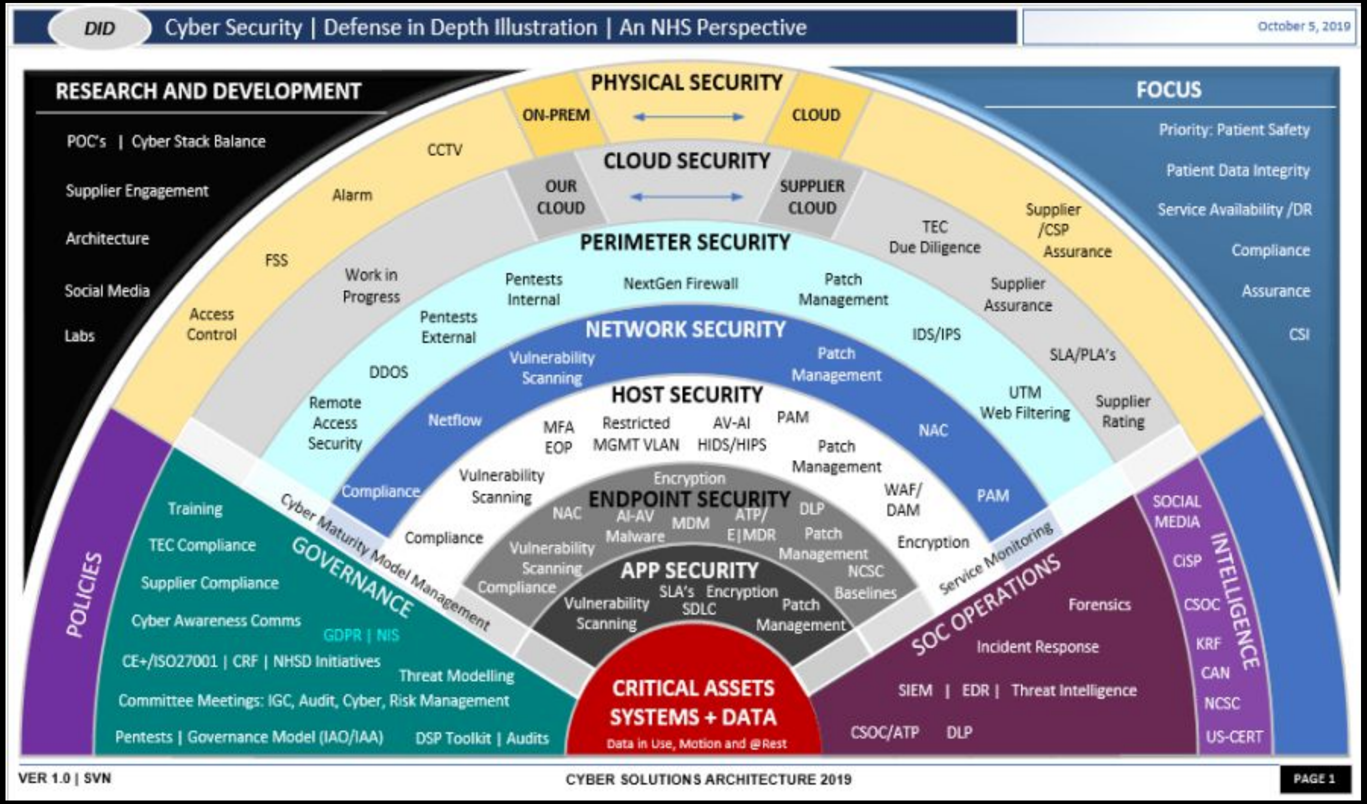
Green Pace

Security Policy Presentation
Developer: Danielle McNeill

Green Pace

OVERVIEW: DEFENSE IN DEPTH



- Protect against common vulnerabilities
- Support defense-in-depth
- Consistent and maintainable
- SEI CERT C++ standards
- Protect against security breaches

Green Pace

THREATS MATRIX

Rule	Severity	Likelihood	Remediation Cost	Priority	Level
STD-001-CPP	High	Likely	Low	High	4
STD-002-CPP	High	Likely	Low	High	4
STD-003-CPP	Medium	Unlikely	Low	Medium	3
STD-004-CPP	High	Unlikely	Low	High	4
STD-005-CPP	High	Unlikely	Medium	High	4
STD-006-CPP	Medium	Likely	Low	Medium	3
STD-007-CPP	Medium	Likely	Low	High	4
STD-008-CPP	High	Likely	Low	High	5
STD-009-CPP	High	Likely	Low	High	4
STD-010-CPP	Medium	Unlikely	Medium	Medium	3

Green Pace

10 PRINCIPLES

- Validate Input Data – STD-009
- Heed Compiler Warnings – STD-006
- Architect for Security – STD-008
- Keep It Simple – STD-004, STD-006, STD-010
- Default Deny – STD-008
- Adhere to Standards – STD-001 to STD-010
- Sanitize Outputs – STD-009
- Practice Defense in Depth – STD-007
- Use QA Techniques – STD-007, STD-010
- Secure Defaults – STD-002, STD-003

Green Pace

CODING STANDARDS

1. STD-008: Never Hard Code Secrets (Level 5)
2. STD-009: Format String Injection (Level 4)
3. STD-002: Validate Integers (Level 4)
4. STD-001: Avoid Integer Overflow (Level 4)
5. STD-005: Limit Pointer Arithmetic (Level 4)
6. STD-004: Array Bounds Check (Level 4)
7. STD-007: Handle Library Errors (Level 4)
8. STD-006: Non-Void Function Returns (Level 3)
9. STD-003: Implicit Type Conversions (Level 3)
10. STD-010: Avoid Signals in Threads (Level 3)

Green Pace

ENCRYPTION POLICIES

- Encryption at rest
    - what it is
    - SHA-256 policy
    - how it applies
- Encryption in flight
    - what it is
    - HTTP and VPN policy
    - how it applies
- Encryption in use
    - what it is
    - secure environments
    - how it applies

Green Pace

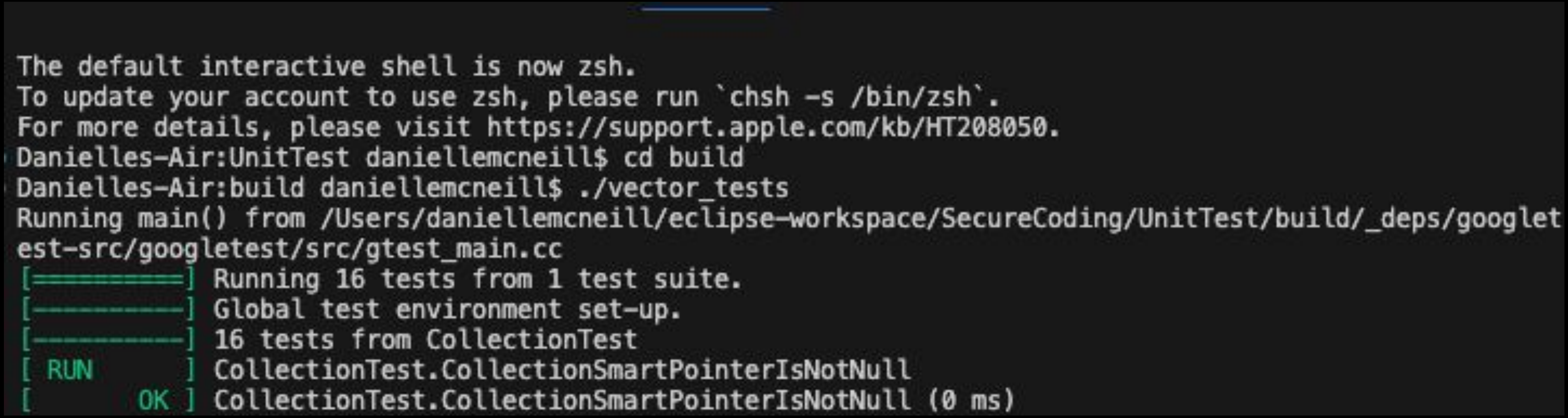
TRIPLE-A POLICIES

- Authentication
  - what it is
  - MFA policy
  - how it applies
- Authorization
  - what it is
  - role-based access policy
  - how it applies
- Accounting
  - what it is
  - tracking/logging policy
  - how it applies

Green Pace

CollectionSmartPointerIsNotNull

- Positive test
- Passed
  - smart pointer is valid and not null

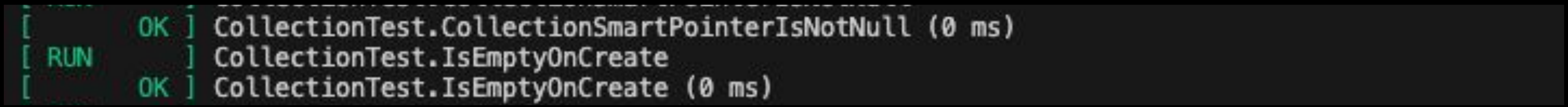


```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Danielles-Air:UnitTest daniellemcneill$ cd build
Danielles-Air:build daniellemcneill$ ./vector_tests
Running main() from /Users/daniellemcneill/eclipse-workspace/SecureCoding/UnitTest/build/_deps/googlet
est-src/googletest/src/gtest_main.cc
[==========] Running 16 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 16 tests from CollectionTest
[ RUN      ] CollectionTest.CollectionSmartPointerIsNotNull
[       OK ] CollectionTest.CollectionSmartPointerIsNotNull (0 ms)
```

Green Pace

IsEmptyOnCreate

- Positive test
- Passed
  - vector is empty and has size 0

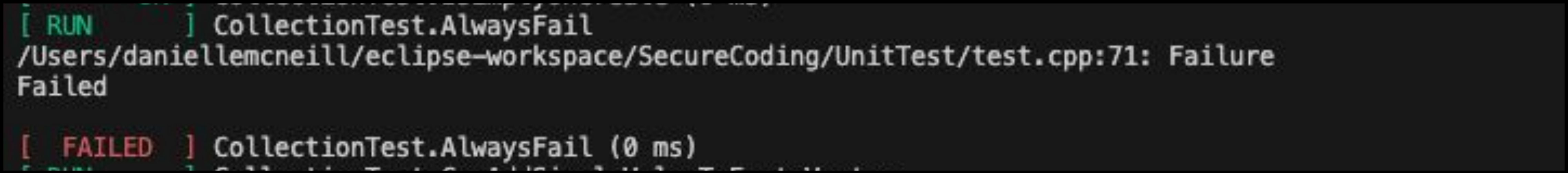


```
[       OK ] CollectionTest.CollectionSmartPointerIsNotNull (0 ms)
[ RUN      ] CollectionTest.IsEmptyOnCreate
[       OK ] CollectionTest.IsEmptyOnCreate (0 ms)
```

Green Pace

AlwaysFail

- Negative test
  - fails intentionally
  - confirms that the test is correct



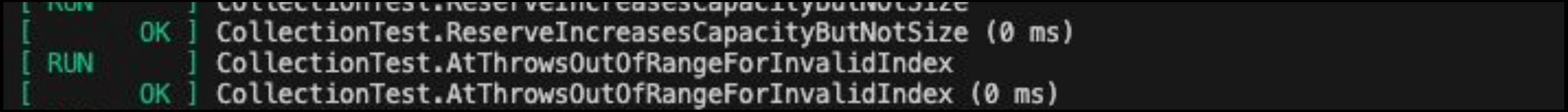
```
[ RUN      ] CollectionTest.AlwaysFail
/Users/daniellemcneill/eclipse-workspace/SecureCoding/UnitTest/test.cpp:71: Failure
Failed

[  FAILED  ] CollectionTest.AlwaysFail (0 ms)
```

Green Pace

AtThrowsOutOfRangeForInvalidIndex

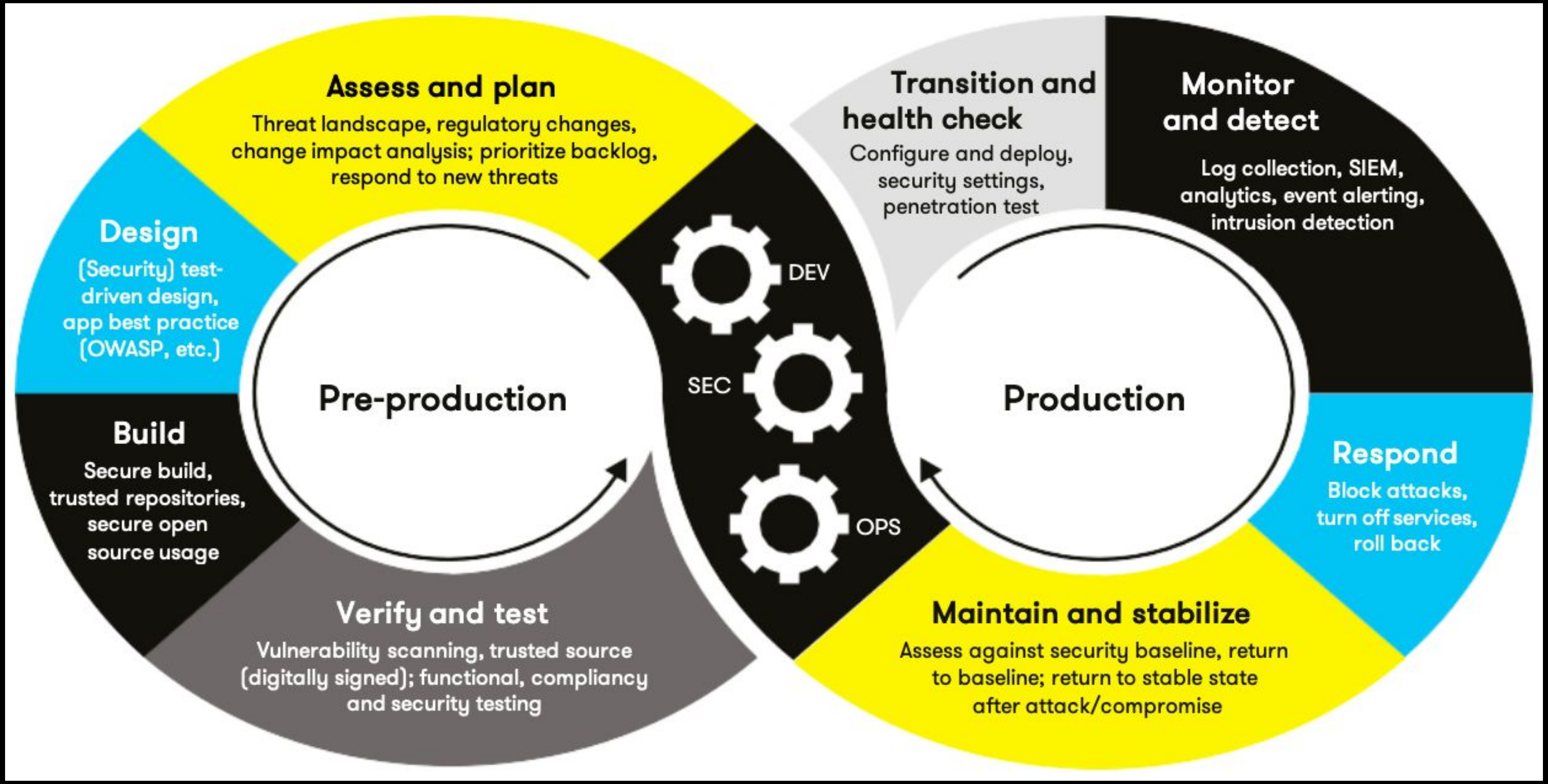
- Negative test
  - test passes only if out of range exception is thrown



```
[       OK ] CollectionTest.ReserveIncreasesCapacityButNotSize (0 ms)
[ RUN      ] CollectionTest.AtThrowsOutOfRangeForInvalidIndex
[       OK ] CollectionTest.AtThrowsOutOfRangeForInvalidIndex (0 ms)
```

Green Pace

AUTOMATION SUMMARY


Assess and plan
Threat landscape, regulatory changes, change impact analysis; prioritize backlog, respond to new threats
Design
(Security) test-driven design, app best practice (OWASP, etc.)
Build
Secure build, trusted repositories, secure open source usage
Verify and test
Vulnerability scanning, trusted source (digitally signed); functional, compliancy and security testing
Pre-production
DEV
SEC
OPS
Transition and health check
Configure and deploy, security settings, penetration test
Monitor and detect
Log collection, SIEM, analytics, event alerting, intrusion detection
Production
Respond
Block attacks, turn off services, roll back
Maintain and stabilize
Assess against security baseline, return to baseline; return to stable state after attack/compromise

Green Pace

TOOLS

- Assess & Plan
  - Cppcheck
- Design
  - Secure architecture
- Build
  - Cppcheck, Clang++
- Test
  - Unit tests
- Deploy
  - Fortify
- Monitor
  - Logging frameworks
- Maintain
  - Update policies and tools regularly

Green Pace

RISKS AND BENEFITS

- Waiting
  - vulnerabilities go undetected
  - inconsistent coding
  - higher cost
  - increased security risk
- Act Now
  - enforce SEI CERT standards
  - Integrate security at the start
  - catch and correct problems early
  - maintain consistency
- Risks vs. benefits
  - increased debt, more risk of attack, reactive policy
  - lower cost, stronger protections, proactive policy

Green Pace

RECOMMENDATIONS

- Close existing gaps in memory safety and privilege access
- Enforce standard adoption through CI pipelines
- Expand testing coverage and automation
- Monitor compliance continuously

Green Pace

CONCLUSIONS

- Add secure logging standards
- Integrate container security checks
- Expand coverage to third-party libraries
- Introduce policies for zero-trust environments

Green Pace

REFERENCES

- Fortinet Inc. (n.d.). What is AAA security? In CyberGlossary. Fortinet. Retrieved July 31, 2025, from https://www.fortinet.com/resources/cyberglossary/aaa-security
- Cybersecurity and Infrastructure Security Agency. (n.d.). Cybersecurity best practices. U.S. Department of Homeland Security. Retrieved August 11, 2025, from https://www.cisa.gov/topics/cybersecurity-best-practices
- Djalovic, N. (2025, April 8). A comprehensive guide to data encryption: Data at rest, in motion, and in use. Jatheon. https://jatheon.com/blog/data-at-rest-data-in-motion-data-in-use/
- Software Engineering Institute. (n.d.). SEI CERT C++ Coding Standard. Retrieved July 11, 2025, from https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88046682
- OWASP Foundation. (n.d.). Secure coding practices quick reference guide: Checklist. OWASP. https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stable-en/02-checklist/05-checklist
- Thurmond, T. (2023, December 27). 8 best secure coding practices learned from OWASP. KirkpatrickPrice. https://kirkpatrickprice.com/blog/secure-coding-best-practices/

Green Pace