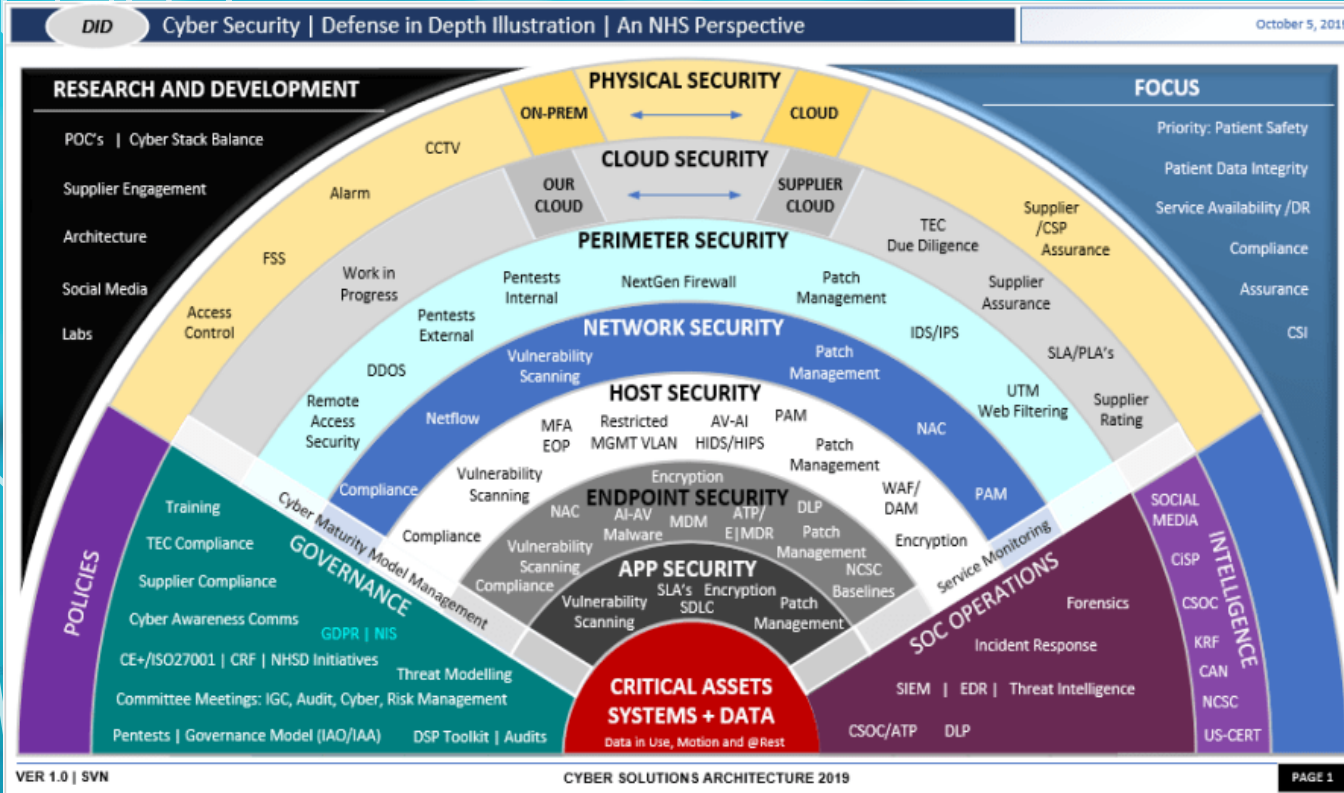


SECURITY POLICY PRESENTATION

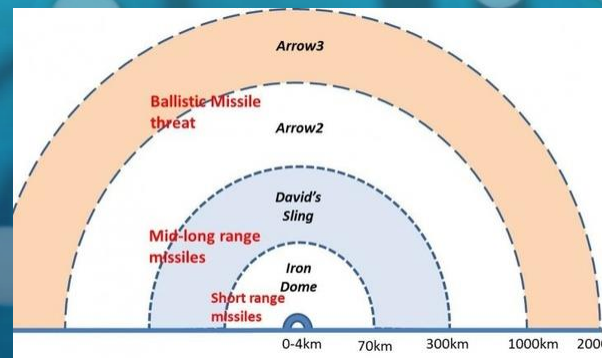
BY: DANI YOSOPOV





DEFENSE-IN-DEPTH

- Defense in Depth (DiD) is an approach in cybersecurity its series of defensive mechanisms are layered in order to protect valuable data and information. If one mechanism fails, another step is up immediately to encounter the attack.



THREATS MATRIX TABLE

	Impact →				
	Negligible	Minor	Moderate	Significant	Severe
Likelihood ↑	Very Likely	Low	Moderate	High	High
	Likely	Low	Moderate	High	High
	Possible	Low	Low	Moderate	High
	Unlikely	Low	Low	Moderate	Moderate
	Very Unlikely	Low	Low	Low	Moderate

- Levels of threats

Secure Coding has different levels of threats that are measured by two factors likelihood and Impact.

- DevSecOps

Development, security, and operations automate the integration of security at every phase of the software development lifecycle, from initial design through integration, testing, deployment, and software delivery.



PRINCIPLES

PRINCIPLES

1. Validate Input Data
2. Heed Compiler Warnings
3. Architect and Design for Security Policies
4. Keep It Simple
5. Default Deny
6. Adhere to the Principle of Least Privilege
7. Sanitize Data Sent to Other Systems
8. Practice Defense in Depth
9. Use Effective Quality Assurance Techniques
10. Adopt a Secure Coding Standard





CODING STANDARDS

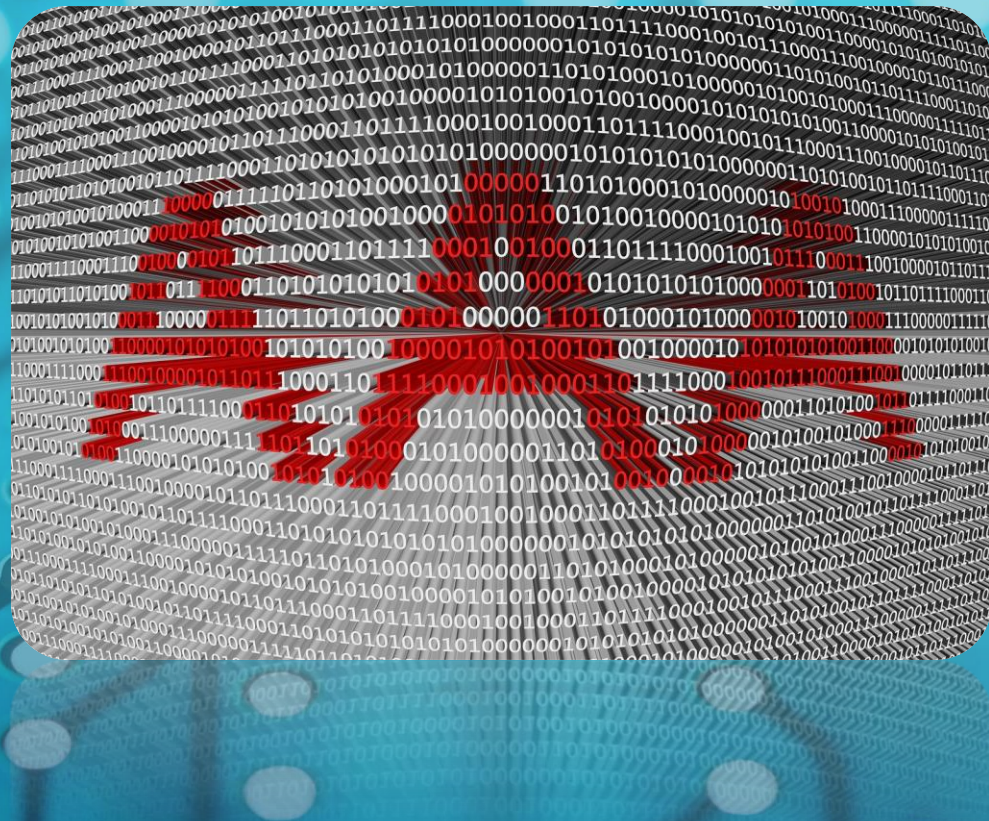
1. Data Type - Do not cast to an out-of-range enumeration value.
2. Data Value - Ensure that division and remainder operations do not result in divide-by-zero errors.
3. String Correctness - Do not attempt to modify string literals.
4. SQL Injection - Sanitize data passed to complex subsystems.
5. Memory Protection - Do not access freed memory.
6. Assertions - Incorporate diagnostic tests using assertions.
7. Exceptions - Honor exception specifications.
8. Object Oriented Programming (OOP) - Do not invoke virtual functions from constructors or destructors.
9. Exceptions and Error Handling (ERR) - Guarantee exception safety.
10. Declarations and Initialization (DCL) - Do not define a C-style variadic function.



ENCRYPTION STRATEGY

- Encryption in rest.
- Encryption at flight.
- Encryption in use





TRIPLE-A FRAMEWORK

- Authentication
- Authorization
- Accounting



```

Running main() from c:\a\1\s\thirdparty\googletest\googletest\src\gtest_main.cc
[=====] Running 16 tests from 1 test case.
[-----] Global test environment set-up.
[-----] 16 tests from CollectionTest
[ RUN      ] CollectionTest.CollectionSmartPointerIsNotNull
[ OK       ] CollectionTest.CollectionSmartPointerIsNotNull (0 ms)
[ RUN      ] CollectionTest.IsEmptyOnCreate
[ OK       ] CollectionTest.IsEmptyOnCreate (0 ms)
[ RUN      ] CollectionTest.AlwaysFail
C:\Users\dani1\source\repos\week4 unit_test\test.cpp(85): error: Failed
[ FAILED   ] CollectionTest.AlwaysFail (1 ms)
[ RUN      ] CollectionTest.CanAddToEmptyVector
[ OK       ] CollectionTest.CanAddToEmptyVector (0 ms)
[ RUN      ] CollectionTest.CanAddFiveValuesToVector
[ OK       ] CollectionTest.CanAddFiveValuesToVector (0 ms)
[ RUN      ] CollectionTest.CheckMaxSizeBiggerOrEqualTo0_1_5_10
[ OK       ] CollectionTest.CheckMaxSizeBiggerOrEqualTo0_1_5_10 (0 ms)
[ RUN      ] CollectionTest.CheckcapacityBiggerOrEqualTo0_1_5_10
[ OK       ] CollectionTest.CheckcapacityBiggerOrEqualTo0_1_5_10 (0 ms)
[ RUN      ] CollectionTest.CheckResizingIncreasesCollection
[ OK       ] CollectionTest.CheckResizingIncreasesCollection (0 ms)
[ RUN      ] CollectionTest.CheckResizingDecreasesCollection
[ OK       ] CollectionTest.CheckResizingDecreasesCollection (0 ms)
[ RUN      ] CollectionTest.CheckResizingDecreasesCollectionToZero
[ OK       ] CollectionTest.CheckResizingDecreasesCollectionToZero (0 ms)
[ RUN      ] CollectionTest.CheckClearErasesTheCollection
[ OK       ] CollectionTest.CheckClearErasesTheCollection (0 ms)
[ RUN      ] CollectionTest.CheckEraseBeginEndErasesTheCollection
[ OK       ] CollectionTest.CheckEraseBeginEndErasesTheCollection (0 ms)
[ RUN      ] CollectionTest.CheckReserveIncreasesTheCapacityButNotTheSizeOfTheCollection
[ OK       ] CollectionTest.CheckReserveIncreasesTheCapacityButNotTheSizeOfTheCollection (0 ms)
[ RUN      ] CollectionTest.CheckOutOfRangeException
[ OK       ] CollectionTest.CheckOutOfRangeException (1 ms)
[ RUN      ] CollectionTest.1_positive
[ OK       ] CollectionTest.1_positive (0 ms)
[ RUN      ] CollectionTest.1_negative
C:\Users\dani1\source\repos\week4 unit_test\test.cpp(206): error: Value of: collection == NULL
Actual: false
Expected: true
[ FAILED   ] CollectionTest.1_negative (2 ms)
[-----] 16 tests from CollectionTest (17 ms total)

[-----] Global test environment tear-down
[=====] 16 tests from 1 test case ran. (22 ms total)
[ PASSED   ] 14 tests.
[ FAILED   ] 2 tests, listed below:
[ FAILED   ] CollectionTest.AlwaysFail
[ FAILED   ] CollectionTest.1_negative

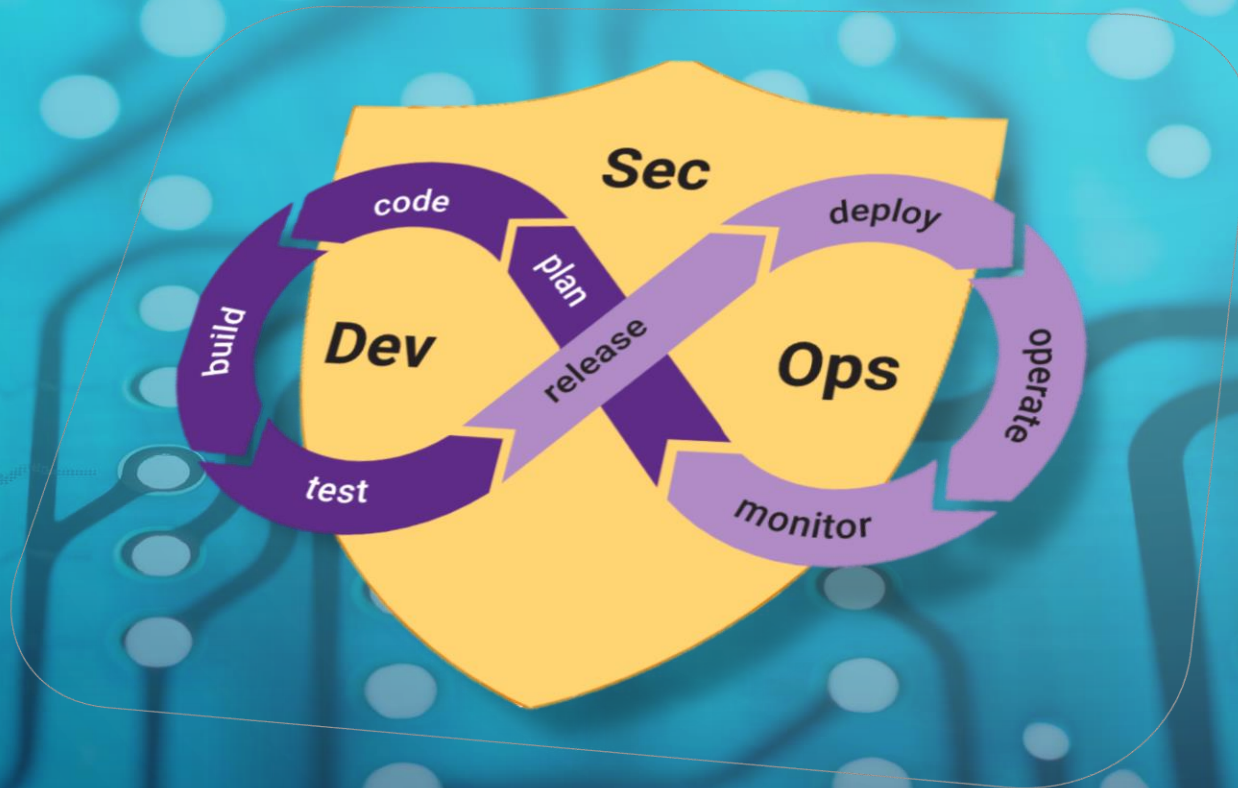
```

UNIT TESTING

- The unit tests will be used with the Google unit testing framework.

DEV-SEC-OPS

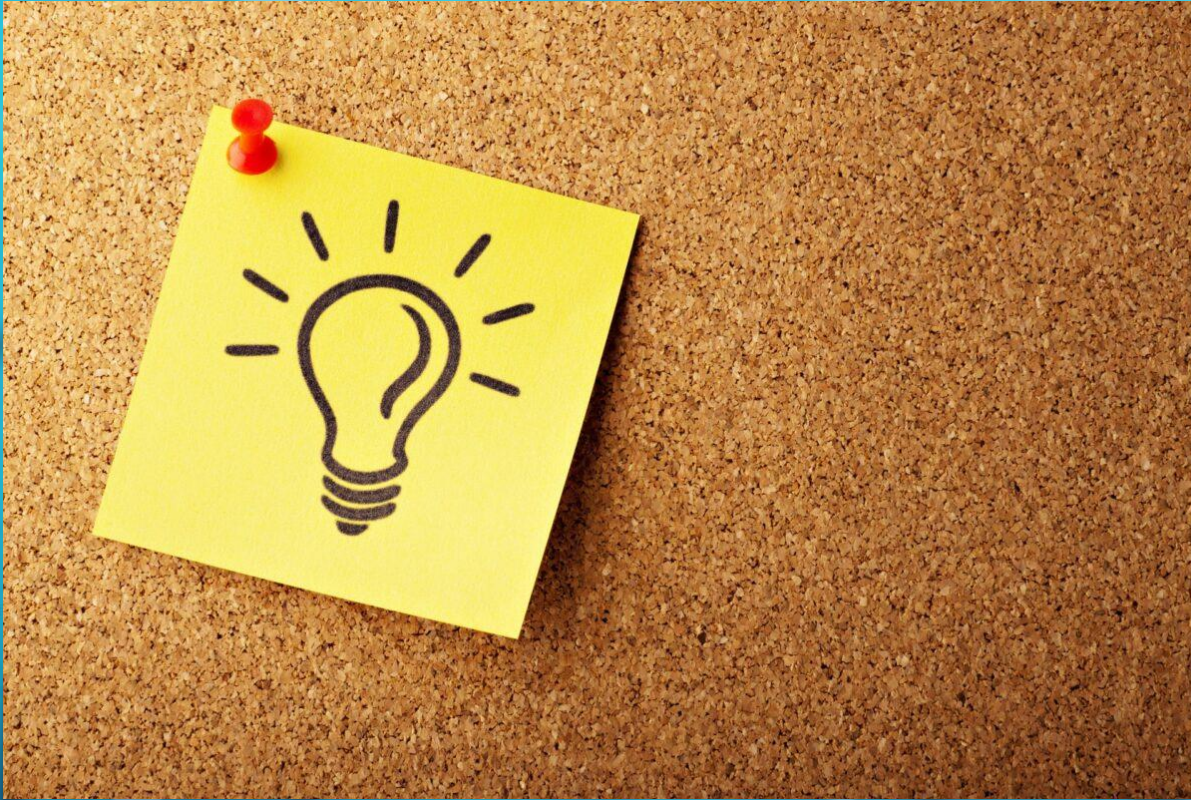
- Static application security testing.
- Software composition analysis.
- Interactive application security testing.
- Dynamic application security testing.





RISKS AND BENEFITS





RECOMMENDATIONS



CONCLUSION

