
# Radioactive Vulnerabilities

**Radiation Fears to Digital Nightmares**


# Jennifer Shannon

- Senior Security Consultant at Secure Ideas
  - Jacksonville HQ office
- Industry Experience
  - Started as SOC Analyst
  - Reverse engineering malware & threat intelligence
  - Pentesting, Security Consulting, & Training
- Other Interesting Facts
  - All around geek
  - Collector of things
  - Lockpick enthusiast
  - My favorite game genre is Survival-Horror

Secure Ideas
professionally evil®

# Kathy Collins

**Security Consultant Secure Ideas, LLC**

- Been with Secure Ideas since 2021
- Based in Jacksonville, Florida
- Sec+, CISSP

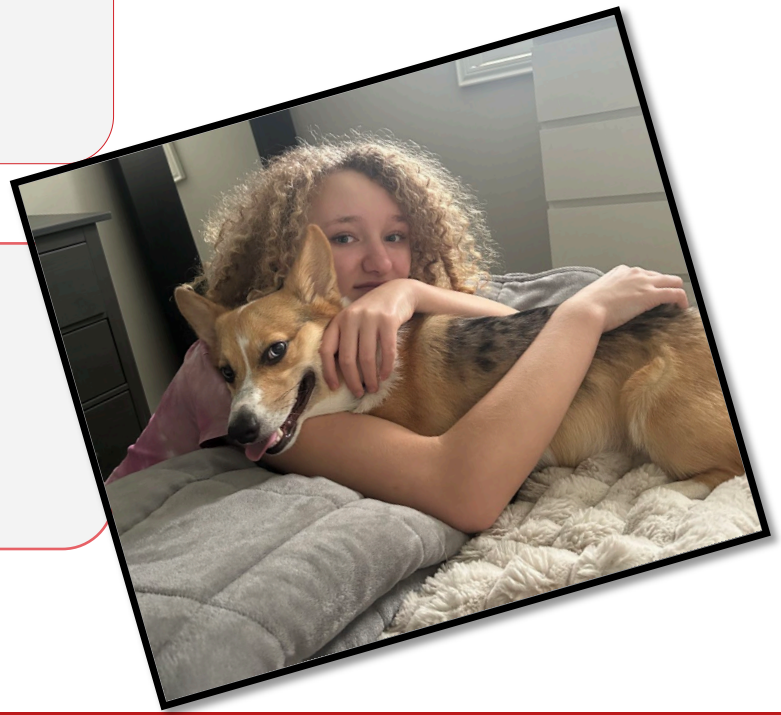

kathy.collins@secureideas.com

**Focus Areas**

- Networking
- Web Applications
- Physical Pentesting

**Other Fun Facts**

- Former Chef
- Corgi/Teen Mom
- Horror Fan
- BSides Jacksonville Coordinator
- Costco Member

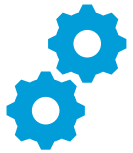


# SDLC

## Waterfall

Called waterfall because it arranges phases sequentially

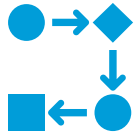
Each phase depends on outcome from previous phase



## Iterative

Breaks the software development process down into smaller segments

Intends to improve and build from the small segments



## Spiral

Combination of Waterfall and Iterative

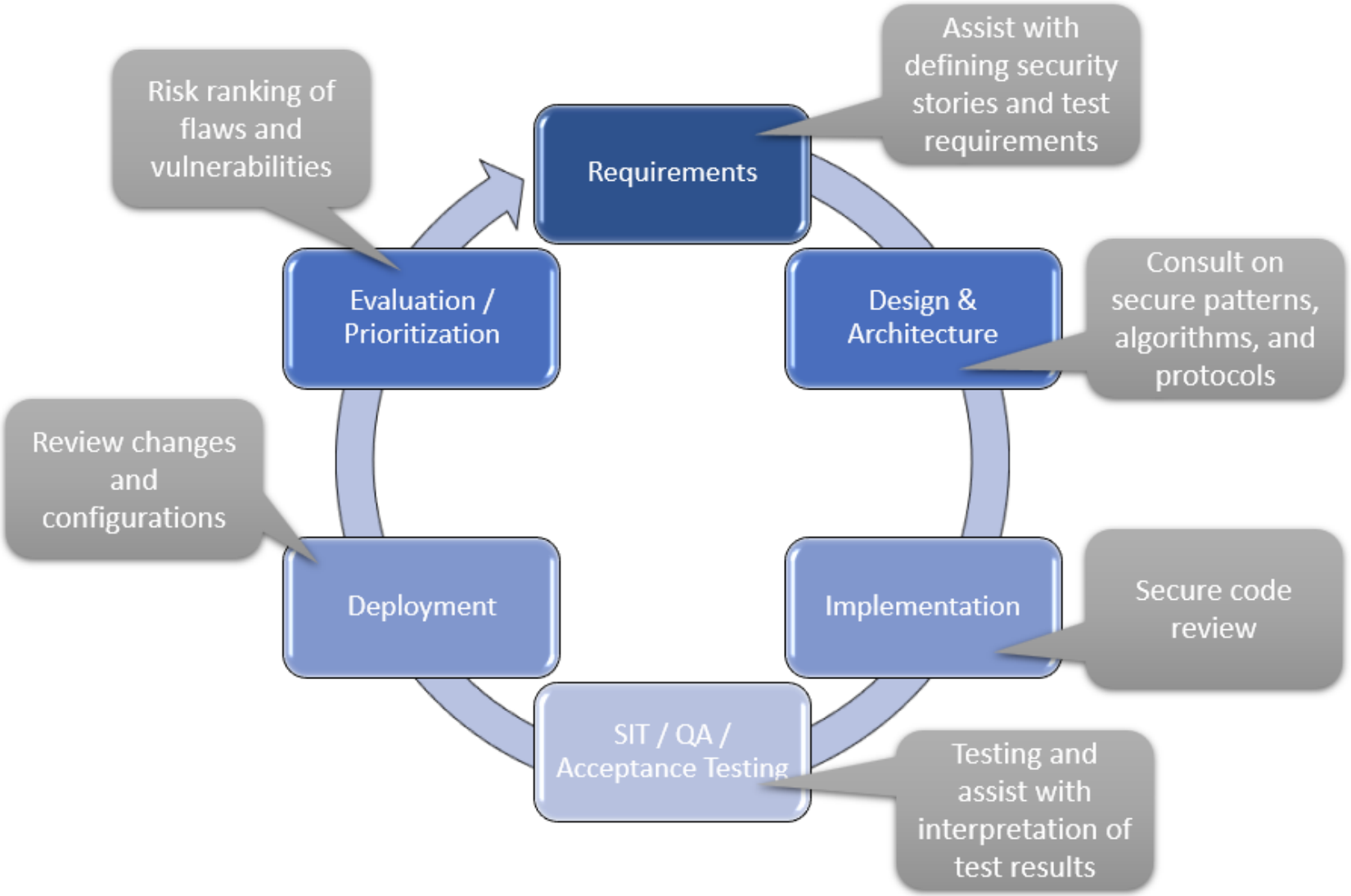
Emphasis on risk assessment

## Agile

Adopts incremental and iterative principles

Has a manifesto….

# Built with security in mind



# Requirements

- The application security team should be heavily involved in the requirements phase of the SDLC for several reasons:
  - The requirements phase is usually the best opportunity for the security team to gain visibility into upcoming features that may have security implications.
  - By participating in sprint planning, the security team can argue for the priority of security features.
  - The security team can immediately begin planning test scenarios to improve the efficiency of testing activities later in the SDLC.

# Design and Architecture

- Ideally, development teams should utilize and trust the security team as advisors during this phase
  - Security teams should:
    - Advocate for secure coding practices
    - Know what algorithms, protocols, and platforms are approved by the organization

# SIT, QA, and Acceptance

- The security team has a few tasks during the various testing activities of the SDLC, as follows:
  - In the time leading up to testing, the security team will assist with defining security test cases.
  - The security team may be responsible for some testing, such as running DAST tools or conducting penetration testing.
  - After testing, the security team will help interpret and filter the results from tooling to ensure issues delivered to the development team are relevant.

# SAST VS DAST

- Static Application Security Testing



- Dynamic Application Security Testing


# Security is a Full-Stack Responsibility

| Database | App Server | Client-side Application |
|---|---|---|
| - Use minimal privileges on Application accounts<br>- Disable direct login for these accounts<br>- When possible, disable features that weaken security e.g. xp_cmdshell<br>- Log key events<br>- Maintain backups<br>- Test the backups | - Auth check every request<br>- Filter and validate input<br>- Use parameterized queries to protect the DB<br>- Use TLS Encryption<br>- Encode user-supplied output<br>- Protect against CSRF in concert with the client<br>- Supply correct security headers on responses e.g. CORS policy, flags on cookies<br>- Avoid incorporating user input into system commands<br>- Log key events (after stripping sensitive data) | - Coordinate with the App Server against common client-side attacks such as CSRF<br>- Avoid DOM manipulation that unsafely incorporates user input<br>- Implement a content-security-policy when possible. Keep it as restrictive as possible. |

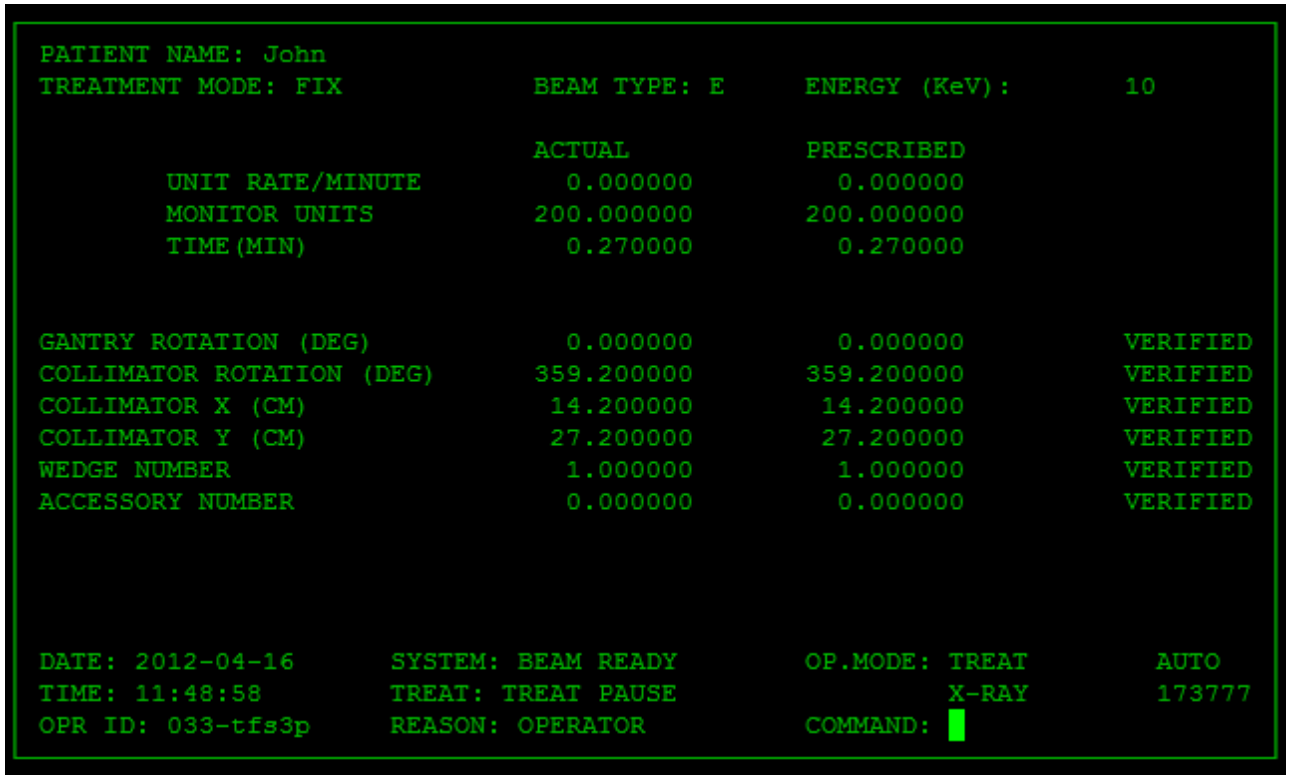



# Case Study: Therac-25

- The Therac-25 was produced by Atomic Energy of Canada Limited (AECL) in 1982 as a computer-controlled radiation therapy machine, following the Therac-6 and Therac-20 models

# Malfunctions & Accidents

- Between 1985 and 1987, at least six accidents occurred with the Therac-25, resulting in patients receiving massive overdoses of radiation due to software errors.
- Patients experienced radiation doses hundreds of times greater than intended, leading to fatalities and serious injuries.
- The accidents were attributed to concurrent programming errors (race conditions) that caused the machine to administer lethal doses of radiation.



# Incidents

July 25th, 1985

Patient received a massive overdose due to an error message "H-tilt" displayed by the machine

# Deadly Code

- The source code has never been publicly released
- The code snippet represents a simplified version of the logic in the Therac-25 machine that may have contributed to the fatal radiation overdoses.
- The presence of a race condition in the software allowed the machine to fire a concentrated X-Ray beam when it should have continued normal operation. This flaw, combined with other issues in the software, led to the horrific catastrophe

```
 GNU nano 6.2                                therac_Sample *
// Code snippet from the Therac-25 machine that led to fatal radiation overdoses
if (Class3 == 1) {
    // Concentrated X-Ray beam firing codepath
    FireXRayBeam();
} else {
    // Normal operation codepath
    ContinueNormalOperation();
}
```

# Manufacturer Responses

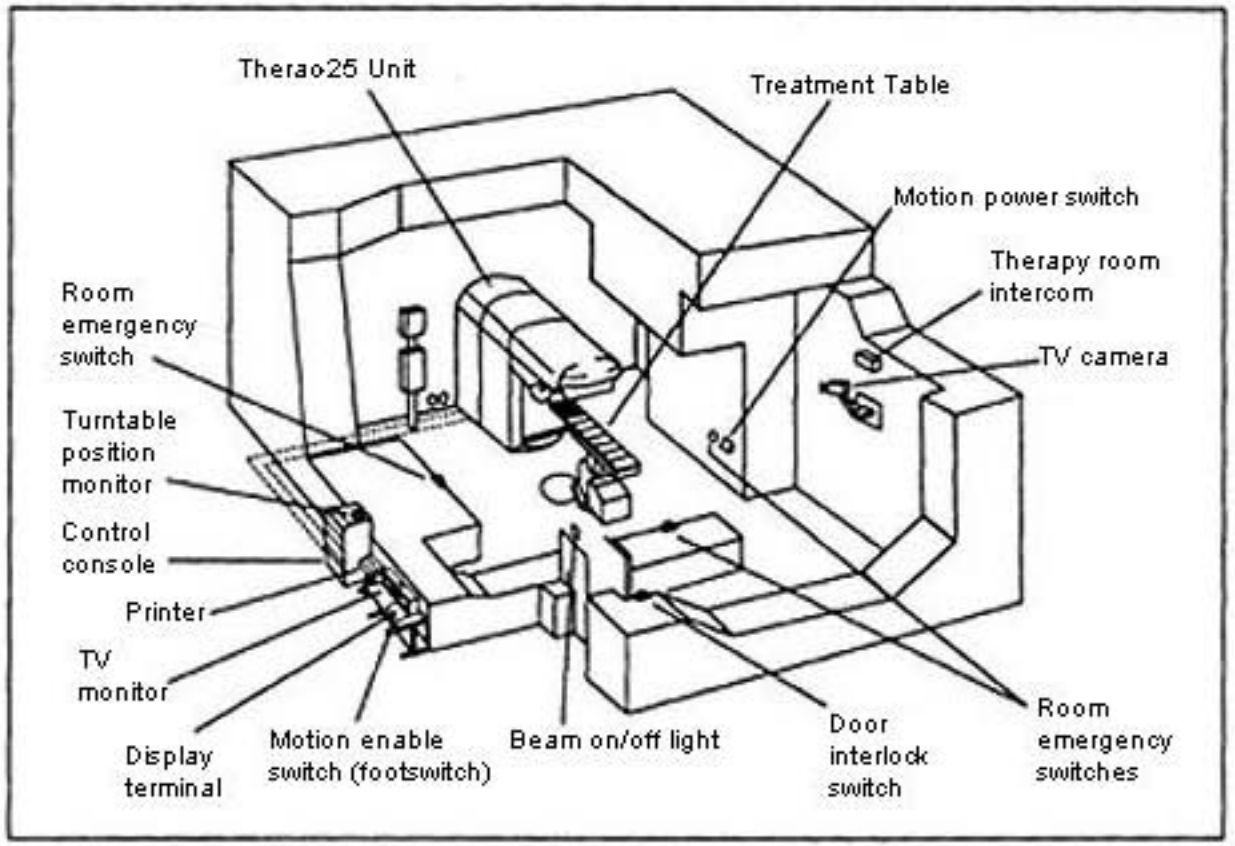


Figure 1. Typical Therac-25 facility

- They denied that the machine could have caused the radiation burns and overdoses experienced by patients.
- They refused to believe that the incidents were linked to the Therac-25 machine.
- At the time of the accidents, the treatment prescription printout feature was disabled, leading to a lack of hard copy treatment data.
- The manufacturer and operators did not acknowledge the machine's role in causing the severe radiation burns until later investigations revealed software errors and issues with the machine's safety mechanisms
- The manufacturer's response included making extensive design changes to the Therac-25 machine, including implementing hardware safeguards against software errors. These changes were made after the machine was recalled in 1987 following the series of accidents


# Investigations

The FDA (Food and Drug Administration) declared the Therac-25 defective under the Radiation Control for Health and Safety Act.

The FDA mandated that the manufacturer submit a corrective action plan (CAP) for approval, which included over 20 changes to the system hardware and software to enhance safety measures


# Legal Actions

Several victims or families of deceased patients filed lawsuits against the manufacturer, AECL (Atomic Energy of Canada Limited).

These lawsuits were settled out of court.

# Questions