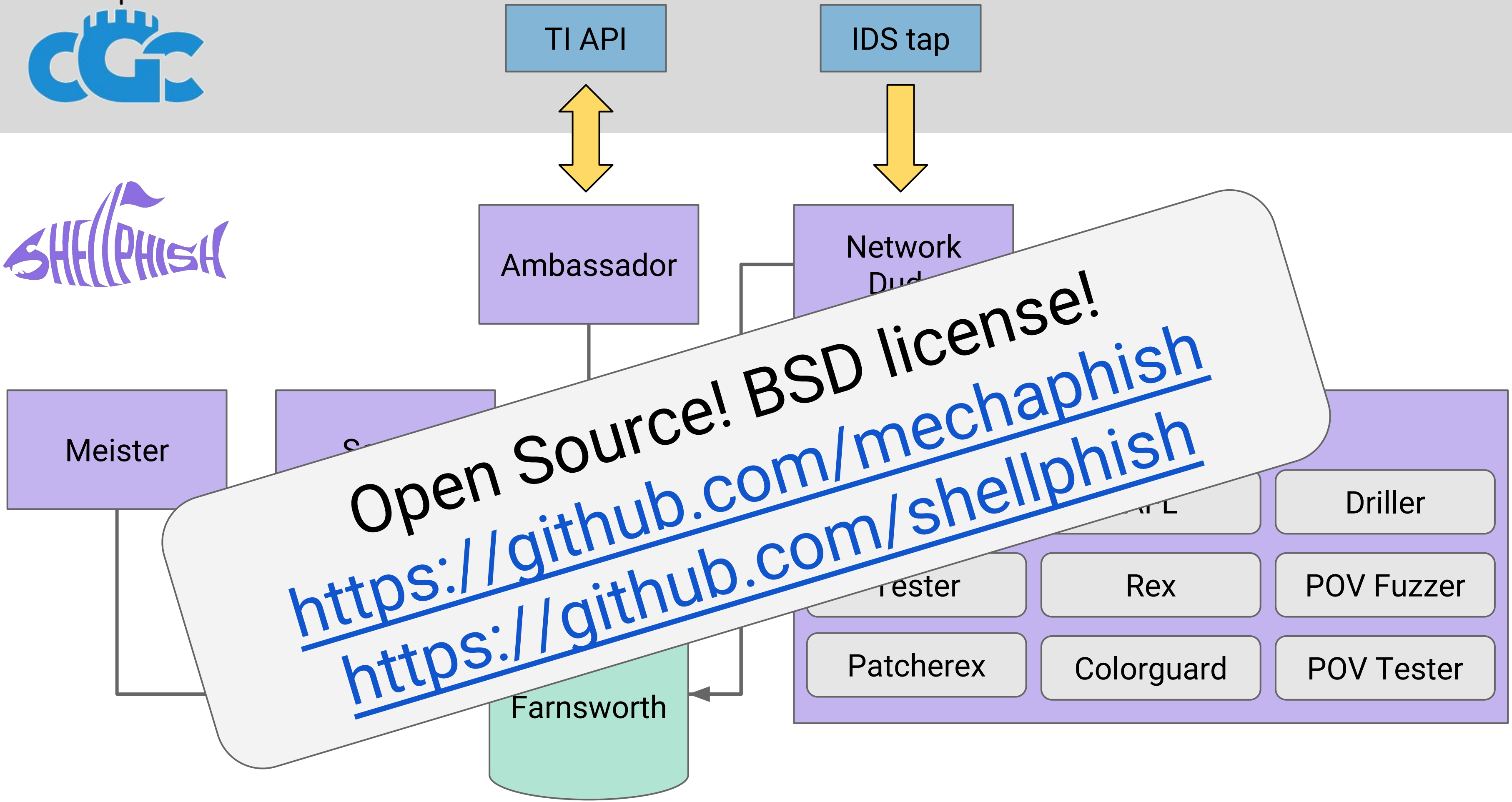
Team	**Defended CS-Rounds (49 rounds | all)**		**CSes Compromised (49 rounds | all)**	
Shellphish	29	68	7	**12**
TechX	27	61	7	14
DeepRed	32	87	**6**	15
ForAllSecure	54	160	7	16
CodeJitsu	61	104	9	16
Disket	66	127	9	17
CSDS	**108**	**189**	9	18

CFE St*p!d Bugs

- Network traffic synchronization
- Race condition in submission logic
- Slow scheduling by Kubernetes

Open source all the code!





Open Source! BSD license!

https://github.com/mechaphish

https://github.com/shellphish

On the Shoulders of Giants

QEMU
open source processor emulator

Z3

python

AFL

kubernetes

Unicorn
Engine

ubuntu

angr

pypy

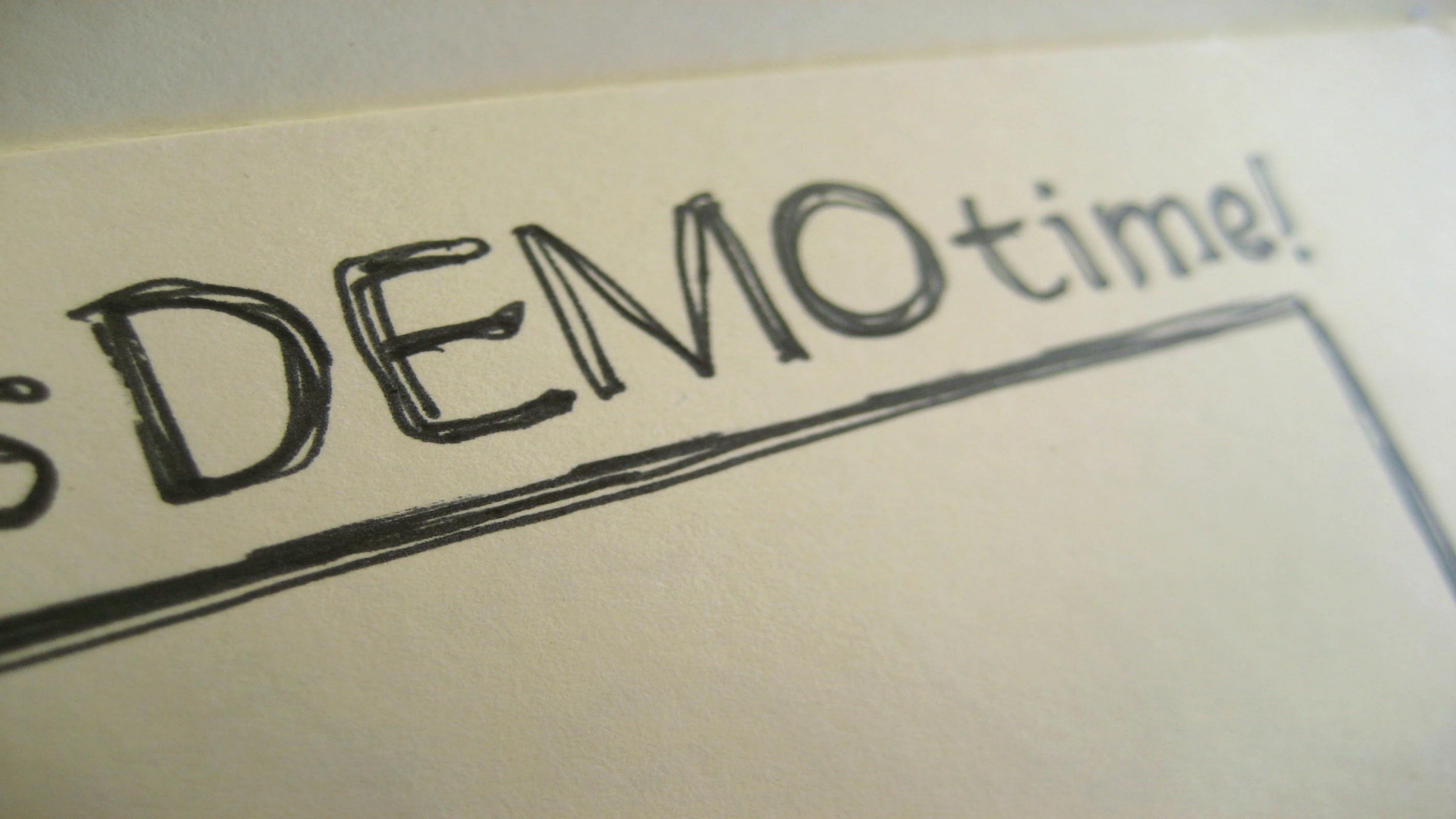
peewee

PostgreSQL

VEX

Capstone
Engine

docker



s DEMO time!

IVAN REIS
&
JOE PRADO
ADR:

Human Augmentation

DEFCON CTF 2016:

- CRS assisted with 5 exploits
- Human exploration → CRS exploitation
  - Semantic understanding of interactions/protocols helps
- Backdoors!

Thank you! Stay in touch!

Student? Looking for an internship? Master thesis? Wanting to do a PhD?

twitter: @shellphish

email: team@shellphish.net or cgc@shellphish.net

irc: #shellphish on freenode

twitter team:

@anton00b - **@caovc** - @giovanni_vigna - @jac_arc - @ltFish_ - @machiry_msdic - @nebirhos - @rhelmot - @zardus