Software Defined Perimeter (SDP)

Matt Zekauskas, matt@internet2.edu
from slides by Steve Wallace, Indiana University and
Florence Hudson, Internet2
cino@internet2.edu

SDP is a project of the Cloud Security Alliance

CSA cloud security alliance®

Software Defined Perimeter (SDP)

- Architectural model for securing network-connected infrastructure
- Complementary to SDN (but not SDN)
- Cloud Security Alliance standard ver 1.0
- Used by major corporations today
- Presentation depicts the prototypical client-to-server implementation
  - SDP defines the client as the Initiating Host (IH), and the server the Accepting Host (AH)
- Can be leveraged to improve the security and accountability of science workflows (and possibly activate SDN-controlled paths)
- Version 1.0 of the protocol can be found here: https://downloads.cloudsecurityalliance.org/initiatives/sdp/SDP_Specification_1.0.pdf
- Presentation to Internet2 End-to-End Trust and Security Working Group: https://spaces.internet2.edu/display/CWG/Webinars (September 1, 2015)

Software Defined Perimeter (SDP) Attributes

- Controller-based authentication and authorization (can use Shib, LDAP, etc.)
- Client to Server connectivity only provisioned for authorized clients
- Client and Server mutually authenticate (bi-directional TLS, a.k.a. Mutual TLS)
- Operates over public IP networks, no special network service required
- Server remains “dark” to non-authenticated connection attempts
- Based on open standards:
  - TCP
  - TLS
  - public key encryption

Software Defined Perimeter - Controller

- SDP controller
  - authenticates and authorizes the client (can be user credentials via ADS, Shib, etc.)
  - enforces authorization policy
  - configures the server to accept a connection from the authorized client
  - shares a common secret (authentication token) with the client and server
  - can store log files that include the record of connections between clients and servers
  - should be protected by conventional defense in depth (firewall, IDS, etc.)
  - can be logically central, however implementations can be resilient and diverse (i.e., no single point of failure)

Software Defined Perimeter - Client

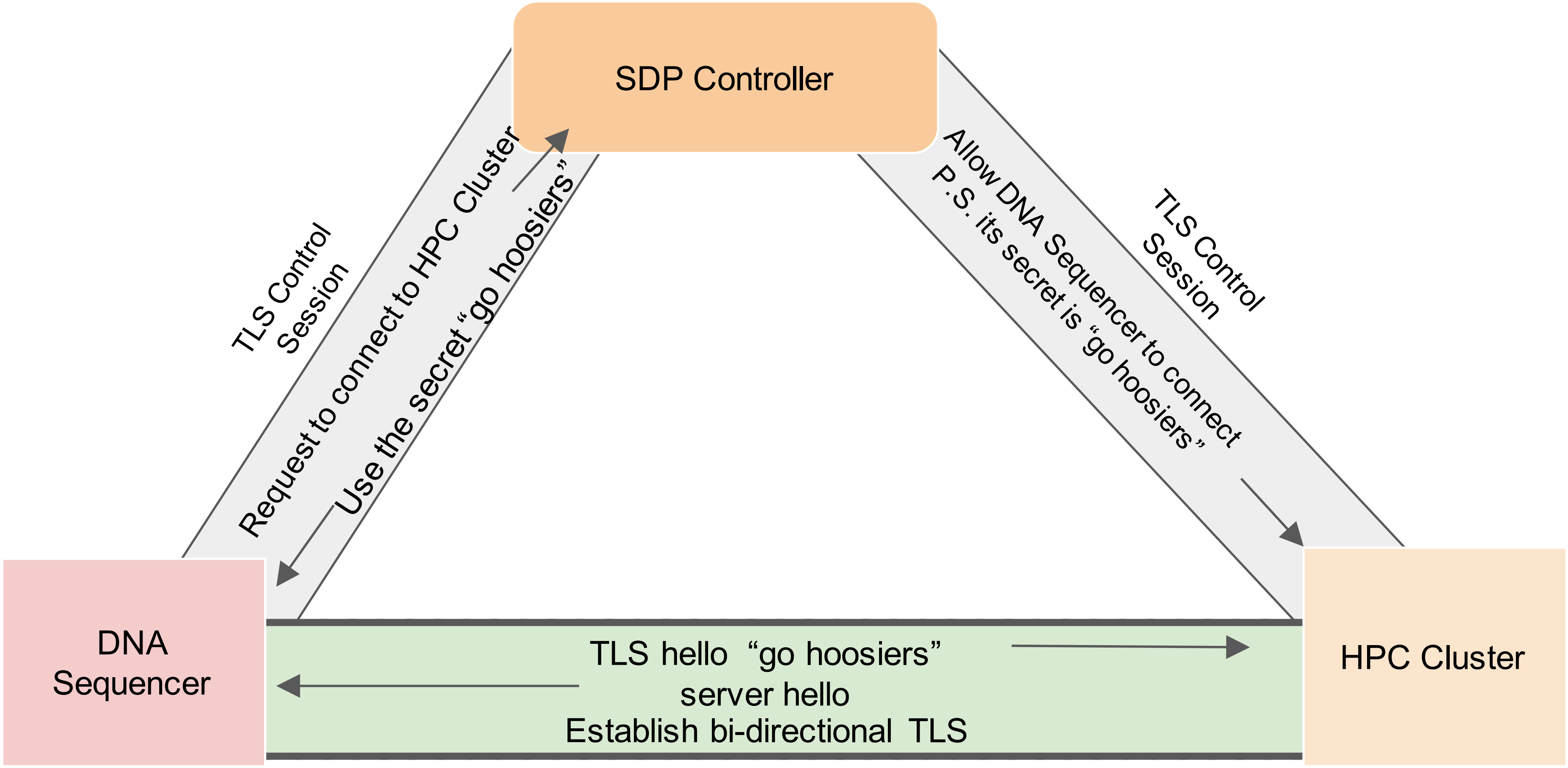
(aka “*Initiating Host*”)

- SDP Client
  - at startup, establishes TLS connection to controller for its control channel (used for client authentication and client requests to connect to servers)
  - client and controller can verify authenticity via bi-directional TLS
  - authenticates to the controller (can be Shib, ADS, etc.)
  - sends server connection requests to controller (e.g., can DNA sequencer connect to HPC cluster)
  - receives a one-time password (i.e., token) with which to establish server connection
  - client and server can verify authenticity via bi-directional TLS

Software Defined Perimeter - Server

(aka “*Accepting Host”*)

- SDP Server
  - at startup, establishes TLS connection to controller for its control channel
  - server and controller can verify authenticity via bi-directional TLS
  - is “dark”
    - doesn’t respond to unauthorized connection requests
    - drops connections unless they contain the proper token
  - receives permission from controller to allow a client connection, as well as the client’s one-time password (i.e., token)
  - server and client can verify authenticity via bi-directional TLS


SDP Controller
TLS Control Session
Request to connect to HPC Cluster
Use the secret "go hoosiers"
Allow DNA Sequencer to connect
P.S. its secret is "go hoosiers"
TLS Control Session
DNA Sequencer
TLS hello "go hoosiers"
server hello
Establish bi-directional TLS
HPC Cluster

Chip Design Company

Business Objective: Accelerate chip design process by leveraging public clouds

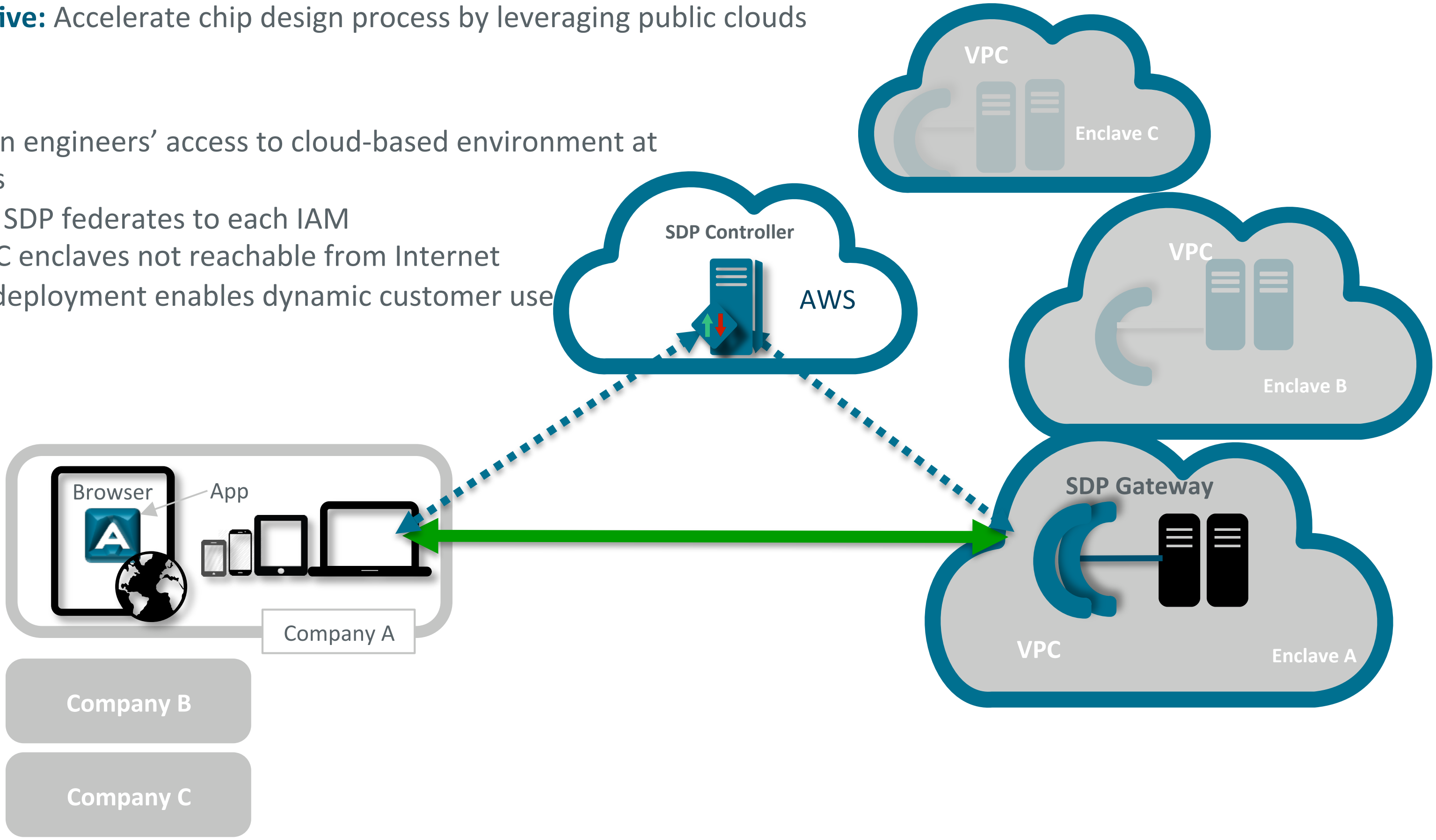
SDP Solution:

✓ Secures design engineers' access to cloud-based environment at customer sites

✓ Single tenant SDP federates to each IAM

✓ Customer VPC enclaves not reachable from Internet

✓ Flexible SDP deployment enables dynamic customer use



From Junaid Islam presentation, 2015-09-01

Defeating Attacks on the Extended Enterprise

- Server exploitation: constant attacks

  ~~Misconfigurations~~
  ~~Vulnerabilities~~
  ~~Injections~~
  ~~Denial of Service~~

Server Isolation
SPA, Dynamic FW

- Credential theft: ⅔ of Verizon DBIR

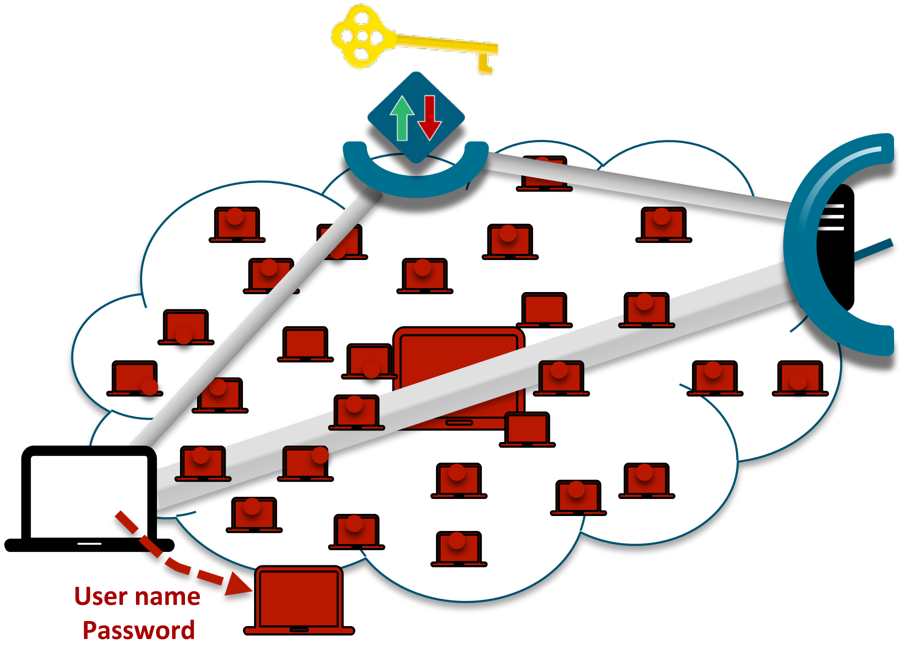
  ~~Phishing~~
  ~~Keyloggers~~
  ~~Brute force~~

Transparent MFA
mTLS, Fingerprint

- Connection hijacking: stealthiest

  ~~Man-in-the-Middle~~
  ~~Certificate forgery~~
  ~~DNS poisoning~~

Encryption,
Pinned Certs,
No DNS



From Junaid Islam presentation, 2015-09-01

Potential role for the SDP Controller to increase science workflow assurance

- The SDP Controller can assume additional roles within the science workflow, these roles are outside the strict scope of SDP, but such extensions can compliment and leverage the SDP Controller function.
- Beyond its core function of provisioning secure communications, the controller can also orchestrate the science workflow.
- Its orchestration can require specific workflow sequences e.g.:
  - after moving DNA sequence data to archive facility, the only permitted next step is completing its metadata
  - upon receiving HIPAA PHI, the next step must be de-identification
  - after initial data analysis, results must be archived before repeat analysis
- The controller can record the progress through workflows, providing an auditable record
- These are NOT part of SDP, but the controller and the science workflow tools could be extended to provide such capabilities

Opportunity: Working on Next Version of Spec

- HPC Use case
  - also extensions for HPC or data-intensive science workflows
- Scalability
- Integrate with Software Defined Networking
  - e.g. ephemeral paths for data flows
- Looking for academic contributors

Want to improve SDP? It's an open spec….

cloud security alliance®

Check out CSA Membership:
https://cloudsecurityalliance.org/membership/

hint, for individuals it's complimentary

Junaid Islam
Co-Chair SDP Workgroup
Cloud Security Alliance

Junaid is actively seeking new collaborators!

(jislam@vidder.com)

Thank you!