Open Network User Group

- Forum started by IT executives (https://opennetworkingusergroup.com)
  - Founded by Nick Lippis and others
- User driven community
- Multiple Working groups

ONUG S-DSS WG

- Software-Defined Security Services working group
  - Security requirements in hybrid (private & public) clouds
- Chairs
  - Rakesh Kumar, Fred Lima, Scott Bradner, Nick Lippis
- Whitepaper
  - https://opennetworkingusergroup.com/software-defined-security-services-white-paper-download/
S-DSS WG – Proposed Work

- Build user requirements
  - Develop user requirements based on targeted use-cases

- Publish specifications
  - Update Software-Defined Security Services Framework document
    - Detailed user requirements
    - Architectural framework for developmental guidelines

- Data models and API definitions
  - Work with standard bodies such as IETF/I2NSF to model user requirements

- Open source efforts
  - Encourage open source development efforts based on ONUG work

- Develop vendor certification program
  - Based on ONUG and IETF/I2NSF specification work
Targeted use-cases

- Apply and bind policies to workloads (use-case #8)
  - Enforced as close to workload as possible by security controller based on security fabric capabilities
    - Physical server, virtual machines, containers, services/micro-services
  - Policies move with workload
    - Between hybrid cloud
- Portable security policies (use-case #9)
  - Policies remain same no matter whether workloads are deployed in private or public cloud
- Operational requirements for workloads (use-case #4)
  - Ability to execute workloads in secure environment (confidentiality, integrity and availability)
  - Ability to define the security posture of security control and management components

Translate use-cases to detailed user requirements

- Use-cases not granular enough for specification work to be done by WG
- Use-cases not granular enough for measuring any vendor compliance

User requirements classification

- Cyber threat management policies (use-case #4)
  - Protection against Botnets, Malware, DDoS and other external attacks
- Business security posture policies (use-case #8 & #9)
  - Workload, Data and access policies
  - Regulation and Compliance policies (PCI-DSS, HIPPA etc.)
S-DSS Framework Document – Architecture goals

- Protect against vendor and technology lock-ins (Portable policies)
  - Decouple policy definition from enforcement
    - Define policies based on abstraction such as user-intent or user-construct (a.k.a user-intent policy)

- Consistent policy enforcement
  - A workload policy remains active while workload moves across hybrid cloud
    - Must happen without manual intervention

- Security function flexibility
  - Must be able to use a wide variety of security enforcement points
    - Networking elements (routers, switches), firewalls
    - Hypervisor-based switches, virtual networks (SDN controller orchestrated), security service chains
    - Workloads running on bare-metal servers, virtual machines, containers
    - Public clouds (AWS, Azure)
S-DSS Framework Document – Architectural framework

- **Security controller**
  - A policy compiler or engine
  - Breaks high-level (user-intent) policy into low-level (security function) policy
  - Hides network and security design complexity from user

- **Security Controller – User interface**
  - A data model driven API interface
  - Portable across vendors and hybrid deployments
  - Allows to express policy in high-level abstraction

- **Security Controller – Security Function interface**
  - A data model driven API interface
  - Technology and vendor implementation independent
  - Flexibility to choose security functions with a goal of supporting large-scale and dynamic changes

- **Policy Enforcement – Everywhere**
  - Security control and management components
  - Network Devices, Appliances, and Services (physical or virtual)
  - Native Cloud-specific Security Controls
  - SDN - Virtual switch (e.g., OVS)
  - Workloads – physical, virtual, containers