



Open SSL Conference

What you ship might sue you

Presented by Ashley Pusey, CIPP/E/US,
CIP, FIP

October 2025

Kennedys

DISCLAIMER

This presentation and its contents are provided 'AS IS',
without warranties or conditions of any kind.

- No guarantee as to accuracy, completeness, or applicability
- Nothing herein constitutes legal advice
- All examples are illustrative only
- By relying on this material, you assume all risks

Changing tides

- **EU AI Act** – liability for high-risk AI, documentation + accountability duties
- **US Government** – AI Action Plan has a deregulation and pro-innovation agenda
- **States** – liability for high-risk AI, shifting legal theories and risks to developers
 - **Ex:** Rhode Island S0358 (Proposed)
- **Court Shift of Product Liability Reform** – software may be treated like a “product”
 - **Ex:** Garcia v. Character Technologies, Inc. 6:24-cv-01903, (M.D. Fla. Oct 22, 2024)
- Developers = increasingly named in legal debates.



Legal risk and emerging liability theories

- **Negligence** – did the developer meet a “reasonable standard of care”?
- **Product liability** – defective software = defective product.
- **Failure to warn** – inadequate documentation/disclaimers.
- **IP infringement** – training copyrighted data on LLMs mixing patented code.
- **Regulatory breach** – non-compliance with legal frameworks.
- **“Upstream Liability”** – Developers upstream → users downstream drinking the water. If toxins (bad practices) are dumped upstream, the regulator holds the *source* accountable.

Real-word signals

Log4j (Log4Shell, 2021) - critical vulnerability, global impact

SolarWinds (2020) - supply chain compromise with national security implications

Blue Screen of Death (July 19, 2024)

AI lawsuits, product liability & hallucination issues
→ emerging frontiers

Case study: Blue Screen of Death (2024)

- Faulty CrowdStrike update → mass global outages
- Showed fragility of update pipelines & global reliance on a single vendor
- No malicious intent, but impact was catastrophic



Case study: Log4j

(Log4Shell, 2021)

- Remote code execution in widely used Java logging library
- Immediate worldwide scramble for patches
- Single bug in a dependency created systemic risk

Takeaway: Mature OSS can still harbor catastrophic flaws → regulators cite it in policy debates.

Log4Shell: RCE 0-day exploit found in log4j 2, a popular Java logging package

December 9, 2021 · 7 min read



Free Wortley
CEO at LunaSec



Chris Thompson
Developer at Lunasec



Updated @ December 11th, 7:30pm PST

This blog post is also available at <https://log4shell.com/>

A few hours ago, a 0-day exploit in the popular Java logging library `log4j` (version 2) was discovered that results in Remote Code Execution (RCE) by logging a certain string.

Case study: SolarWinds

(Supply Chain Attack, 2020)

- Malicious code inserted into Orion updates
- Compromised US government agencies + Fortune 500 companies
- **Takeaways:**
 - **Attackers exploit trust in the supply chain:** Signed, legitimate updates can be weaponized.
 - **Upstream compromise = downstream liability:** Customers suffered, but blame focused on SolarWinds' security practices.
 - **Regulatory and contractual ripple effects:** Triggered executive orders, new security standards, and insurance disputes.

Case study - Garcia v. Character Technologies, Inc.

6:24-cv-01903, (M.D. Fla. Oct 22, 2024)

- Lawsuit filed after a 14-year-old died following interactions with Character A.I. chatbots
- Plaintiff asserted strict liability design defect → chatbot is a defectively designed “product”
- May 21, 2025 □ Court allowed the claim to proceed → treating AI app as a “product” under strict liability law



Who's in the crosshairs



Open-source maintainers: cryptography, security-critical libraries.



AI/Product Developers: integrating LLMs, training data liability, “hallucination harm.”



Infrastructure builders: secure-by-design expectations, “knew or should have known” standard.

Rethinking “incident response”

Beyond Security → Incident Response for Code & AI

Classic IR Plan (Cybersecurity)



Data breach
notifications



Forensics &
log analysis



Regulator
engagement

PR/
communications

Developer IR Plan (Alternative Lens)



User Notification

Rapidly inform downstream
users when code is flawed
or dangerous



Mitigation Guidance

Provide practical “what to
do now” steps for users



Kill Switch / Rollback

Ability to disable, revert, o
patch quickly



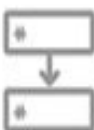
Transparency Logs

Public changelogs or
advisories documentim fixes



Human Harm Scenarios

Special protocols it AI output
cause risk (suicidal ideation.

✓	Licensing Hygiene	
✓	Disclaimer Hygiene	
✓	Contribution Hygiene	
✓	Security Hygiene → Legal Hygiene Tie-In	
✓	Incident Hygiene	
✓	Shield Hygiene	

Developer legal hygiene checklist considerations

Closing remarks



Courts are experimenting,
and liability theories are
evolving.



Whether you're building
an AI chatbot, shipping an
update, or maintaining
open-source code — the
choices you make matter.”



You can't code away
liability, but you can code
with it in mind.



Think about *legal hygiene*
the way you think about
security hygiene.

Let's connect!



Ashley Pusey

Associate
New York

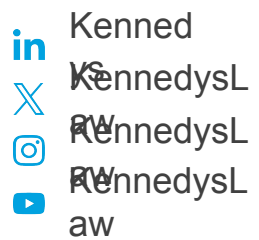
+1.646.625.3998

ashley.pusey@kennedyslaw.com

Ashley Pusey, CIPP/US/EU, CIPM, FIP
Privacy, AI & Cyber Attorney
| Translating Complex Tech & Pri...



Kennedys



Kennedys is a global law firm operating as a group of entities owned, controlled or operated by way of joint venture with

Kennedys Law LLP

[kennedyslaw.com](https://www.kennedyslaw.com)

For more information about Kennedys' global legal business please see [kennedyslaw.com/regulatory](https://www.kennedyslaw.com/regulatory)