Compiler Flags for Security

- Requirement:
  - Intel has a requirement that all software released by Intel to Open Source must comply with Intel's Security Dev Lifecycle (SDL)
- The Intel team is responsible for all SDL work
  - But is very grateful for past community contributions to the effort!
- There are just a few items remaining to be completed that may impact the community
  - Security validation - this will be done by Intel and any issues found will be reported to the Security team
  - Compiler Flags for security

GCC Flags Required for C/C++ code

Stack execution protection:	LDFLAGS="-z noexecstack"
Data relocation and protection (RELRO):	LDLFAGS="-z relro -z now"
Stack-based Buffer Overrun Detection:	CFLAGS=”-fstack-protector-strong”if using GCC 4.9 or newer, otherwise CFLAGS="-fstack-protector"
Position Independent Execution (PIE)	CFLAGS="-fPIE -fPIC" LDFLAGS="-pie" (PIE for executables only)
Fortify source:	CFLAGS="-O2 -D_FORTIFY_SOURCE=2"
Format string vulnerabilities:	CFLAGS="-Wformat -Wformat-security"

Plan / Resourcing

- Code changes for the -W changes have been submitted
  - https://review.openstack.org/#/c/629329/ (merged)
  - https://review.openstack.org/#/c/629331/ (merged)
  - https://review.openstack.org/#/c/629332/ (review pending)
- Micro benchmarking is in progress for flags with possible performance impact
  - These benchmarks will exaggerate the performance impact of these flags
- We currently have little capability for system performance testing and changing compiler flags is likely to have little/no measurable impact on system or application performance ...
- We will verify that by performance testing actual STX services
- We'd like some guidance as to how to craft a realistic benchmark from the services:
  - Which code paths are critical? How can we exercise them in isolation from the system?