

Running Zeek with AI: Is It Helping?

Aashish Sharma
Lawrence Berkeley National Lab

Berkeley Workshop, 2026



U.S. DEPARTMENT OF
ENERGY



**UNIVERSITY OF
CALIFORNIA**



Is AI Actually Helping?

A field report from AI-assisted Zeek operations at Berkeley Lab:

- a. Tool generation
- b. IR & Forensics
- c. Detection scripts
- d. Operational debugging

Aashish Sharma · Lawrence Berkeley National Laboratory



The prevailing wisdom

Every AI-in-security talk this year has a version of this slide. It is tidy, it is reasonable, and I agree with all of it.

Category	What it means	Who decides what	How it fails
Tool generation	The tooling around Zeek	You specify, AI writes and rewrites	Plausible, wrong for your hardware
Detection scripts	Zeek policy that makes notices and logs	You have the idea, AI drafts the Zeek	Parses fine, never fires
Operational debugging	Why is this sensor misbehaving	You point at symptoms, AI finds candidates	Confident wrong root cause
IR and Forensics	Hunting, indicators, timelines	AI sweeps, you own the conclusion	Fabricated evidence

It is also not evidence.



My take...

Category	What I actually built	What reached production
Tool generation	the UT Austin Dorkbot parser · A NIC/NUMA/IRQ config generator · a 20-feed intel aggregator	Dorkbot is in use so is NIC config. The aggregator still runs, and **21% of its rows are dead.**
Detection scripts	A C2 beacon detector · a Bayesian anomaly detector · a social-engineering campaign detector · a cloud-asset intel package · a scan-detector modernisation · two blocking-pipeline monitors	Four deployed. Two live on one sensor, cluster path proven only by btest. Rest in “improvement”
Operational debugging	Four investigations in three weeks, each with a written method. Then 300 GB proxies, and one bloom filter misconfig.	Three reusable runbooks. One upstream fix merged.
Forensics and IR	A CVE hunt, published package · two report-to-intel pipelines · an affidavit extraction	One package published with anonymization and tests.

Category	What I actually built	What reached production
Tool generation	the UT Austin Dorkbot parser · A NIC/NUMA/IRQ config generator · a 20-feed intel aggregator	The generator is in use. The aggregator still runs, and **21% of its rows are dead.**
Forensics and IR	A CVE hunt, published package · two report-to-intel pipelines · an affidavit extraction	One package published with anonymization and tests.
Detection scripts	A C2 beacon detector · a Bayesian anomaly detector · a social-engineering campaign detector · a cloud-asset intel package · a scan-detector modernisation · two blocking-pipeline monitors	Four deployed. Two live on one sensor, cluster path proven only by btest. Rest in “improvement”
Operational debugging	Four investigations in three weeks, each with a written method. Then 300 GB proxies, and one bloom filter misconfig.	Three reusable runbooks. One upstream fix merged.

Everything on this slide is checkable; the companion paper cites each one

UT Austin's ISO scans the internet and emails us one message per vulnerable host. About a hundred came that day.

What do you do ?

1. Are these really flagging badness ?
2. How do I process
 - a. Manually
 - b. Spreadsheet - populate ?
 - c. Zeek ?

Parse them on the wire, not from an mbox.

```
21917 0 May 21 UT Information ( 13K) [ISOTicket:6131816] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21918 0 May 21 UT Information ( 13K) [ISOTicket:6131826] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21919 0 May 21 UT Information ( 13K) [ISOTicket:6131828] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21920 0 May 21 UT Information ( 13K) [ISOTicket:6131832] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21921 0 May 21 UT Information ( 13K) [ISOTicket:6131846] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21922 0 May 21 UT Information ( 13K) [ISOTicket:6131858] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21923 0 May 21 UT Information ( 13K) [ISOTicket:6131861] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21924 0 May 21 UT Information ( 13K) [ISOTicket:6131865] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21925 0 May 21 UT Information ( 13K) [ISOTicket:6131869] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21926 0 May 21 UT Information ( 13K) [ISOTicket:6131871] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21927 0 May 21 UT Information ( 13K) [ISOTicket:6131883] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21928 0 May 21 UT Information ( 13K) [ISOTicket:6131886] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21929 0 May 21 UT Information ( 13K) [ISOTicket:6131918] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21930 0 May 21 UT Information ( 13K) [ISOTicket:6131923] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21931 0 May 21 UT Information ( 13K) [ISOTicket:6131931] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21932 0 May 21 UT Information ( 13K) [ISOTicket:6131935] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21933 0 May 21 UT Information ( 13K) [ISOTicket:6131955] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21934 0 May 21 UT Information ( 13K) [ISOTicket:6131979] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21935 0 May 21 UT Information ( 13K) [ISOTicket:6131982] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21936 0 May 21 UT Information ( 13K) [ISOTicket:6131983] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21937 0 May 21 UT Information ( 13K) [ISOTicket:6131997] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21938 May 21 UT Information ( 13K) [ISOTicket:6132003] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21939 0 May 21 UT Information ( 13K) [ISOTicket:6132006] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21940 0 May 21 UT Information ( 13K) [ISOTicket:6132014] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21941 0 May 21 UT Information ( 13K) [ISOTicket:6132024] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21942 0 May 21 UT Information ( 13K) [ISOTicket:6132029] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21943 0 May 21 UT Information ( 13K) [ISOTicket:6132036] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21944 0 May 21 UT Information ( 13K) [ISOTicket:6132038] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21945 0 May 21 UT Information ( 13K) [ISOTicket:6132046] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21946 0 May 21 UT Information ( 13K) [ISOTicket:6132050] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21947 0 May 21 UT Information ( 13K) [ISOTicket:6132054] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21949 0 May 21 UT Information ( 13K) [ISOTicket:6132072] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21950 0 May 21 UT Information ( 13K) [ISOTicket:6132075] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21951 0 May 21 UT Information ( 13K) [ISOTicket:6132080] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21952 0 May 21 UT Information ( 13K) [ISOTicket:6132094] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21953 0 May 21 UT Information ( 13K) [ISOTicket:6132096] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21954 0 May 21 UT Information ( 13K) [ISOTicket:6132117] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21955 0 May 21 UT Information ( 13K) [ISOTicket:6132119] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21956 0 May 21 UT Information ( 13K) [ISOTicket:6132124] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21957 0 May 21 UT Information ( 13K) [ISOTicket:6132161] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21958 0 May 21 UT Information ( 13K) [ISOTicket:6132163] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21959 0 May 21 UT Information ( 13K) [ISOTicket:6132167] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21960 0 May 21 UT Information ( 13K) [ISOTicket:6132176] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21961 0 May 21 UT Information ( 13K) [ISOTicket:6132185] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21962 0 May 21 UT Information ( 14K) [ISOTicket:6132190] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
21963 0 May 21 UT Information ( 13K) [ISOTicket:6132192] UT/ISO: Publicly Exposed Vulnerability Detected [LBL.60V]
```



Three sessions, one Zeek log

Session	Clock	My prompts	It asked	Out
mbox to TSV/CSV	14 min	4	0	253-line parser; its CSV became the spec
Python to Zeek, live SMTP	30 min	9	3	585-line Zeek script + policy + a runbook
Production bug, mirror, rename	18 min	6	4	ticket-keyed dedup, JSON mirror

I wrote no requirements document, no tests, and no Zeek.



It reasoned from the RFC. The network disagreed.

```
# its design, 21 May:  
Key: c$smtp$msg_id when present  
# "list expansion keeps the original  
# Message-ID (RFC convention)"
```

```
# its own test rule, same file:  
confirm dedup is working  
  (unique tickets == row count)
```

A week later, on production traffic: **1823 rows for 104 tickets**. Its own rule failed. Re-keyed on the ticket number UT Austin already puts in every subject.

Seventeen copies of every notification cross the wire. One row is written. And the key it first chose does not exist in this traffic at all.



A script to write node.cfg

Capability	Flag	For
Dry run by default	(default)	--apply commits sysfs, ethtool and IRQ writes; needs root
Interface selection	-i "ifcs"	auto-discovers otherwise. Name your taps.
AF_PACKET fanout	--af-packet	required when workers exceed hardware queues
Hyperthread policy	--allow-ht	off by default; siblings share L1/L2
Management placement	--mgmt-node auto	picks the NUMA node with no capture NIC
Exclusive cores	--allow-core-overlap	default divides a node's cores between its NICs
Name preservation	--worker-name IFC=LABEL	the block name is the peer name in every log
Persistence	--install-systemd	offloads and IRQ pinning do not survive a reboot
Self-test	--self-test	unit tests against synthetic /sys fixtures; no root, no NICs
Measurement	--worker-stats	per-worker drops and lag from stats.log



Category	What I actually built	What reached production
Tool generation	the UT Austin Dorkbot parser · A NIC/NUMA/IRQ config generator · a 20-feed intel aggregator	The generator is in use. The aggregator still runs, and **21% of its rows are dead.**
Forensics and IR	A CVE hunt, published package · two report-to-intel pipelines · an affidavit extraction	One package published with anonymization and tests.
Detection scripts	A C2 beacon detector · a Bayesian anomaly detector · a social-engineering campaign detector · a cloud-asset intel package · a scan-detector modernisation · two blocking-pipeline monitors	Four deployed. Two live on one sensor, cluster path proven by btest. Rest in “improvement”
Operational debugging	Four investigations in three weeks, each with a written method. Then 300 GB proxies, and one bloom filter misconfig.	Three reusable runbooks. One upstream fix merged.

Everything on this slide is checkable; the companion paper cites each one

AI Tailored outreach, 120 messages, no payload

I came across your work at the Lab... From your perspective, when a utility scale project enters a community early, which factor tends to matter most: clarity around local benefits, the quality of engagement, or the predictability of the permitting process?

Persona B to a staff scientist, March 2026 · anonymised

No URL, no attachment, no request for anything. Correct grammar, real research, and a question only an expert would enjoy answering. **Every gateway control passed it.**

What arrived

Individually tailored messages ~120

Distinct recipients 45+

Fabricated personas, one voice 2

URLs, attachments, money asks 0

Messages to the top target 6+

Hierarchy-wide: director, PIs, facility and operations staff, **and retirees**. Interleaved with ordinary internal mail.



First pass, on the smtp.log alone

Phase 2	Initial contact	IN PROGRESS
Phase 3	Engagement	NOT YET
Phase 4	Trust building	NOT YET
Phase 5	The ask	NOT YET

The campaign has NOT progressed beyond Phase 2 based on this data. But absence of evidence is not evidence of absence — replies may exist in threads not captured here.

the same report, one line later

Then one thread was added

A TARGET REPLIED IN

31 h

warm, thoughtful, on a weekend

THE PERSONA ANSWERED IN

26 min

Saturday — monitored in real time

Phase 4, confirmed — offer to meet locally and handed over a personal address and a mobile number. Intent was that the **communication channel leave the institution.**

New patterns added for 178 real-world subjects:

Two Fabricated Personas

Attribute	Amy Bloom	Lillian Briger
Origin	"Grew up in Monaco"	"Grew up in Brunel"
Education	"Design in Paris, then finance in US"	"Business administration at Stanford"
Claimed work	Art collections, investment structures, scientific philanthropy	Advising royal families on next-gen education planning
Authority signals	Royal families, European/American family offices	Royal families, prominent American families
Closing	"Warm regards, Amy Bloom"	"Warm regards, Lillian Briger"
Email format	Mix of plaintext and HTML <div>	Mostly HTML <div> with gmail-isSelectedEnd classes

Both personas are **fictional constructs** operated by the same threat actor (identical sentence structures, rhetorical patterns, and strategic goals).

Common Attack Vectors Table

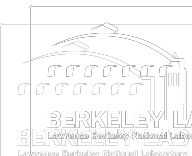
Vector	Technique	Evidence in Corpus
Hyper-personalization	Each email references the target's specific research papers, projects, and lab affiliations (BELLA Center, CUORE, Phytosome, ALS, Molecular Foundry, CASCADE, DESI, etc.)	Every scan email is uniquely tailored – zero generic language about research
Fabricated authority	Claims of working with "royal families," "family offices," "Stanford," "Monaco" – unverifiable prestige markers	Identical boilerplate across all emails
Intellectual flattery	"It is fascinating how...", "It is remarkable how..." – validates the target's life work	Present in 100% of scan emails
Low-commitment ask	"If you ever have a moment to share a quick thought" – trivial to respond to, opens the channel	Universal closing pattern
Binary question framing	"Will breakthroughs come from X, or from Y?" – designed to be irresistible for domain experts to answer	Every email ends with an either/or question about the target's field
No malicious payload	Zero links, zero attachments, zero money requests – passes all automated filters	Intentional: this is Phase 1 only
Dual-persona coverage	Same targets (██████████ etc.) receive emails from both Amy and Lillian	Multiple targets appear 2-4 times with near-identical content
Template reuse	Identical bio paragraphs, identical closing lines, swapped research details	Boilerplate bio is copy-pasted verbatim across dozens of emails
Broad targeting	Lab ██████████, senior scientists, engineers, administrative/operations staff, facility managers	~30+ distinct targets across the entire lab hierarchy
Legitimate email camouflage	Scan emails are mixed with real internal lab communications (██████████ newsletter) suggesting the corpus is from a real inbox	Scan emails designed to blend into a scientist's normal correspondence

Red Flags Summary

The most alarming finding is the sheer volume of repeated sends to the same targets. This reveals the attacker's priority list and persistence:

Target	Amy Bloom Sends	Lillian Briger Sends	Total Hits	Notes
Dr. [REDACTED] (Lab Director)	26	0	26	HIGHEST priority target – retiring lab director, leadership access
Dr. [REDACTED]	26	0	26	DNA repair / cancer biology – high-profile researcher
Dr. [REDACTED]	0	23	23	ALS infrared spectroscopy
Dr. [REDACTED]	0	24	24	Scientific computing, HEP data
Dr. [REDACTED]	23	0	23	HPC, Roofline model
Robert [REDACTED] (ate)	23	0	23	Earth system modeling
Dr. [REDACTED]	0	22	22	Cellular aging & cancer
Dr. [REDACTED]	22	0	22	Plant genomics, Phytosome
Prof. [REDACTED]	22	0	22	Surface chemistry, catalysis pioneer
Prof. [REDACTED]	22	0	22	Cryo-EM pioneer
Prof. [REDACTED]	22	0	22	Climate modeling
Dr. [REDACTED]	22	0	22	Environmental geochemistry
Dr. [REDACTED]	22	0	22	Indoor air quality
Dr. [REDACTED]	22	0	22	Building airflow modeling
Dr. [REDACTED]	22	0	22	Building energy systems
Dr. [REDACTED]	22	0	22	Neutrino physics (CUORE)
Dr. [REDACTED]	22	0	22	Exascale computing
Dr. [REDACTED]	22	0	22	Battery interfaces
Dr. [REDACTED]	22	0	22	X-ray imaging, ALS
Dr. [REDACTED]	22	0	22	HVAC fault detection
Dr. [REDACTED]	22	0	22	ISO 50001, energy management
Dr. [REDACTED]	22	0	22	Low-swirl combustion
Prof. [REDACTED]	22	0	22	Ultrafast molecular science
Mr. [REDACTED]	22	0	22	Molecular Foundry
Dr. [REDACTED]	22	0	22	Climate statistics
Dr. [REDACTED]	22	0	22	Superconducting magnets
Dr. [REDACTED]	22	0	22	BELLA Center, laser-plasma
Dr. [REDACTED]	22	22	24	Building energy / scientific computing
Dr. [REDACTED]	0	22	22	Distributed computing
Mr. [REDACTED]	0	22	22	Building ventilation
Dr. [REDACTED]	0	22	22	Energy efficiency policy
Mr. [REDACTED]	0	22	22	Utility-funded efficiency
Dr. [REDACTED]	0	22	22	DC power distribution
Mr. [REDACTED]	0	22	22	Sustainable transportation
Dr. [REDACTED]	0	22	22	DESI, dark energy
Dr. [REDACTED]	0	22	22	Building energy systems
Dr. [REDACTED]	22	0	22	Environmental microbiology
John [REDACTED] (er)	22	0	22	Research infrastructure
Dr. [REDACTED]	22	0	22	ALS accelerator upgrades
Mr. [REDACTED]	0	22	22	ALS Photon Science ops
Open [REDACTED] named	0	22	22	"Hello" only – no name used
Dr. [REDACTED] sen	1	0	1	Isotope geochemistry
Prof. [REDACTED] ng	0	22	22	Synthetic biology (different question style)
Dr. [REDACTED] lo	0	1	1	Cardiovascular genomics
Mr. [REDACTED]	0	1	1	Energy Technologies Area
Mr. [REDACTED]	0	1	1	Electricity markets
Mr. [REDACTED]	0	1	1	Electricity market design

Total unique targets: ~45+ individuals
Total scam emails in corpus: ~120+



Key Forensic Observations

1. Format Variation = Multiple Send Attempts

Many targets received the **same email in both plaintext and HTML** (<div dir="ltr">) format. This indicates:

- Different email clients or sending infrastructure per attempt
- Attempts through different email gateways (one may have been blocked)
- A/B testing which format gets past spam filters

2. Persona Division of Labor

Persona	Targeting Pattern
Amy Bloom	Senior scientists, lab leadership, technical researchers – higher profile
Lillian Briger	Operations staff, facility managers, policy researchers, mid-career – broader sweep

Amy handles the "prestige" targets. Lillian handles the "infrastructure access" targets. Together they map the entire org chart.

3. The [REDACTED] Email is Different

Professor [REDACTED]'s email from Lillian uses a **completely different question structure**: "What is the most common and most important misunderstanding people outside your field have about it?" This is a psychologically more personal question designed to elicit a longer, more reflective answer – suggesting [REDACTED] may be a **higher-value target** getting a more refined approach.

4. Some Emails Target Unnamed Recipients

Several emails say "Hello Dr.," with no name – these are likely **failed lookups** where the attacker had an email address but couldn't match it to a public profile. They still sent the email anyway, which reveals:

- Automated pipeline with a fallback template
- Willingness to sacrifice quality for coverage

5. Legitimate Emails Repeat Too

The [REDACTED] emails also repeat multiple times. This suggests the corpus is aggregated from **multiple inboxes** (likely a shared security/phishing report system), not a single person's mailbox.

Attack Phase Assessment

Phase 1: Reconnaissance	☒	COMPLETE	(targets identified, research scraped)
Phase 2: Initial Contact	☒	IN PROGRESS	(this corpus – mass first-touch)
Phase 3: Engagement	☐	NOT YET	(no replies detected)
Phase 4: Trust Building	☐	NOT YET	(would follow a reply)
Phase 5: The Ask	☐	NOT YET	(money, access, or information)

The campaign has **NOT progressed beyond Phase 2 based on this data**. But absence of evidence is not evidence of absence – replies may exist in threads not captured here.



What Happens Next – Prediction Tree

Based on common scam/social engineering patterns, here's what I'd expect:

She replies to [REDACTED]

- Scenario A: Romance/Confidence Scam (40%)
 - Warmer tone, personal questions ("do you live alone?", "what do you do for fun?")
 - Photos shared ("here's me at a conference")
 - Move to WhatsApp/Signal/Telegram ("easier to chat")
 - Eventually: financial ask (travel money, investment opportunity, crypto)
- Scenario B: Advance Fee / Investment Scam (25%)
 - "The families I advise are interested in clean energy infrastructure"
 - "Would you be open to an advisory role?"
 - Some kind of financial instrument, fund, or "opportunity" presented
 - Eventually: upfront fee, wire transfer, or credential harvesting
- Scenario C: She Proposes a Real Meeting (15%)
 - Names a specific café and date
 - Actually shows up
 - This is the hardest scenario to read – could be legitimate, could be a different kind of con that requires face-to-face
- Scenario D: Phishing / Credential Harvesting (10%)
 - Sends a link ("my portfolio", "a paper I wrote", "calendar invite")
 - Link leads to credential harvesting page
 - Goal: access to his accounts or institutional systems
- Scenario E: She Disappears (10%)
 - The move to personal email broke her script
 - [REDACTED]
 - Thread dies

Category	What I actually built	What reached production
Tool generation	the UT Austin Dorkbot parser · A NIC/NUMA/IRQ config generator · a 20-feed intel aggregator	The generator is in use. The aggregator still runs, and **21% of its rows are dead.**
Forensics and IR	A CVE hunt, published package · two report-to-intel pipelines · an affidavit extraction	One package published with anonymization and tests.
Detection scripts	A C2 beacon detector · a Bayesian anomaly detector · <u>a social-engineering campaign detector</u> · a cloud-asset intel package · a scan-detector modernisation · two blocking-pipeline monitors	Four deployed. Two live on one sensor, cluster path proven by btest. Rest in “improvement”
Operational debugging	Four investigations in three weeks, each with a written method. Then 300 GB proxies, and one bloom filter misconfig.	Three reusable runbooks. One upstream fix merged.

Everything on this slide is checkable; the companion paper cites each one

Ship the detector, not the Intel

The package

- 20 weighted body phrases from the data; structural amplifiers (**plain-text MIME, freemail entropy, no links**) fire only **after** a body match
- `BodyIndicators` at 6.0, `CampaignMatch` at 8.0 — 10 notices, 4 of them page a human
- `VictimReply` fires the moment one of ours answers — the transition the first pass could not see
- Confirmed senders auto-feed the Intel framework; internal senders excluded so a forwarded copy cannot poison it
- `se_campaign.log`, cluster-aware state, **25 tests over 3 anonymised pcaps**

The honest column

As stated	What was true
Not past Phase 2	One target was through Phase 4
"No one else replied" — self-scored LOW	A second scientist had replied
Persona A's address: not identified	Still unknown; forwarding lost the headers
Foreign collection as primary motive	Left at MODERATE; fraud not excluded

The row it marked LOW is the one that mattered.



Category	Indicator	Weight	Example
Persona fingerprint	Amy Bloom bio	4.0	"grew up in Monaco...studied design in Paris"
Persona fingerprint	Amy Bloom work description	4.0	"private art collections...scientific philanthropy"
Persona fingerprint	Lillian Briger bio	4.0	"grew up in Brunei...business administration...Stanford"
Persona fingerprint	Lillian Briger work description	4.0	"advising royal families...education planning"
Persona signature	Known sign-off	5.0	"Warm regards...Amy Bloom" / "Lillian Briger"
Manipulation	Guilt phrase	5.0	"Are you unhappy with what I said"
Manipulation	Emotional pressure	5.0	"didn't sit right with you"
Authority claim	Royal families	3.0	"collaborating with several royal families"
Authority claim	Family offices	3.0	"European and American family offices"
Question template	Broader question	3.0	"very curious to hear your perspective on one broader question"
Question template	Binary framing	2.5	"Will the biggest breakthroughs come from"
Question template	Transformative advances	2.0	"Where do you see the most transformative"
Closing template	Low-commitment ask	2.5	"If you ever have a moment to share"
Closing template	Flattery close	2.0	"I would truly/genuinely value/enjoy hearing"
Closing template	Scientist flattery	2.0	"Conversations with scientists...incredibly insightful"
Infrastructure	Long-term interest	1.5	"long-term scientific/education infrastructure"
Philanthropy	Support claim	1.5	"supporting research, education, and cultural projects"
Flattery	Template phrase	1.0	"It is fascinating/remarkable how"
Flattery	Research opener	1.0	"I recently came across your work"
Location harvesting	Location question	3.0	"what area are you based in"



Notice Types

Notice	Severity	Trigger	Email Alert
KnownSender	High	Sender address, display name, pattern, or learned IOC matches	No
SubjectMatch	Medium	Subject line matches campaign regex pattern or learned subject	No
BodyIndicators	High	Body content score exceeds body threshold (default: 6.0)	No
CampaignMatch	Critical	Combined score exceeds campaign threshold (default: 8.0). Email suppressed for 1hr per subject (deduplicates mailing list fan-out)	Yes
SenderSpray	Medium	Single freemail sender emails 500+ unique internal recipients in 24h	No
VictimReply	Critical	Internal user sent a reply to a detected campaign email	Yes
VIPTargeted	Critical	Campaign email targeted a configured high-value recipient	Yes
DisplayNameReuse	High	Known display name appeared with an unrecognized, non-internal sender address	No
InternalForwarding	High	Internal user forwarded a campaign email to colleagues (lateral propagation)	No
ConversationEscalation	Critical	Attacker followed up after victim replied (active conversation detected)	Yes
CampaignWave	High	3+ unique senders using the same display name within 1 hour (coordinated wave)	No



main 1 Branch 0 Tags

Go to file Code

initcont Updated Readme	602230 6 months ago 6 Commits
scripts	Reducing FP for CampaignMatch 6 months ago
tests	Phishing Campaign heuristics 6 months ago
README.md	Updated Readme 6 months ago
zkg.meta	Phishing Campaign heuristics 6 months ago

README

SocialEngineering

[Make sure you run in dev environment and redeef all the configurations to your needs - I yet don't have a filter on people reporting this as phish or such - so those legit reports and their responses also get flagged - in other words - detection is a bit strict]

A multi-layered Zeek package for detecting a coordinated social engineering campaign targeting research institutions. Built from real-world threat intelligence gathered from 120+ phishing emails that used fabricated personas to target 45+ scientists, engineers, and operational staff at a national research laboratory.

The package goes beyond simple IOC matching — it scores email bodies against 20 behavioral indicators extracted from the campaign corpus, tracks detected campaign emails across sessions, detects campaign waves using disposable addresses, tracks full attacker-victim conversation lifecycles, and alerts in real time when an internal user replies to or forwards a phishing email.

Threat Background

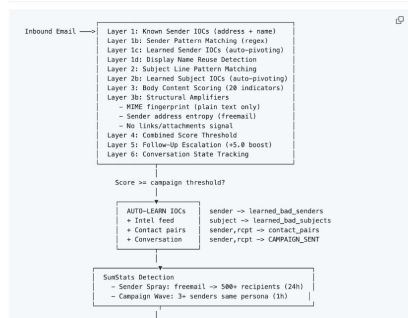
The campaign uses two fabricated personas:

Persons	Claimed Background	Targeting
Amy Bloom	Moracco upbringing, design in Paris, finance in the US. "Private art collections, long-term investment structures, scientific philanthropy."	Senior scientists, lab leadership, high-profile researchers
Lillian Briger	Brunel upbringing, business administration at Stanford. "Advising royal families on next-generation education planning."	Operations staff, facility managers, policy researchers

Both personas send hyper-personalized emails referencing the target's specific research, ask flattering binary questions ("Will the biggest breakthroughs come from X or Y?"), and close with low-commitment requests ("If you ever have a moment to share a quick thought..."). The emails contain no malicious links or attachments — they are pure social engineering designed to open a communication channel for later exploitation.

At least one target was successfully compromised: a retired scientist shared his personal email, cell phone number, physical location, and offered to meet the attacker in person.

Detection Architecture



About

A multi-layered Zeek package for detecting a coordinated social engineering campaign targeting research institutions.

Readme

Activity

Stars

Watching

Tolls

Report repository

Releases

No releases published

Contributors 1

initcont Aashish Sharma

Languages

Zeek 100%



47901	Aug 28	zeek@phish.lbl.	(15)	[IR (phish) zeek]	SocialEngineering::CampaignMatch
47958	Aug 28	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::VictimReply
47963	Aug 28	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::ConversationEscalation
48110	Aug 28	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::VictimReply
48187	Aug 29	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::ConversationEscalation
48193	Aug 29	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::VictimReply
48243	Aug 30	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::ConversationEscalation
48272	Aug 30	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::VictimReply
48287	Aug 30	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::ConversationEscalation
48289	Aug 30	zeek@phish.lbl.	(13)	[IR (phish) zeek]	SocialEngineering::VictimReply

Message: CRITICAL: Active attacker-victim conversation. Attacker anna823565@gmail.com is following up with victim [REDACTED]@lbl.gov who previously replied

Sub-message: From: anna823565@gmail.com | To: [REDACTED]@lbl.gov | Subject: Re: A question about single molecule structures

Connection: 10.10.0.1:42060 → 10.0.0.1:25

Connection uid: CNBXLA2Yz9IiMShAla

Email Extensions

zeek-scripts.tar.gz

--

[Automatically generated]

Search across your environment to find any email. Results match the exact phrase entered. Regular expressions are supported for select fields (see [documentation](#) for details). Search results display one row per message per recipient, so when searching by recipient, you may see rows for other recipients of the same email. Remediation in Search & Respond moves selected messages to the designated location, either the Recoverable Items folder or Inbox, effectively dealing with any misdirected messages, unwanted mail, or malicious mail in your environment.

Lawrence Berkeley National Laboratory

Filter By

Abnormal

anna823565@gmail.com

Aug 11, 2026 - Sep 9, 2026

13 records matched your search.

Select All 13 Results

Remediate

☐ Received	Subject	Recipient	Sender	Email Location	Abnormal Judgement	Judgement Source
☐ Sep 7, 2026, 9:56am PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Sep 4, 2026, 8:06pm PDT	Re: A question about designing batteries for recycling	@lbl.gov	anna823565@gmail.com	Trash	Safe	Abnormal System
☐ Sep 4, 2026, 12:36pm PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Sep 1, 2026, 8:08pm PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Aug 30, 2026, 1:20pm PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Aug 30, 2026, 1:00pm PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Aug 30, 2026, 12:59pm PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Aug 30, 2026, 12:37pm PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Aug 30, 2026, 5:14am PDT	Re: A question about designing batteries for recycling	@lbl.gov	anna823565@gmail.com	Spam	Safe	Abnormal System
☐ Aug 30, 2026, 5:12am PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Aug 29, 2026, 5:39pm PDT	Re: A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Aug 28, 2026, 6:00pm PDT	A question about single molecule structures	@lbl.gov	anna823565@gmail.com	Inbox	Safe	Abnormal System
☐ Aug 28, 2026, 5:59pm PDT	A question about designing batteries for recycling	@lbl.gov	anna823565@gmail.com	Trash	Safe	Abnormal System



Search across your environment to find any email. Results match the exact phrase entered. Regular expressions are supported for select fields (see [documentation](#) for details). Search results display one row per message per recipient, so when searching by recipient, you may see rows for other recipients of the same email.

Remediation in Search & Respond moves selected messages to the designated location, either the Recoverable Items folder or Inbox, effectively dealing with any misdirected messages, unwanted mail, or malicious mail in your environment.

Message Discovery Activity Log

Download Spreadsheet

Lawrence Berkeley National Laboratory Filter By Abnormal anna823565@gmail.com Aug 11, 2026 - Sep 9, 2026

13 records matched your search. Select All 13 Results

Remediate

Vendor mail-security console showing the same campaign after reporting. No count from this product is compared with Zeek's here.

Re: A question about single molecule structures @lbl.gov anna823565@gmail.com Inbox Safe Abnormal System

Search & Respond > Message Discovery

Campaign marked as attack by Abnormal on Sep 8, 2026, 12:14pm PDT.

Re: A question about single molecule structures
Received: Sep 07, 2026, 9:56 AM PDT

ATTACK SCORE
100

ATTACK TYPE
Scam

ATTACK ANALYSIS
Name Impersonation
None / Others
Text

Analysis Overview

Abnormal has detected this as a possible Scam for the following reasons:

UNUSUAL SENDER

The sender's email address anna823565@gmail.com is rarely seen in messages to this company, having appeared in only 11 out of over 99 million total safe emails received by the organization. Additionally, the sender's display name 'anna' is rare in emails to or from this company, making the sender overall uncommon and suspicious.

POTENTIAL SCAM OR EXTORTION

The email body contains vocabulary associated with scam attempts, including repeated polite follow-ups and requests for a reply, which are common tactics used to pressure the recipient into responding. However, there are no explicit threats or demands for payment in the message.

Remediation Actions

Flagged as Attack
Sep 09, 2026, 12:12 PM PDT

Tenant: Lawrence Berkeley National Laboratory

Email Location CATEGORY_PERSONAL_SPAM

Remediate

1-13 of 13 Emails 1 20 / page



Category	What I actually built	What reached production
Tool generation	the UT Austin Dorkbot parser · A NIC/NUMA/IRQ config generator · a 20-feed intel aggregator	The generator is in use. The aggregator still runs, and **21% of its rows are dead.**
Detection scripts	A C2 beacon detector · a Bayesian anomaly detector · a social-engineering campaign detector · a cloud-asset intel package · a scan-detector modernisation · two blocking-pipeline monitors	Four deployed. Two live on one sensor, cluster path proven by btest. Rest in “improvement”
Operational debugging	Four investigations in three weeks, each with a written method. Then 300 GB proxies, and one bloom filter misconfig.	Three reusable runbooks. One upstream fix merged.
Forensics and IR	A CVE hunt, published package · two report-to-intel pipelines · an affidavit extraction	One package published with anonymization and tests.

Everything on this slide is checkable; the companion paper cites each one

	healthy	proxy A	proxy B
RSS	459 MB	251 GB	286 GB
user CPU	4%	59%	50%
events per minute	3,835	1,218	6
timer lag	0.0001 s	25 s	604 s

Six events a minute is not a busy process. It is a wedged one. Growth ~0.4 GB/min, four to seven days to bake, single regex compiles reaching **36 minutes**.



0.4 GB a minute, and the first three answers were wrong

A mail-heavy cluster, 4 to 7 days to bake (O)

	healthy	proxy A	proxy B
RSS	459 MB	251 GB	286 GB
user CPU	4%	59%	50%
events/min	3,835	1,218	6
timer lag	0.0001 s	25 s	604 s

Growth ~0.4 GB/min. Single `string_to_pattern()` calls reaching 36 minutes.

Two defects, compounding

Ours: the policy calls `string_to_pattern()` inside per-message loops, five call sites, recompiling against accumulated per-host state.

Zeek's: NFA state IDs come from a process-lifetime counter, and `epsilon_closure()` tracks visits in a function-static bitset indexed by them. It expands to the highest ID ever seen, never shrinks, and is `memset` on every call.

What it took to get there

- Before assistants: months chasing table expiries on the HRW path. Wrong layer entirely.
- First AI hypothesis: `Software::register`. Wrong, and dropped the same evening.
- The agent then claimed the defect was present "8.0.1 through current master." It had read the code, not tested it.
- A maintainer caught exactly that: `"the agent claims the code is unchanged, but that's different from validation."`
- A real prior fix existed, merged Dec 2025, shipped in 8.0.5 and never backported to 8.1.x.
- A maintainer first read the reproducer as already fixed, then re-read it: `"Oh! A vs D speed difference due to memset()."`

Outcome (M)

zeek/zeek **#5706**, "NFA: Drop the static closuremap," opened and merged 2026-07-20. No patch was sent. A reproducer was.



NFA: Drop the static closuremap #5706

 Merged [awelzel](#) merged 2 commits into [master](#) from [topic/awelzel/nfa-no-static-closuremap](#)  on Jul 20

 Conversation [5](#)  Commits [2](#)  Checks [33](#)  Files changed [2](#)



awelzel commented on Jul 20

Contributor ...

Since [546687a](#), the closuremap is cleared and identifiers re-used, but remained a static instance.

[@initcont](#) reports the `memset()` call within `Clear()` of the `IntSet` becomes significant overhead for subsequent trivial `string_to_pattern()` calls once the closuremap grew large. This is due to always zeroing out all of the allocated memory.

Seem easiest to just use a new instance and select the initial size to be the number of NFA states and rely on the allocator to be fast, rather than trying to find some magic constant.

cc [@vpax](#) - does that look reasonable to you?

 [NFA: Drop the static closuremap](#) 

 [1da087b](#)



 **bbannier** reviewed on Jul 20

[View reviewed changes](#)

[> src/NFA.cc](#)

 Show resolved

 [IntSet: Drop Clear\(\) method](#) 

 [715f21e](#)



vpax commented on Jul 20

Contributor ...

Yes looks reasonable. (FYI This is *ancient* code, likely pre-dates Zeek even, back when the allocation cost was a concern.) `vector<bool>` makes sense too, providing there's a guarantee that state IDs are confined to a ~contiguous range starting at 0-ish (which I think they are); so would `unordered_set`. But also reasonable to not introduce a data structure change here.

 [awelzel](#) requested a review from [bbannier](#) [2 months ago](#)



 **bbannier** approved these changes on Jul 20

[View reviewed changes](#)

bbannier left a comment

Member ...



 [awelzel](#) merged commit [b5fda8b](#) into [master](#) on Jul 20

37 of 39 checks passed

[View details](#)

 [awelzel](#) deleted the [topic/awelzel/nfa-no-static-closuremap](#) branch [2 months ago](#)



Bloomfilters ! Oh I didn't thought right ...

```
# built-bloom.zeek:9 — the blacklist holds
246,820 entries
BL_bloom = bloomfilter_basic_init(0.001,
100000000);
```

PER NODE

171.5 MiB

measured on the production
binary, not computed

× 91 NODES

15.6 GiB

permanently resident, on one
host

Rebuilt and pushed to every node on **every feed reload**:
about **26 GB of allocation in 23 seconds, every 15 minutes** — for a filter sized 400× larger than the list it holds.

No Zeek process ever died. Three service checks flapped; logging stopped.

What that did to a box already at 120%

Link	Measured
Anonymous demand	451 GiB on a 377 GiB host
Direct reclaim	pgscand/s 0 → 35,324
Swap is files on RAID5	4 KB random writes, 774 ms/page
Workers and logger block	conn.log stops growing
The monitoring plugin	18.5 s against a 10 s timeout
What anyone actually saw	CHECK_NRPE CRITICAL, 5 days

91-node single-host cluster, Aug 2026 · same constant again 3 weeks later on a second cluster: manager OOM-killed 9× in 21 h

Four incidents, four root causes

Date	Where	Symptom	Root cause	Time to cause
Aug 14	test sensor, 376 GB	663 GB anonymous memory across 44 processes; 311 of 335 GB swap used	a nine-line test script calling catch-and-release on every connection: 6,482,114 block creates vs 2 deletes per worker	≈40 min
Aug 21	a cluster manager	792 spurious cron mails in 37 days	a fixed temp filename written outside the zeekctl lock; a */10 job collides with */5 on every run	≈20 min
Aug 24	91-node single-host cluster	monitoring flapping five days; no process died	Bloom filter sized 100,000,000 for 246,820 entries: 171.5 MiB per node, ≈26 GB in 23 s every 15 min	≈2 h, mostly waiting for the event
Sept 4	112-node single-host cluster	manager OOM-killed 9 times in 21 h	the same constant: 205.7 MB to 110 peers every 5 min, on a feed rewritten unchanged by cron	23 min, 13:03 to 13:26

Twice the size of the workers, and three wrong answers about why

What stands (M)

role	n	total	each
proxy	10	80.8 GB	8.08 GB
worker	100	414.8 GB	4.15 GB
manager	1	4.2 GB	4.2 GB
logger	1	3.7 GB	3.7 GB

Ten proxies inside a 2% band, 8.04 to 8.21 GB. A proxy captures no packets.

live container entries	7,486,428
bytes per entry	1,182
three largest tables	3.09M · 2.57M · 524K
Scan::potential_scanner, proxy vs worker	187,150,990 vs 0

1,182 bytes per entry is far above a plain record. It fits the sketch-valued scanning tables, `c_distinct_peers` and `c_likely_scanner`, a HyperLogLog per key.

Candidate, not measurement.

What I retracted, same afternoon

I wrote	actually
still climbing, unbounded	plateaued: flat over 6.2 h and over a 180 s RSS sample
top two tables 0.1% apart, a paired structure	one table read from two profiling blocks; my window spanned a boundary
seven site tables declare no expiry	they do; a <code>&default=function(){...;}</code> ends a naive parse early

What would settle it

`global_container_footprints()` returns footprints by global name, the gap in Zeek 8's `prof.log`. Probe written and parse-checked. Deploying restarts the proxies, so it names the table on the next fill, not this one.



Silent failures in generated policy



Zeek accepts it. Nothing fires. Nobody knows.

What the assistant wrote	What happened
<code>sin(PI_C * x) in Zeek script</code>	fatal parse error; no such function
<code>"RSTOS" for RSTOS0</code>	parsed fine; SYN+RST scans never counted for two weeks
<code>https://pdf-fast.com/... Intel::URL</code>	loads clean; can never match. Made twice, a month apart (Apr 25, May 28); 60,422 rows of a live feed are dead
<code>Broker::publish "structurally sound" under ZeroMQ</code>	silent drop; the log never appeared in production for months
<code>"eod.meh" as the only internal domain</code>	1,603 alert emails, about 92% false; 26 of 26 btests green throughout
<code>function domain_of(addr: string)</code>	reserved type name; a @load of it made the package unloadable

Zeek accepts it. Nothing fires. Nobody knows.

What the assistant wrote	What happened
<code>sin(PI_C * x) in Zeek script</code>	fatal parse error; no such function
<code>"RSTOS" for RSTOS0</code>	parsed fine; SYN+RST scans never counted for two weeks
<code>https://pdf-fast.com/... Intel::URL</code>	loads clean; can never match. Made twice, a month apart (Apr 25, May 28); 60,422 rows of a live feed are dead
<code>Broker::publish "structurally sound" under ZeroMQ</code>	silent drop; the log never appeared in production for months
<code>"eod.meh" as the only internal domain</code>	1,603 alert emails, about 92% false; 26 of 26 btests green throughout
<code>function domain_of(addr: string)</code>	reserved type name; a @load of it made the package unloadable

```
## conn_state values counted as "failed to establish".
const failure_states: set[string] = {
    "S0", "REJ", "RSTOS", "RSTRH", "SH", "SHR",
} &redef;
```

The row that loads cleanly and can never match

What our aggregator stores (M)

```
# parsers.py:184 - the validate() gate
if intel_type == "Intel::URL":
    return value.startswith(("http://", "https://"))
```

It does not tolerate the scheme, it **requires** it. A correctly formatted row is rejected before it is ever written.

What Zeek looks up

```
## base/frameworks/intel/main.zeek:21
## A complete URL without the prefix "http://".
URL,

# http-url.zeek -> HTTP::build_url() = host + uri
# smtp-url-*.zeek -> find_all_urls_without_scheme()
# matching is exact string equality; no normalisation
```

Why nothing complained (M)

Four deliberately malformed rows, one file, one load:

row loaded		Zeek's reaction
not-an-ip	Intel::ADDR	error, names file and line
https://e.com/x	Intel::URL	accepted silently
https://e.com	Intel::DOMAIN	accepted silently
zzz	Intel::FILE_HASH	accepted silently

The bug is mine. The validator requires the one form Zeek cannot match. **The silence is the framework's.** ADDR is cast to a typed value and fails loudly; the other three are stored as strings, so a wrong-shaped string is indistinguishable from a right-shaped one that never appears.

The test that would have caught it is not "does the file load." It is five lines: fire a synthetic sighting in the form `build_url()` produces, and assert a match.

The same ticket, before and after

FILED AUG 21 (M)

1,393

words. A summary, two reproducers, a root-cause section, field data, a patch, "patch verified."

REWRITTEN (M)

264

words. Symptom and onset, verbatim error, real reproducer, one hedged code pointer, workaround, disclosure line naming the tool and what it did versus what I did. No patch !!!!

The observation was valid and is being fixed upstream. The patch was withdrawn: it silenced the error and serialized nothing, and the error was the only evidence the race had happened. The maintainer's comment on the ticket:

Limitations of this talk

1. One Aashish, one lab, one network. Nothing here generalizes by itself.
2. The cases were not designed as experiments. Tools, models, and tasks varied together.
3. The error counts are floors. They count errors that were noticed and written down.
4. Per-line authorship is not recorded in any package. "AI-assisted" is the honest label.
5. Most investigation logs were written by the assistant during the work. Their time figures and wrong turns are self-reports, corroborated by raw data timestamps where those exist.
6. Several "after" numbers are projections from a data, not post-deployment counts.

What does the record show?

About seventeen months, me, four kinds of Zeek work. What helped, what went wrong, what caught it.

The claim this evidence supports

AI-assisted work reached production or upstream when its output was checked against the running system.

When the check was skipped, or could not see the failure, the same tools produced confident wrong conclusions and scripts that parsed cleanly and never fired.

Not claimed

- No hours saved.
- No comparison with other tools or teams.
- No statement about models in general.
- One control case, used qualitatively.

1. If it parses, that tells you nothing. The thing to be afraid of is a script that loads fine and never fires.
2. All my tests passed. The detector was still wrong. A test only catches what it can see.
3. Stop writing specs. Point at a file. A CSV, an mbox, a pcap — each one better than a page of prose.
4. Don't do the arithmetic - let data speak for itself
5. Reading the code is not validating it. A maintainer had to tell me that.
6. The reproducer moves things, not the diagnosis - Mine was wrong three times.



Thanks!!

Questions

security@lbl.gov

asharma@lbl.gov

