DARPA's Cyber Grand Challenge: What Happened and What's Next

Tim Vidas

2017-10-18
NEbraskaCERT, CSF

Tim Vidas @tvidas

- UNO graduate (BS,MS in CS)
  - CMU graduate (PhD in ECE)
- DEF CON go-er for > 10 years
  - CTF player, CTF organizer, review board
- Cyber Grand Challenge devteam lead
  - The team that designed CGC and made the competition work
- Of note:

DDTEK

Shmoo

IEEE

Sk3wl 0f r00t

ACM

PPP

Shmoo

DEF CON Black Badges

DC3 Forensic Challenge Champion

CTF?

What is CTF in this context?

- A cyber security based Capture-the-Flag contest (aka exercise, event, game)

- Typically these contests involve demonstrating proficiency or excellence in one or more areas of computer and network security

- There are different models for architecting these contests, which can stress different skills, lend to particular objectives

- Increasingly popular, common

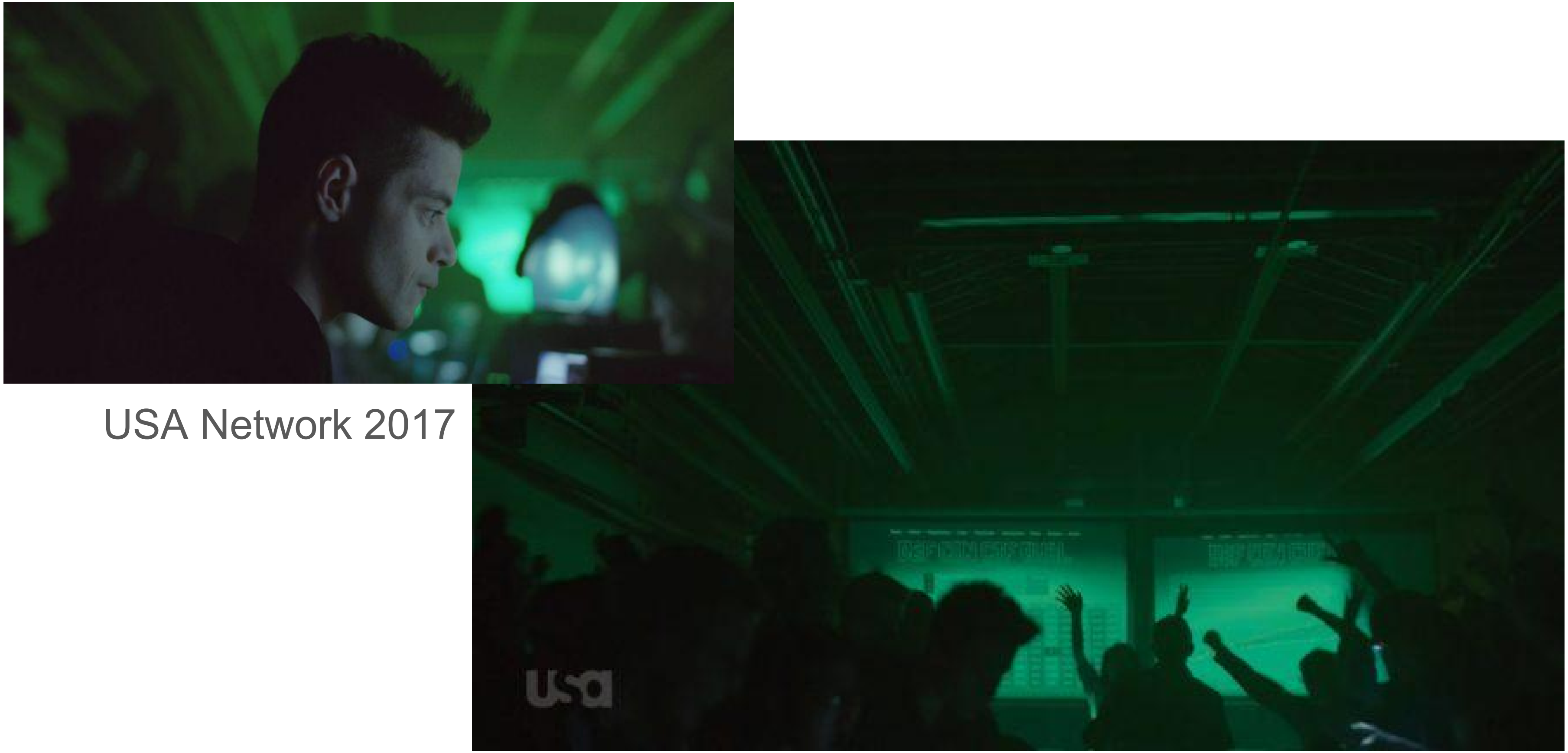
It is not:

- A game kids play with physical flags on hills
- A first-person shooter video game CTF (usually)
- Focused in the field of Social Engineering
- A hackathon

Though there are certainly similarities to these other games.

Today, the characters “CTF” are appended to many contests, in most cases this simply means “contest,” sometimes there are flags involved

CTF: Hollywood style (well, USA Network)



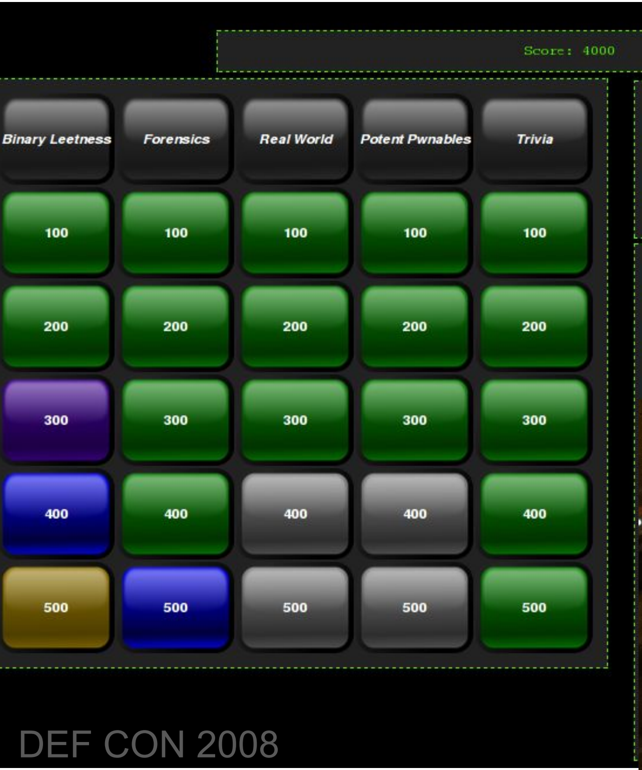
USA Network 2017

CTF: real life

DEF CON 2016

DEF CON 2002

CTF: real life



```
Logout TheUnd

Score: 4000
```

Binary Leetness	Forensics	Real World	Potent Pwnables	Trivia
100	100	100	100	100
200	200	200	200	200
300	300	300	300	300
400	400	400	400	400
500	500	500	500	500

```
What libc function is this?
  • file

i hate you    I owned it

Leaders
  1. Routards (5200)
  2. Pandas with Gambas (5200)
  3. Guard@MyLan0 (4800)
```

DEF CON 2008

DEF CON 2012

DEF CON 2011?

CGC?

Could a purpose-built super computer play in DEF CON’s Capture-the-flag (CTF)?

Autonomous…

- Binary analysis
- Binary patching
- Vulnerability discovery
- Service Resiliency (availability)
- Network Defense (IDS)

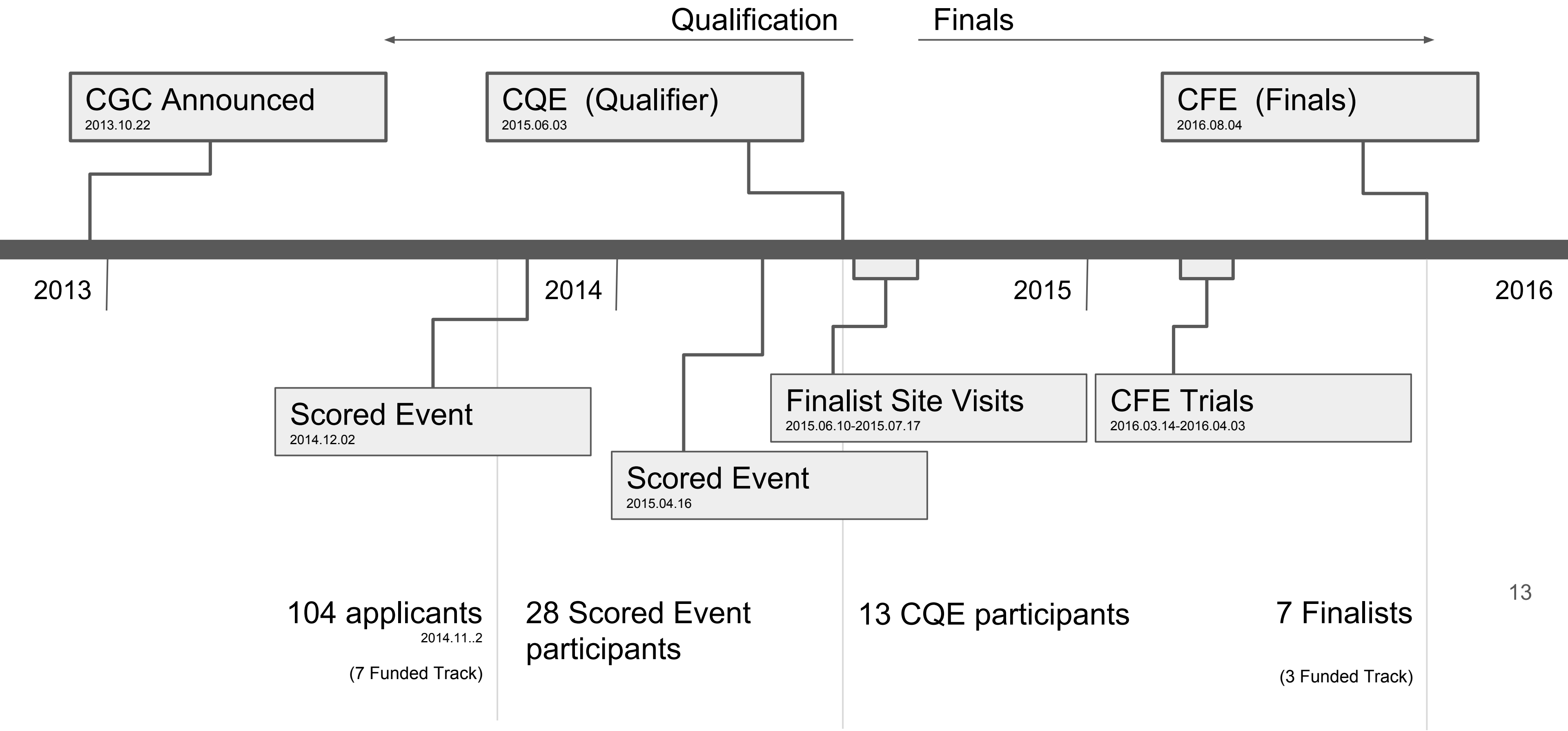
CGC: Real life

Image: DARPA

CGC: Real life

Image: DARPA

Competition Overview



Qualification

Finals

CGC Announced
2013.10.22

CQE (Qualifier)
2015.06.03

CFE (Finals)
2016.08.04

2013

2014

2015

2016

Scored Event
2014.12.02

Scored Event
2015.04.16

Finalist Site Visits
2015.06.10-2015.07.17

CFE Trials
2016.03.14-2016.04.03

104 applicants
2014.11..2

(7 Funded Track)

28 Scored Event participants

13 CQE participants

7 Finalists

(3 Funded Track)

CGC Qualification Event (CQE)

590 Explicit Flaws
131 Challenge Sets
24 hours
28 Participants
>=**5** CRSes on Twitter
$**750K** to prize to each unfunded qualifier

CRS Requirements:

- Demonstrate rudimentary capability
- Crashing inputs
- Mitigations
- Consensus evaluation

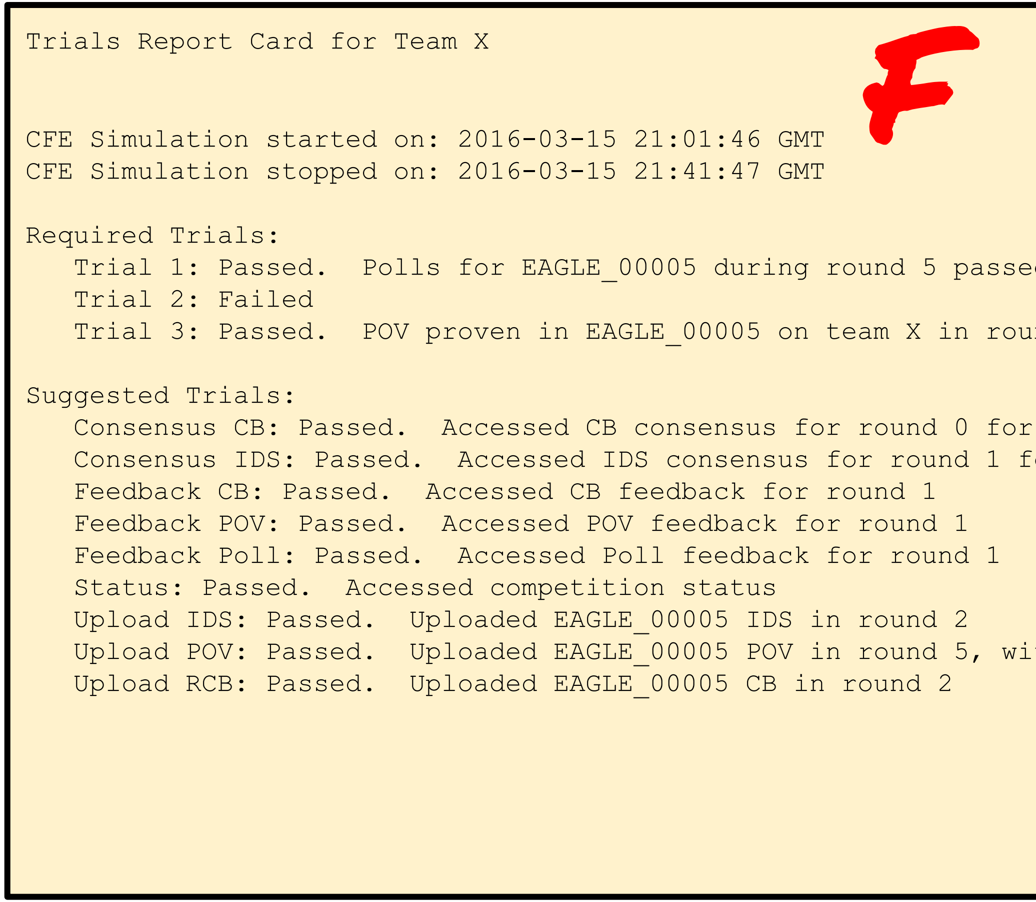
CFE Sparring/Trials

Conducted from 2016-02 to 2016-08

Opponents simulated by “sparring partner” software

CRS Requirements:

- Interact with API
- Upload POV (POV must succeed)
- Upload patched binary (patched binary must prevent POV)
- Upload IDS rule (IDS rule must be valid)



```
Trials Report Card for Team X


CFE Simulation started on: 2016-03-15 21:01:46 GMT
CFE Simulation stopped on: 2016-03-15 21:41:47 GMT

Required Trials:
   Trial 1: Passed.  Polls for EAGLE_00005 during round 5 passe
   Trial 2: Failed
   Trial 3: Passed.  POV proven in EAGLE_00005 on team X in rou

Suggested Trials:
   Consensus CB: Passed.  Accessed CB consensus for round 0 for
   Consensus IDS: Passed.  Accessed IDS consensus for round 1 f
   Feedback CB: Passed.  Accessed CB feedback for round 1
   Feedback POV: Passed.  Accessed POV feedback for round 1
   Feedback Poll: Passed.  Accessed Poll feedback for round 1
   Status: Passed.  Accessed competition status
   Upload IDS: Passed.  Uploaded EAGLE_00005 IDS in round 2
   Upload POV: Passed.  Uploaded EAGLE_00005 POV in round 5, wi
   Upload RCB: Passed.  Uploaded EAGLE_00005 CB in round 2
```

96 Rounds
9h 13m 17s duration
82 Challenge Sets
410 unique RCBs fielded
1299 unique PoVs fielded
(total of **270772** throws)
7 Functioning CRSes
1 Failed water pump
$**3.75M** USD prizes awarded

CGC Final Event (CFE)

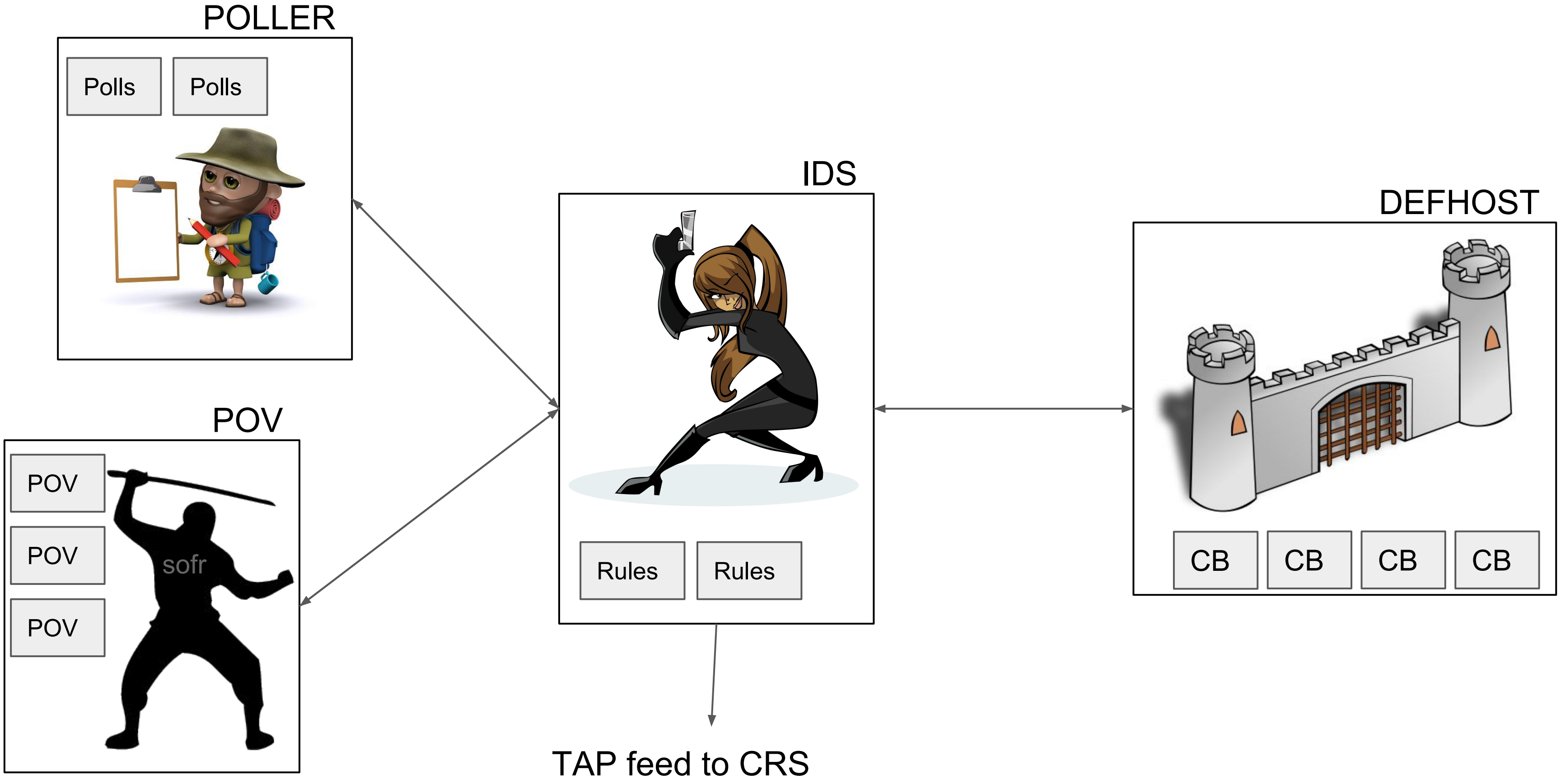
- Live event held at DEF CON in Aug 2016
- More expected of competitors than in CQE
  - IDS filters available
  - Full access to competitors mitigated binaries and IDS filter
  - Live network traffic feed available as tap on IDS
  - Stronger requirements for proof of vulnerability
- Infrastructure only evaluates performance and functionality
- Otherwise, infrastructure deploys mitigated binaries and launches POVs on behalf of competitors (a brokered competition)

CFE Game Flow

6 physical machines dedicated to “infrastructure side” for each competing CRS
Each CRS connected to the infrastructure via **2** ethernet cables

- Competitors interact with a “Team Interface” (**TI**)
  - Web server providing status updates and upload capability
- Defended host (**DEFHOST**)
  - Runs all Challenge Binaries or their CRS-supplied replacements (reformulated CB; RCB)
- Network Appliance (**IDS**)
  - Runs competitor supplied filter rules
  - Filters installed on a per-challenge set basis
  - ALL connections to Challenge Binaries run through IDS
- Poller (**POLLER**)
  - Runs DARPA generated functionality test interactions against active challenges
- POV (**POV**)
  - Runs CRS-provided POVs against active challenges

CFE Game Flow



POLLER

Polls

Polls

POV

POV

POV

POV

sofr

IDS

Rules

Rules

DEFHOST

CB

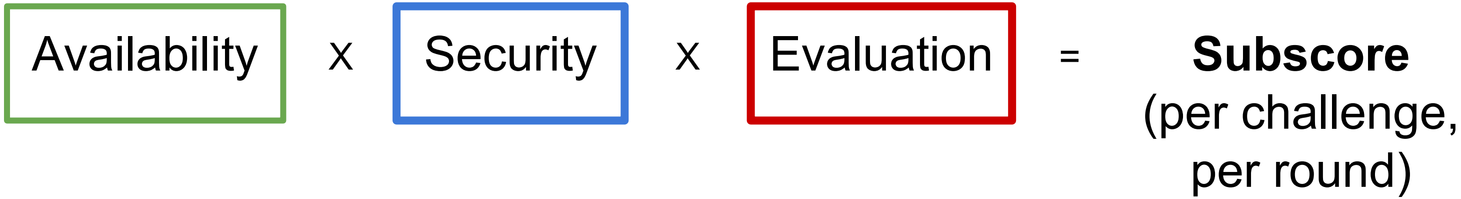
CB

CB

CB

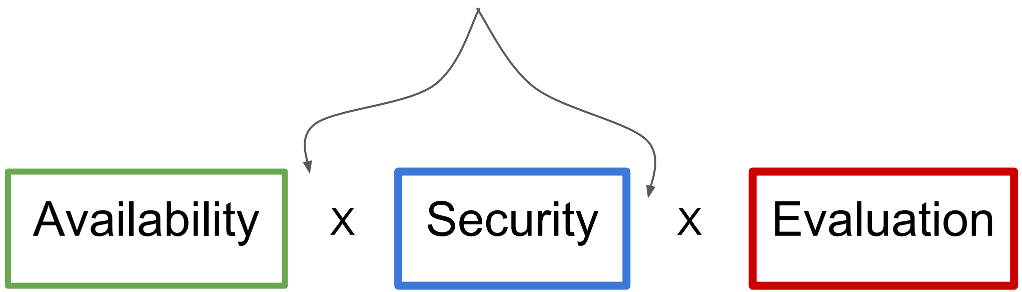
TAP feed to CRS

Scoring



Availability x Security x Evaluation = **Subscore** (per challenge, per round)

Scoring



Product, so a factor can drive the score to 0

Availability x Security x Evaluation

“SLA” & perf.

0 - 1.0
(e.g. “100%”)

Based on how many successful “polls” - simulated use of a service

&

Performance as measured in memory and CPU overhead relative to reference binary

“Defense”

1 or 2

How many competitors actually scored against your services / how well you protected your flags

“Offense”

1 - 2.0

Proportional based on how many teams you successfully scored against

Evaluating a POV

2 types of POVs in CFE
During CFE, **118708** Type-1 and **152064** Type-2 were negotiated by CRSes (**7512** and **5975** successful, respectively)
Vulnerabilities were proven in **20** (of 82) Challenge Sets in CFE
All **7** CRS successfully proved at least one vulnerability

Two POV types specified for CFE

- Type 1
  - Competitor POV claims it **can control EIP and one other register**
  - Negotiation transaction dictates specific values to POV
  - POV interacts with challenge set to cause a crash in the dictated state
  - **Crash state** (if any) examined to confirm success or failure of POV
- Type 2
  - Competitor POV claims it **can read from an arbitrary memory location**
  - Negotiation transaction dictates a region of memory from which POV must obtain 4 bytes
  - POV interacts with challenge set to leak said 4 bytes and submits them to complete the negotiation
  - **Submitted value** is examined to confirm success or failure of POV

Building the Competition

- Design concerns from the outset
    - Repeatability
        - Anyone should be able to verify CFE results
    - Competition integrity
        - Concerns with running competitor-provided code (POV/RCB)
        - Concerns with parsing competitor-provided data (IDS filters)
    - Data collection
        - Desire to publish corpus to serve as a reference for program analysis going forward

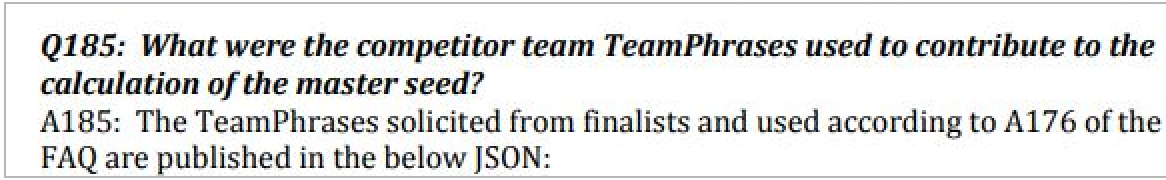
5 instructions trapped by hypervisor modified the behavior of **11** additional instructions

Repeatability

- Design goal was for every transaction to be as deterministic as possible
  - Modulo TCP
- Eliminated all sources of randomness that might be accessible to CGC binaries and made available the “random” system call
  - CGC hypervisor trapped all instructions that might be used to gather entropy
    - rdpmc, rdrand, rdtsc, rdtscp, rdseed
  - Some other instructions emulated or forbidden
    - cpuid, lgdt, lidt, sgdt, sidt, lldt, ltr, sldt, str, in, out
    - cpuid returned same values as developer’s MacBook Pro laptop
- Random pulled from a PRNG seeded by the CGC loader at process creation time
  - All seeds generated ahead of game time and recorded for later use

Competition Integrity

- Given the amount of prize money at stake, integrity of the competition was a grave concern and drove many design decisions
- Randomness was limited and/or made to be deterministically pseudorandom
- However, **nobody** should be able to predict aspects of CFE
    - The entire event was seeded with input from DARPA and all competitors (XORed) (Collected between June 10-17, 2016)
    - To ensure that DARPA did not select a particular input after knowing all competitor inputs DARPAs input was cryptographically committed to early (June 10,2016)
- Similarly, the CFE event plan (including challenge set schedule was committed to on Aug 2, 2016)
    - Organizers could not change the schedule in order to influence the event outcome



Q185: What were the competitor team TeamPhrases used to contribute to the calculation of the master seed?

A185: The TeamPhrases solicited from finalists and used according to A176 of the FAQ are published in the below JSON:

Weeks of my life were lost to this

https://github.com/CyberGrandChallenge/Event-FAQ/blob/master/event_faq.md

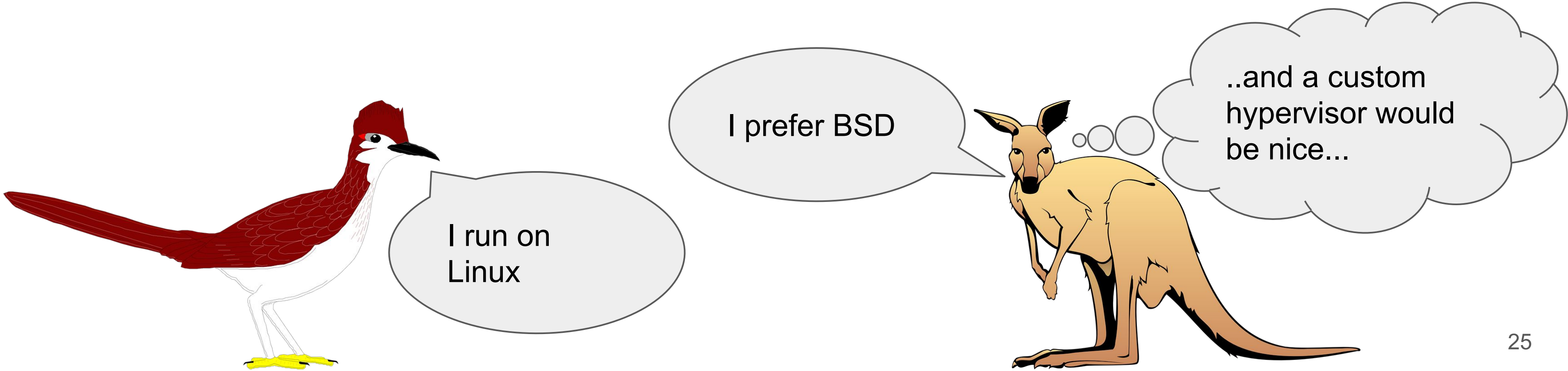
http://archive.darpa.mil/cybergrandchallenge_competitorsite/Files/CGC_FAQ.pdf

Competition Integrity

7 system calls

_terminate, transmit, receive, fdwait, allocate, deallocate, random

- Committed to kernels versions released prior to announcement of CGC
- Designed DECREE syscall environment / file format to reduce attack surface
- All game infrastructure components released to the public had private internal implementations
  - Notably, CFE ran on 64-bit FreeBSD 10 with a custom hypervisor module



Competition Integrity

- Air Gap
    - ○

Image: Vidas

Competition Integrity

- Air Gap
  - Power, cooling

Image: Vidas

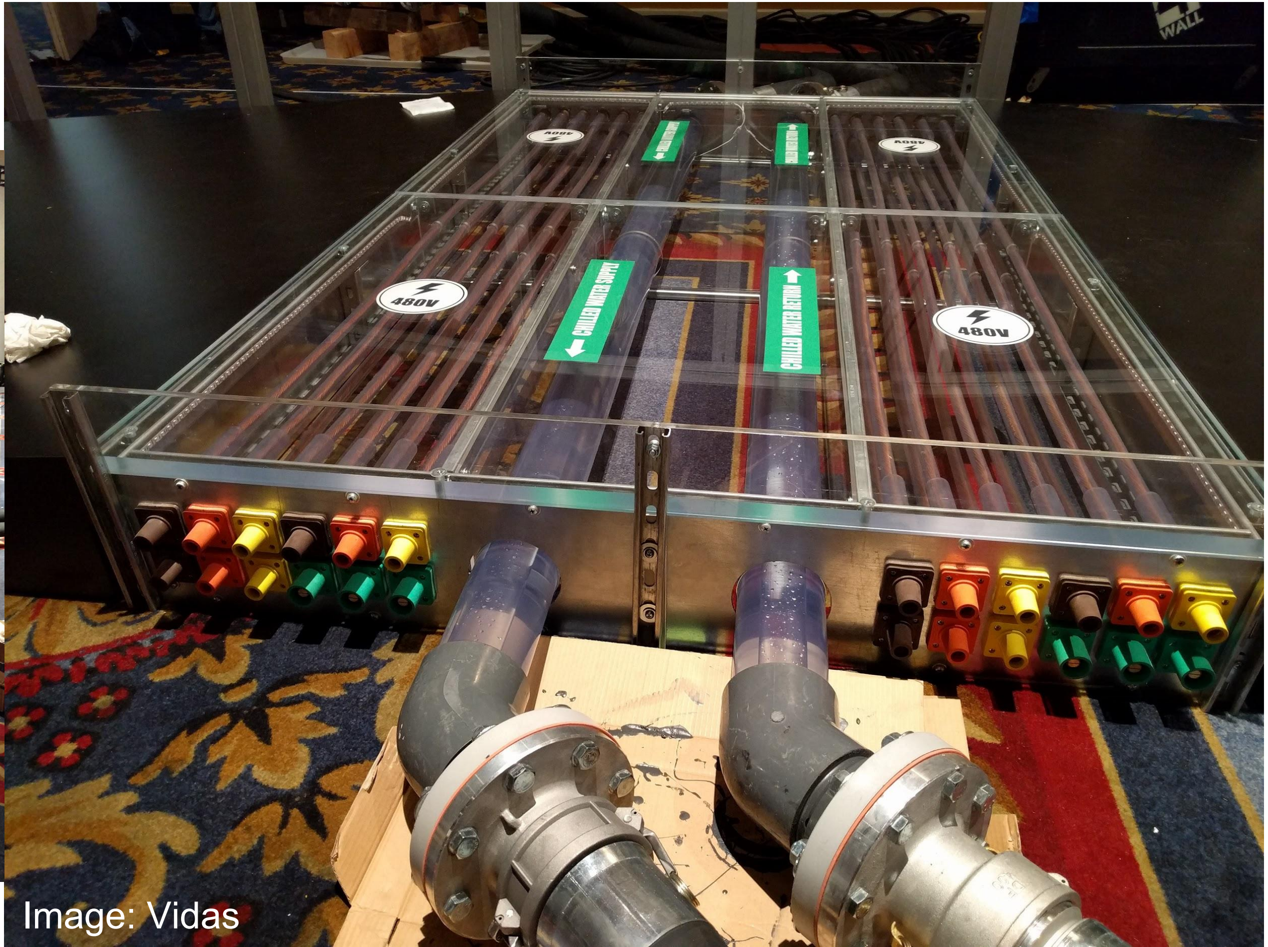


Image: Vidas

Competition Integrity

- Air Gap
  - One-way data

Image: Vidas

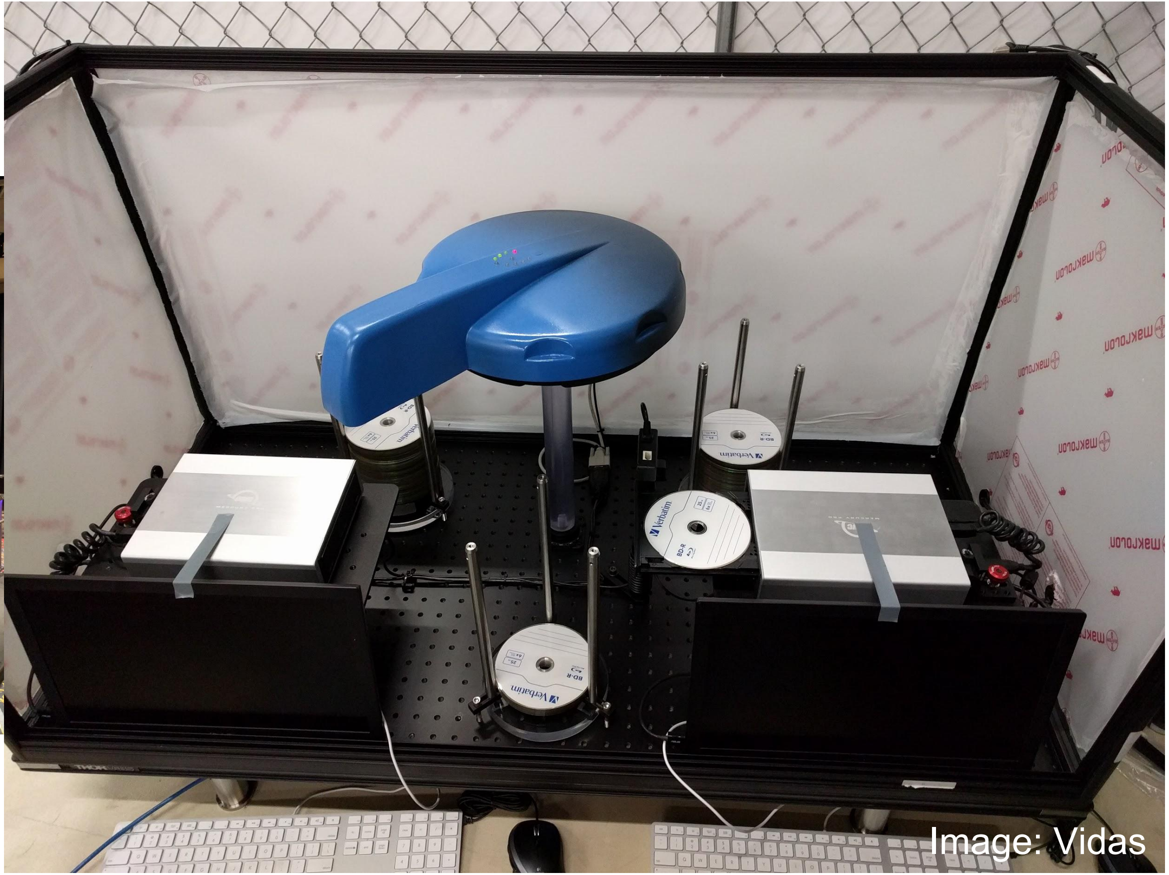


Image: Vidas

Competition Integrity

- Competitors were required to be autonomous, organizers weren’t
- Referees
- However, air gap

- Redundant HW
- Power/cooling
- Monitoring

Image: DARPA

Competition Integrity: Forensics

- Real-time forensics harness to vet software
  - Monitor OS for execution & data integrity
  - Built upon a full system emulator (Simics)
  - High fidelity x86 model from Intel
- Evaluated non-trusted code (POV/RCB) for attempts to breakout of DECREE environment
- Analyst replay tool
  - Replay any CFE session via IDA Pro gdb client
  - Reverse execution & scoring event detection

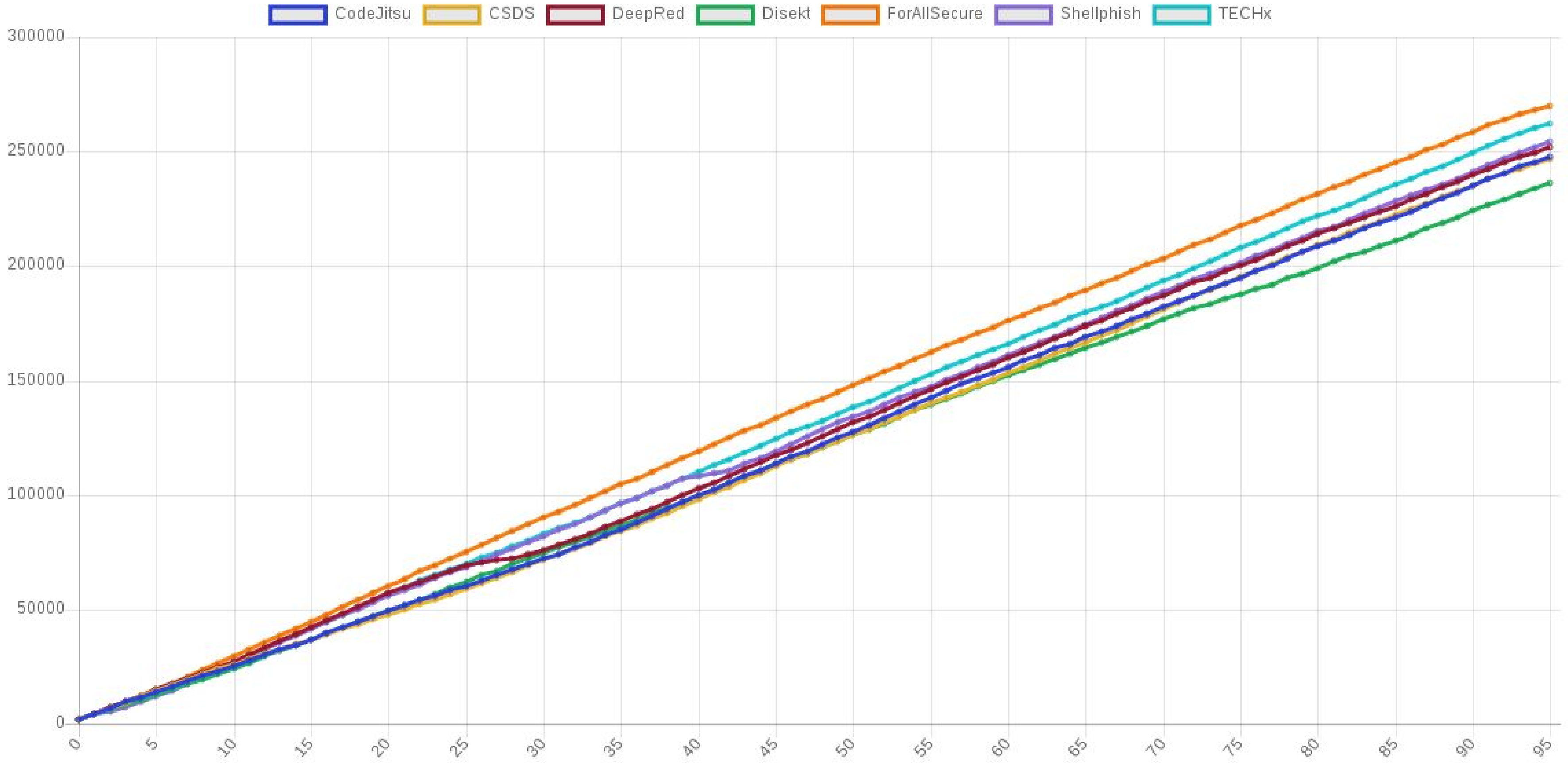
Data Collection

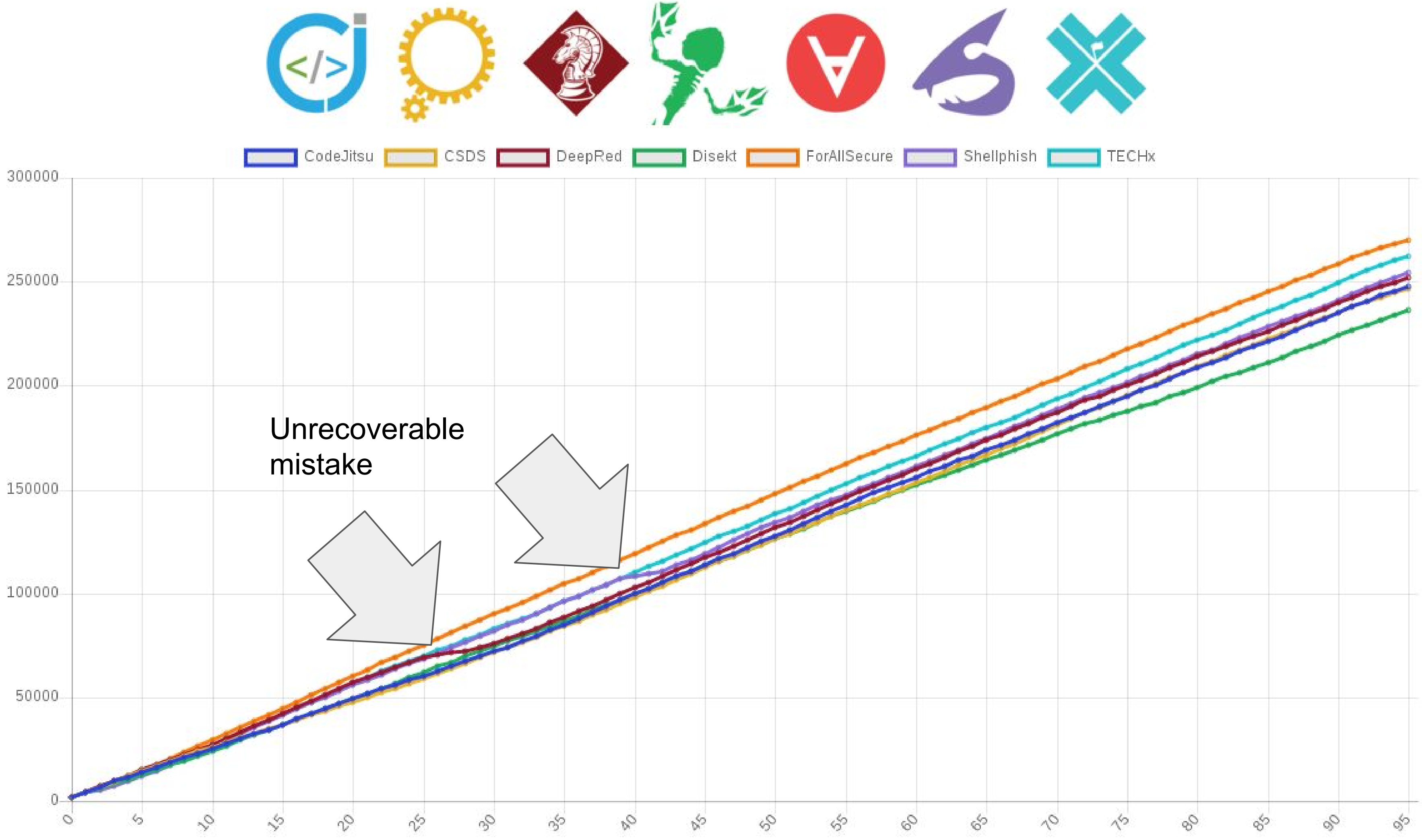
- From the outset we wanted to be able to be able to contribute a corpus of vulnerable challenge binaries of known provenance following CFE
    - Perhaps to serve as a reference for future program analysis research
- Additionally we wanted the game to be replayable and verifiable by any interested parties after the event.

http://www.lungetech.com/cgc-corpus/

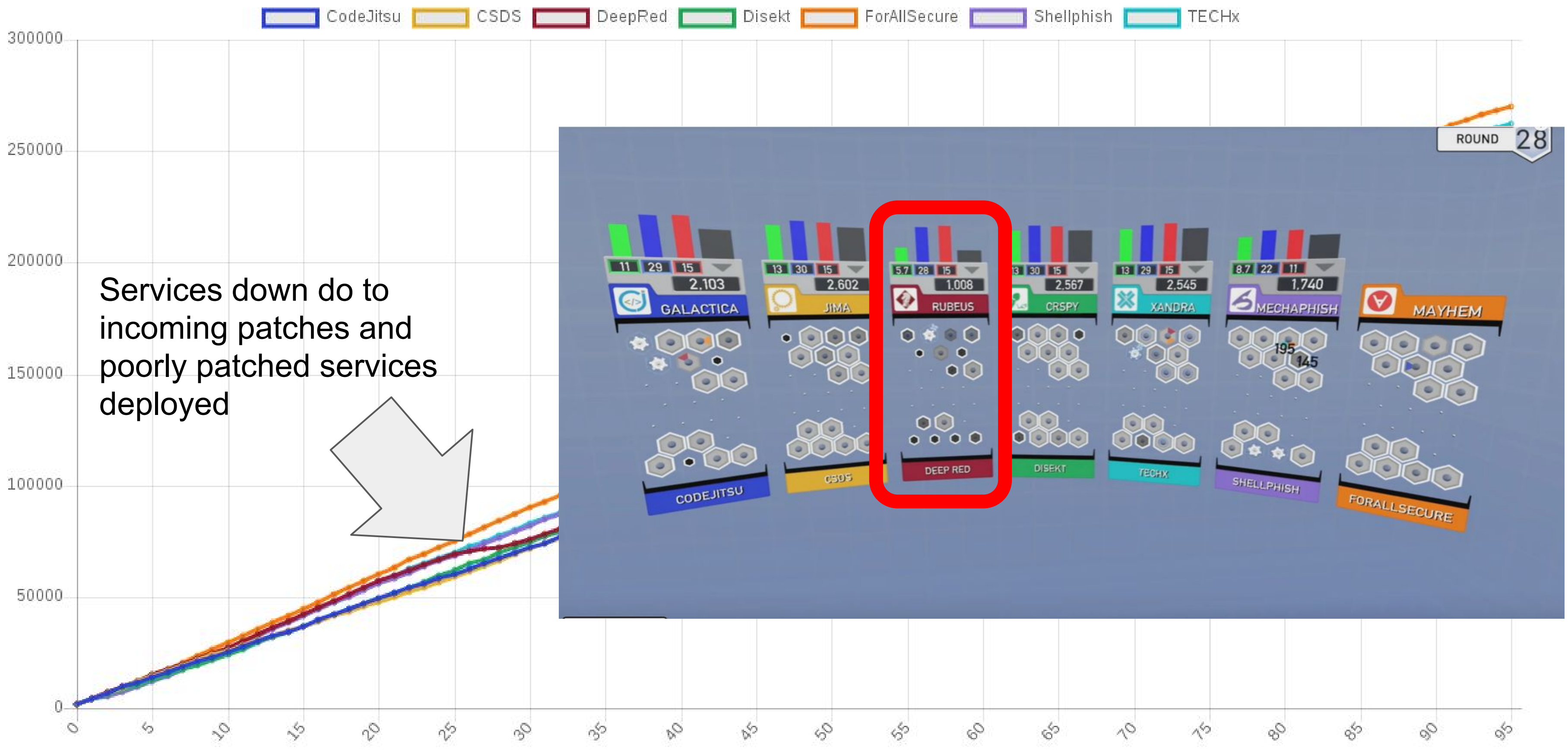
CGC:

- Proved that a CRS could be built
    - A computer could play CTF, by itself
- Provided specification for an autonomous and/or brokered CTF (CFE)
    - Which was used (kind of) for at least one other CTF: DEF CON 24 CTF
- Provided a corpus of software (w/ identified bugs, proofs, polls, etc)
    - http://repo.cybergrandchallenge.com/cfe/
    - http://www.lungetech.com/cgc-corpus/
    - https://github.com/lungetech/cgc-challenge-corpus
- Defined state-of-art data points for each CRS “component”
    - Less concrete, but broadly true and accepted
- Created interesting visualizations for binary analysis and CTF play

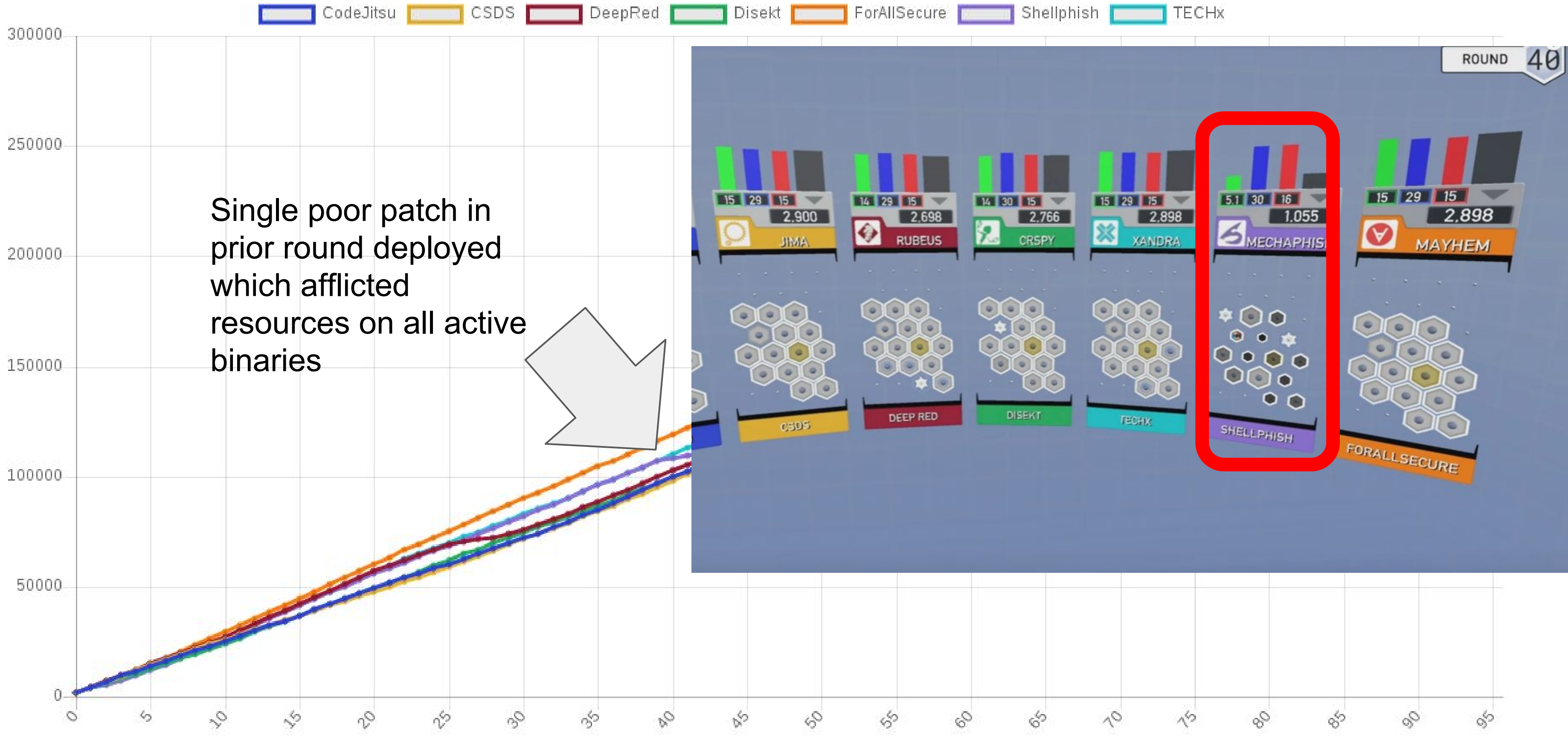

CodeJitsu
CSDS
DeepRed
Disekt
ForAllSecure
Shellphish
TECHx
300000
250000
200000
150000
100000
50000
0
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95



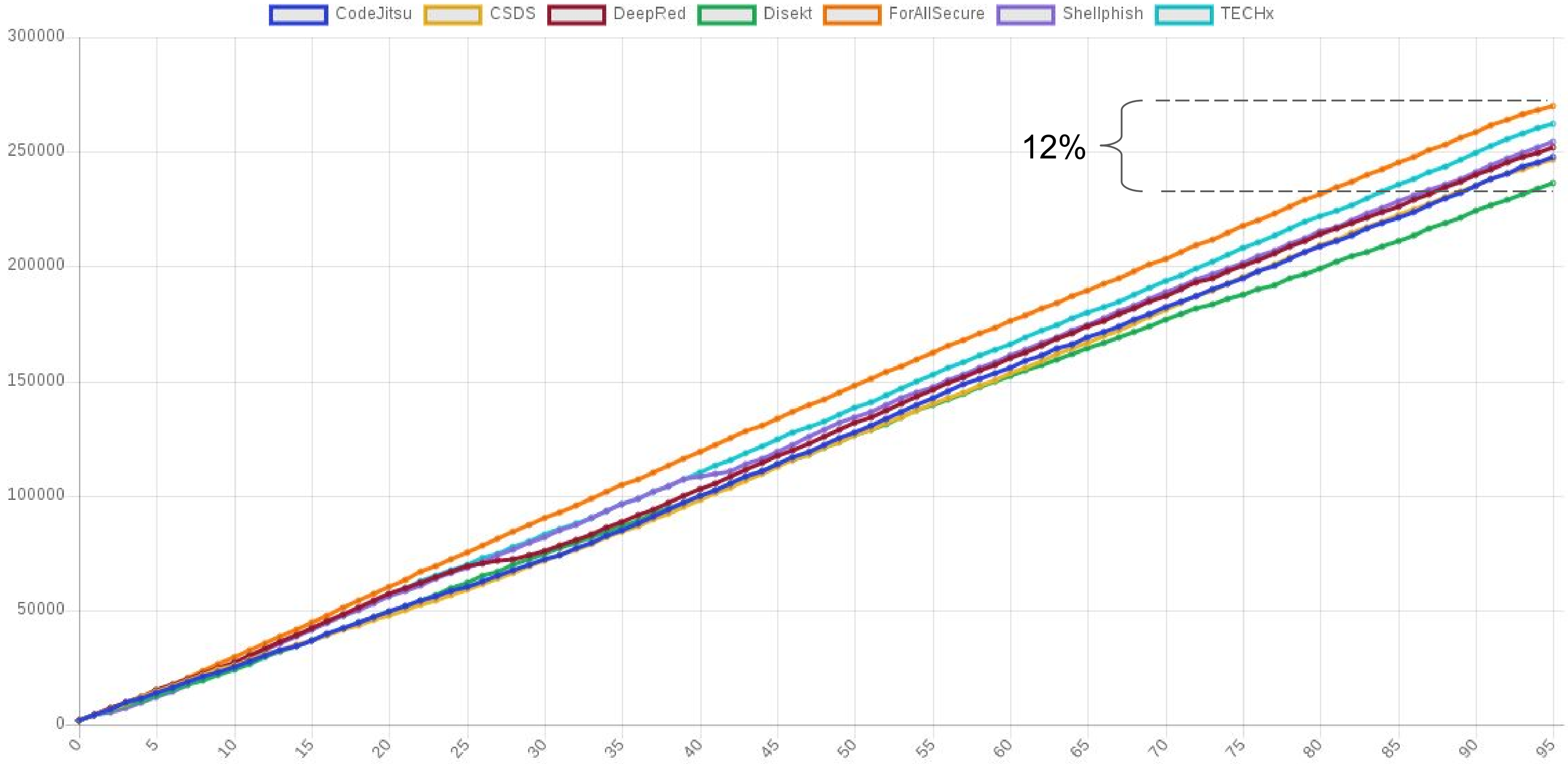
Unrecoverable mistake



Services down do to incoming patches and poorly patched services deployed



Single poor patch in prior round deployed which afflicted resources on all active binaries



CodeJitsu
CSDS
DeepRed
Disekt
ForAllSecure
Shellphish
TECHx
300000
250000
200000
150000
100000
50000
0
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
12%

Smithsonian exhibit

Image: Vidas

Human-computer hybrid

- Mayhem (the winning CRS) played “by itself” in DEF CON CTF
  - Not entirely true due to API incompatibilities
- Shellphish (3rd place CGC team) also qualified for DEF CON CTF
  - And were permitted by DARPA to use their CRS
  - The feedback loop reportedly had interesting effects like “finishing human work”

- There are interesting directions to take in this arena:
  - Machines assisting expert users (make one trained person perform like 100)
  - Machines assisting novice users (crowdsource useful information from 1000 strangers)
    - Test cases
    - Gamification

CGC:

- Did **NOT** demonstrate that AI has taken over the world, that computers are sentient, etc
- Did **NOT** reportedly employ any particularly complex “reasoning”
    - Recall that CRS internals are not necessarily known
- Did **NOT** find / exploit / break / etc existing or deployable real-world software
    - CGC used custom binary format and syscall interface
    - All challenge binaries were novel software (w/ mostly novel protocols, libc, etc)
    - Some bugs in real software were found during CGC development process, and reported
- Vulnerabilities were proven in **only 20 out of 82** software challenges

Not a 1, or 5, or 20 person undertaking

Further reading

CGC Website https://www.cybergrandchallenge.com/

CGC Release Repo http://repo.cybergrandchallenge.com/

CGC GitHub Repo https://github.com/CyberGrandChallenge

DARPA page http://www.darpa.mil/program/cyber-grand-challenge

Browsable data corpus http://www.lungetech.com/cgc-corpus/

highlight reel https://www.youtube.com/watch?v=v5ghK6yUJv4

smithsonian exhibit http://invention.si.edu/ai-and-challenge-cybersecurity

Rules https://cgc.darpa.mil/CGC_Rules_18_Nov_14_Version_3.pdf

Master Schedule https://cgc.darpa.mil/CGC_Master_Schedule_15_Apr_15.pdf

CQE news http://www.darpa.mil/news-events/2015-07-08

CRS Twitter feeds https://twitter.com/tvidas/lists/cgc-crses/

CGC Competitor Portal https://cgc.darpa.mil/

Shellphish competitor related info: http://shellphish.net/cgc/
ForAllSecure competitor related info: https://forallsecure.com/blog/tag/cgc/

CFE commentary

- CFE officially started at 16:00:45 UTC
- 40 rounds had completed by 19:41:09 UTC
- Power failure outside of airgap resulted in momentary failure in receiving data to feed visualization (Round 43 utilized our contingency data export protocol)
- CFE ended at max rounds (96) at 01:13:17 UTC
- Not counting original CBs, there were 512 unique RCBs uploaded, 410 of which were fielded
- Of 3570 unique POVs uploaded, 1299 were fielded, totalling 284823 throw opportunities, 270772 completed negotiations, and 13487 successful proofs

Some POV Related Numbers

Team	Type 1	Type2
CodeJitsu	2438	1202
CSDS	3	145
DeepRed	235	630
Disekt	89	1936
ForAllSecure	218	583
Shellphish	2398	1479
TECHx	2131	0

CSET	Type 1	Type 2
CROMU_00046	220	
CROMU_00051	83	70
CROMU_00055	68	2068
CROMU_00058		5
CROMU_00064		187
CROMU_00065	786	
CROMU_00073	95	7
CROMU_00088		6
CROMU_00094	779	400
CROMU_00095	25	

CSET	Type 1	Type 2
CROMU_00096		127
CROMU_00097		80
CROMU_00098	72	
KPRCA_00065	542	443
KPRCA_00094	148	
NRFIN_00052	1405	10
NRFIN_00059		620
NRFIN_00062	346	120
YAN01_00015	1652	730
YAN01_00016	1291	1102

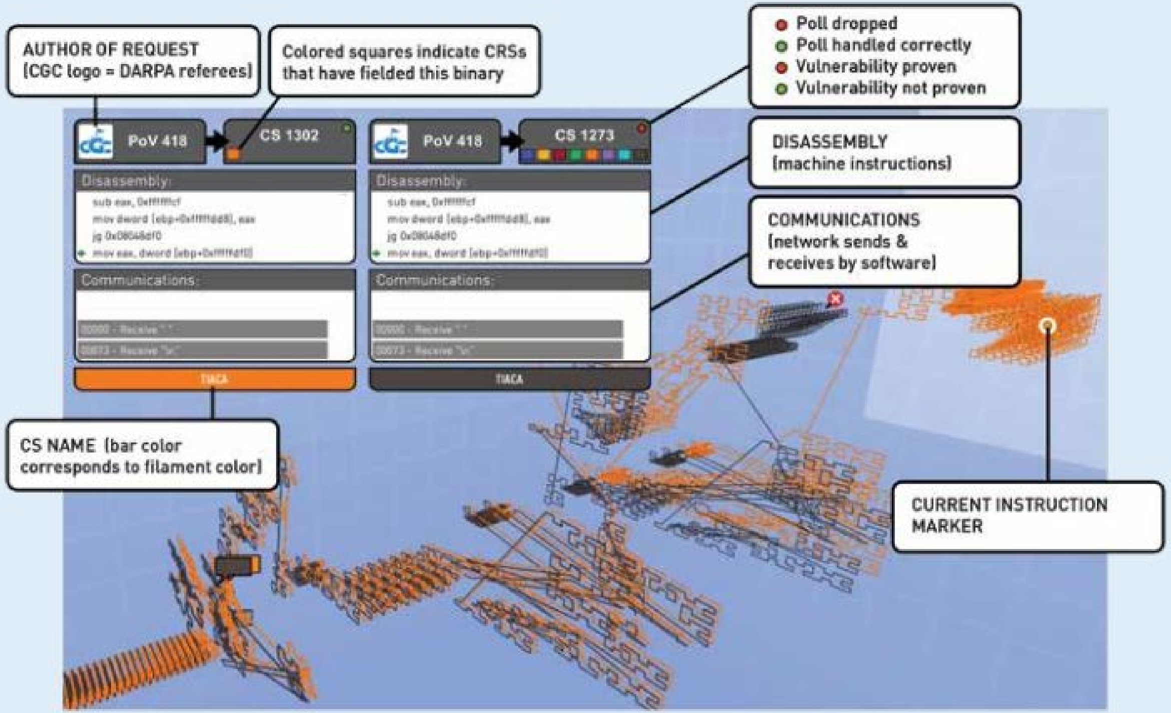
ON
OFF

NSCA™
NATIONAL STRENGTH AND
CONDITIONING AS
NON-POTABLE WATER
Rental Systems
1002141
Carrier
CHANGER
PORT COMPANY

Visualization

FILAMENT VIEW

Filament view traces the execution of software over a given input over time, moving from left to right. For example, a trace of an email client processing an email. The program begins executing on the left and time flows to the right. Visual loops are code loops; long straight lines show a long jump.



```
sub eax, 0xfffffcf
mov dword [ebp+0xffffdd8], eax
jg 0x08048df0
mov eax, dword [ebp+0xfffffdf0]
```

```
00000 - Receive " "
00073 - Receive "\n"
```

AUTHOR OF REQUEST (CGC logo = DARPA referees)

Colored squares indicate CRSs that have fielded this binary

- 🟥 Poll dropped
- 🟩 Poll handled correctly
- 🟥 Vulnerability proven
- 🟩 Vulnerability not proven

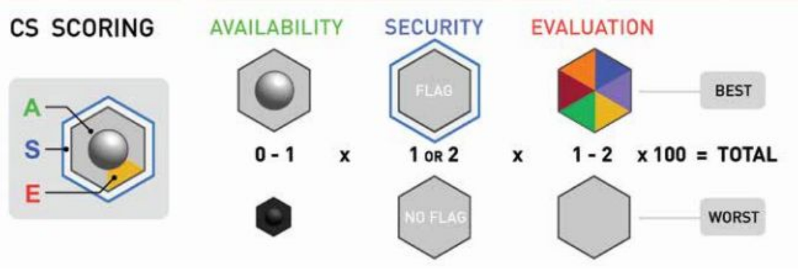
DISASSEMBLY (machine instructions)

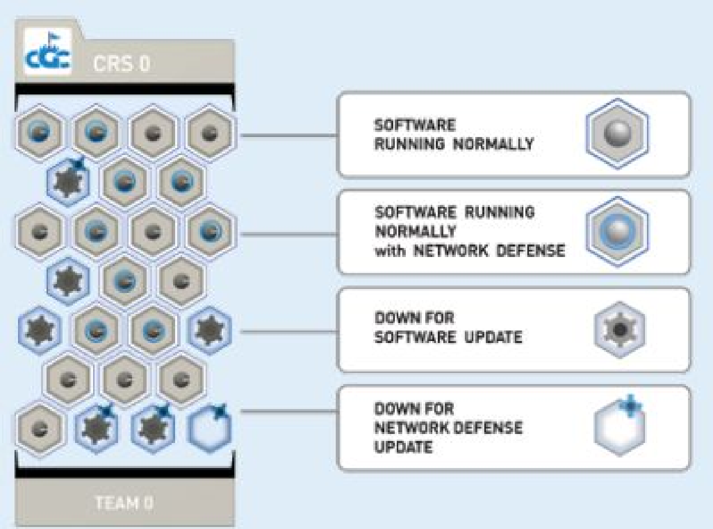
COMMUNICATIONS (network sends & receives by software)

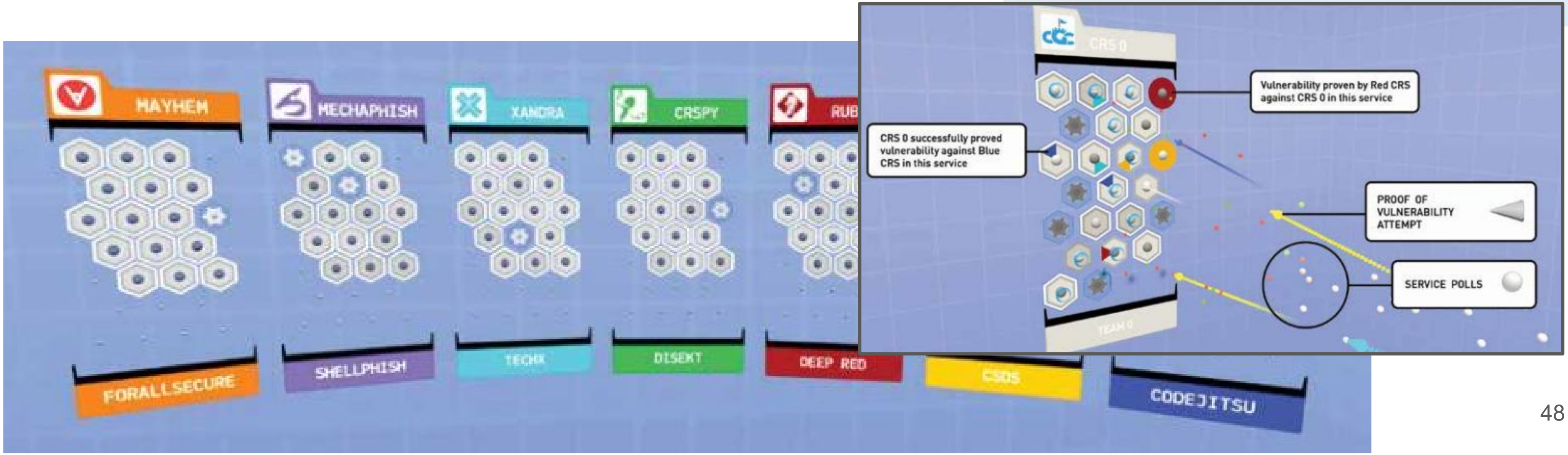
CS NAME (bar color corresponds to filament color)

CURRENT INSTRUCTION MARKER

Visualization




CRS 0
TEAM 0
SOFTWARE RUNNING NORMALLY
SOFTWARE RUNNING NORMALLY with NETWORK DEFENSE
DOWN FOR SOFTWARE UPDATE
DOWN FOR NETWORK DEFENSE UPDATE


MAYHEM
MECHAPHISH
XANDRA
CRSPY
RUB
FORALLSECURE
SHELLPHISH
TECHX
DISEKT
DEEP RED
CSDS
CODEJITSU
CRS 0
TEAM 0
Vulnerability proven by Red CRS against CRS 0 in this service
CRS 0 successfully proved vulnerability against Blue CRS in this service
PROOF OF VULNERABILITY ATTEMPT
SERVICE POLLS