NetSTAR Global, Inc.

Since 2001

The Global Leader in
URL, IP and Web App Categorization
& Intelligence for OEMs.

Making OpenSSL Approachable

Interactive Labs, Automation, and AI

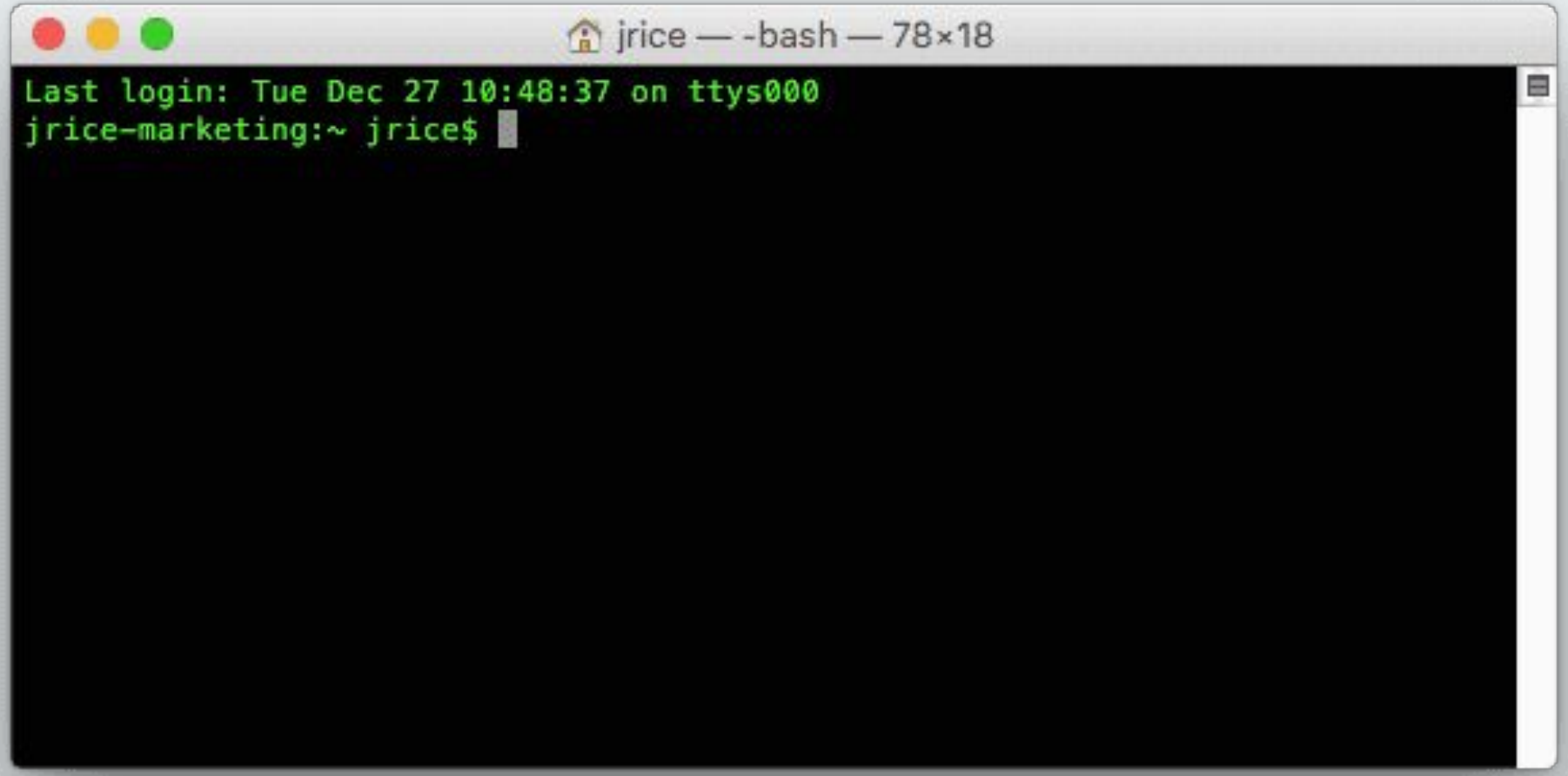
Aaron Escamilla
SMLSE
NetSTAR Global, Inc.

- 275+ Global OEM Partners
- 1.8 billion global endpoints
- Many partners have used NetSTAR for 10+ years
- Web and SaaS categorization and Threat Intelligence technologies for the OEM, embedded, and service provider/telco markets

NetSTAR Suite of Solutions:

INCOMPASS	**URL and IP Categorization & Risk Intelligence**		The Leading URL & IP Address Categorization & Threat Detection Solutions for OEMs
WEBAPP	**Web Application Categorization & Intelligence**		Category & risk Intelligence to monitor and manage access to cloud services.
INSITE	**Real-Time Threat Detection Feed**		Actionable URL & IP Threat Intelligence to Enhance Cybersecurity Solutions
IP/COMPASS	**IP Address Categorization with Geolocation & Metadata**		Filter Network Traffic Associated with single and multi-domain IP Addresses
ADVANTAGE	**Internet Categorization Solution for AdTech**		Website category data and risk intelligence for ad tech solutions
LOG COMPASS	**Log Management Solution**		Log reporting and data management solution

So why don’t more of us know how to use it?



It’s easy right?

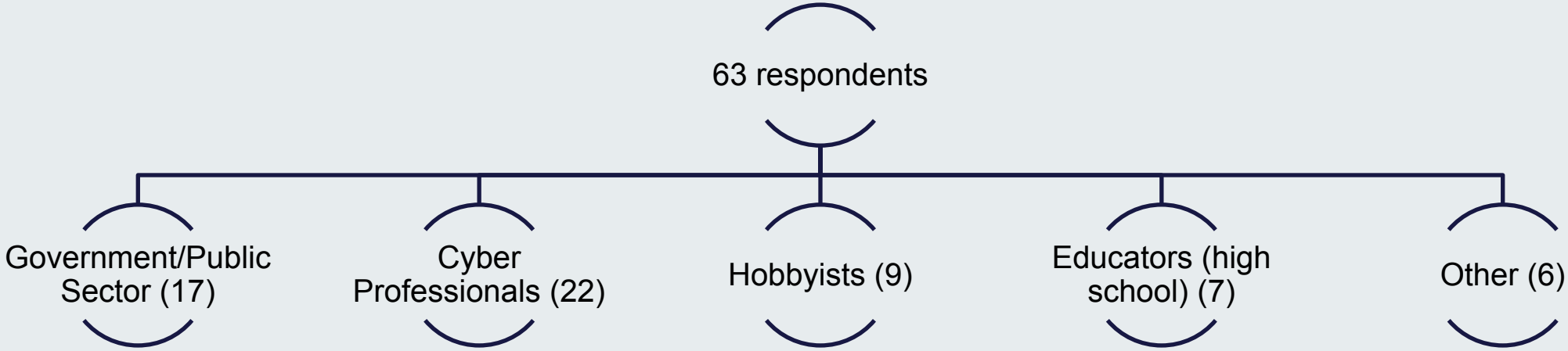
openssl rsa -in privkey.pem -pubout > key.pub
– Extracts the public key from a private RSA key.
openssl x509 -pubkey -noout -in cert.pem > pubkey.pem
– Extracts the public key from an X.509 certificate.
openssl rsautl -encrypt -in plaintext.txt -out encrypted.txt -pubin -inkey pubkey.pem
– Encrypts a file using an RSA public key.
openssl s_client -showcerts -host example.com -port 443 </dev/null
– Connects to a server over TLS and displays its certificate chain.
openssl x509 -x509toreq -in MYCRT.crt -out CSR.csr -signkey privateKey.key
– Converts a certificate into a certificate signing request (CSR).
openssl rsautl -decrypt -in encrypted.txt -out plaintext.txt -inkey privkey.pem
– Decrypts a file using an RSA private key.
openssl rsa -in server.pem -out newserver.pem
– Rewrites or re-encodes a private RSA key into a new PEM file.

Right???

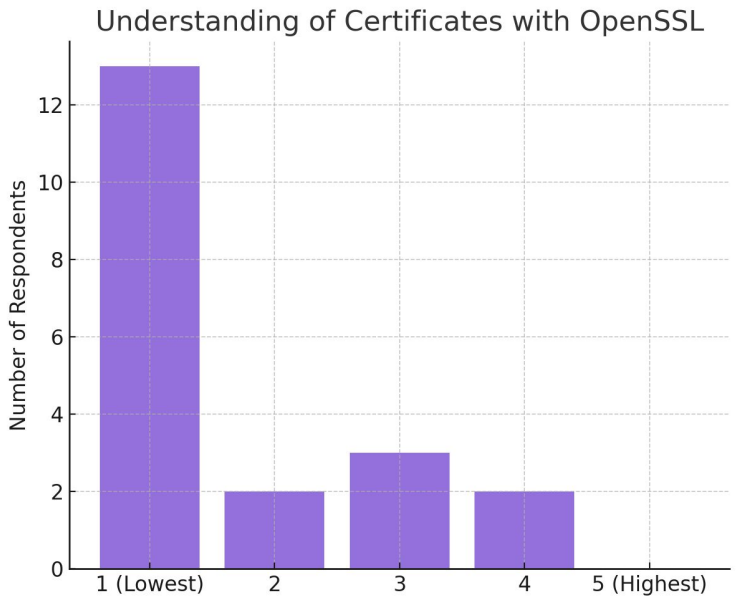
```
openssl req -new -newkey ec:<(openssl ecparam -name prime256v1) -nodes \
  -keyout site.key -out site.csr -sha256 \
  -subj "/C=US/ST=Arizona/O=Example Inc/CN=example.com" \
  -addext "subjectAltName=DNS:example.com,DNS:api.example.com,IP:10.0.0.5" \
  -addext "keyUsage=digitalSignature,keyEncipherment" \
  -addext "extendedKeyUsage=serverAuth,clientAuth" \
  -addext "tlsfeature=status_request"
```

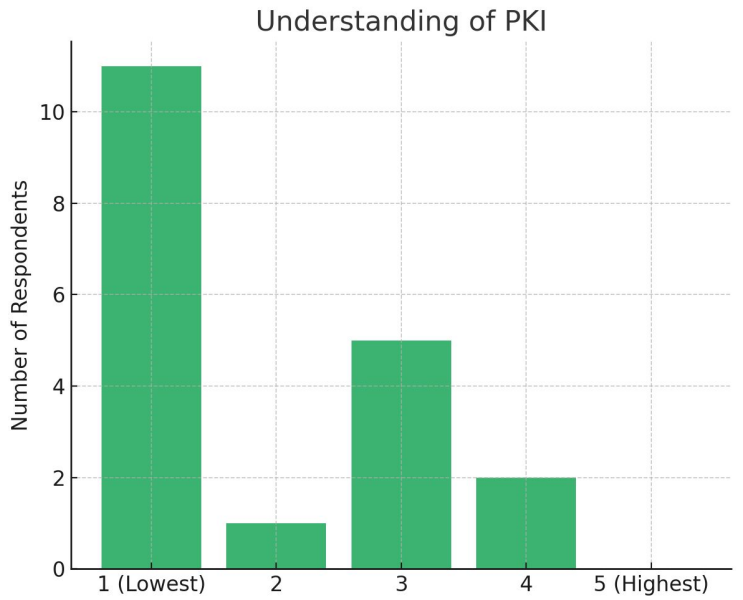
Sample Learning Group

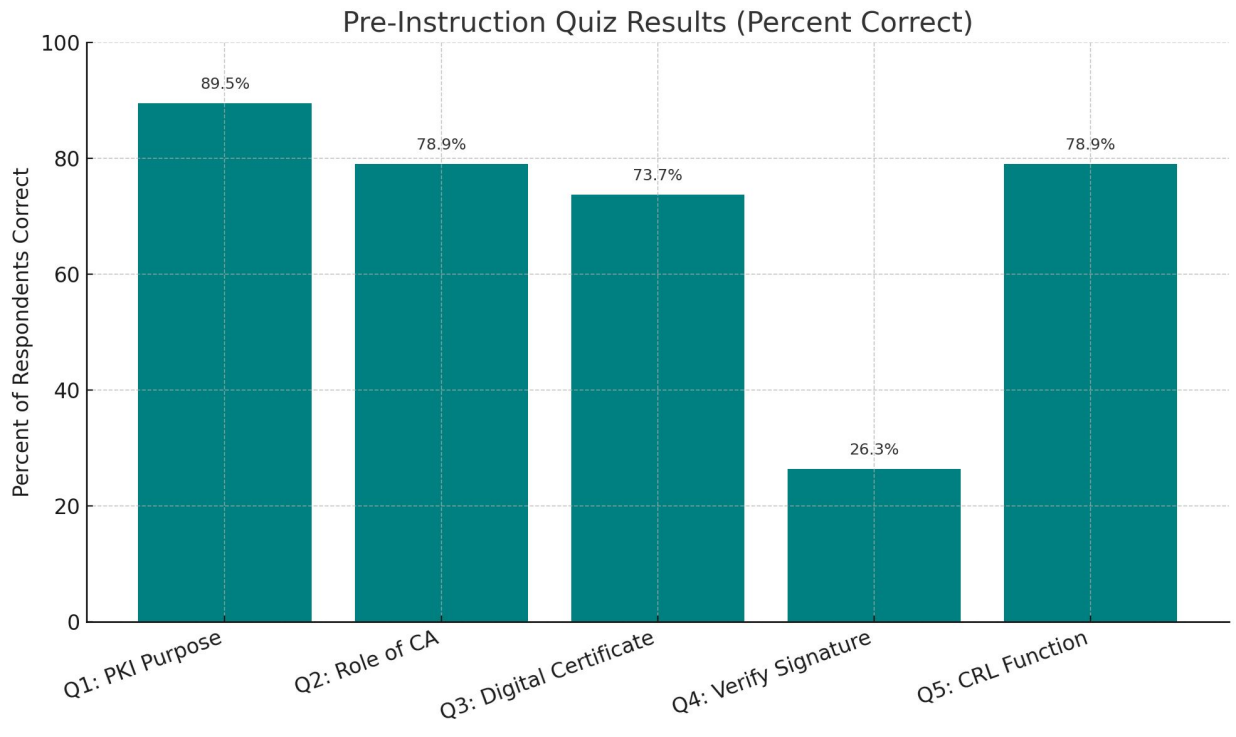
How can we improve awareness of certificate security and tools like OpenSSL?



We established a baseline of where each learner was.







Questions:

What is the primary purpose of a Public Key Infrastructure (PKI)?

What role does a Certificate Authority (CA) play in PKI?

Which of the following best describes a digital certificate?

What is typically used to verify a digital signature in PKI?

What is the function of a Certificate Revocation List (CRL)?

Solving Common Learning Gaps

1. Active Learning

- Learners use a simulated terminal to run real OpenSSL commands in a safe, hands-on environment—making mistakes and seeing immediate outputs without risk to production systems.
- Tasks like “Compare key and CSR modulus” or “Build a certificate chain” are performed directly by the learner, not just passively observed.

2. Guided Discovery

- Step-by-step labs break down tasks (“Create a CSR,” “Verify pairing,” “Generate CRL”) and walk the user through them, providing hints and contextual help.
- Visual cues and prompts help learners understand the “why” and “how” of each step (not just rote command syntax).

3. Immediate Feedback

- The platform offers instant feedback on command correctness, concept quizzes, and visual indicators for errors or success (e.g., “Your CSR matches your key!” or “Chain is broken, try again”).
- This encourages faster learning and stronger retention compared to static, document-based tutorials.

4. Retrieval Practice & Spaced Repetition

- Quizzes and checkpoints reinforce key ideas like “What is a modulus?” “How do you tell if a CRL update succeeded?” at intervals throughout the lab modules.
- Users revisit concepts just as they’re about to forget them, boosting retention.

5. Visual Learning

- Abstract procedures (like certificate chains, CRL lifecycle, modulus matching) are illustrated with diagrams, process flows, and decision trees.
- Visual mapping connects CLI steps to underlying cryptographic structures.

6. Error-Driven Learning

- Learners encounter realistic failure scenarios: mismatched keys, incomplete chains, missing CRLs—then are guided through troubleshooting using OpenSSL diagnostics.
- This builds confidence to handle real-world issues.

7. Scaffolded Progression

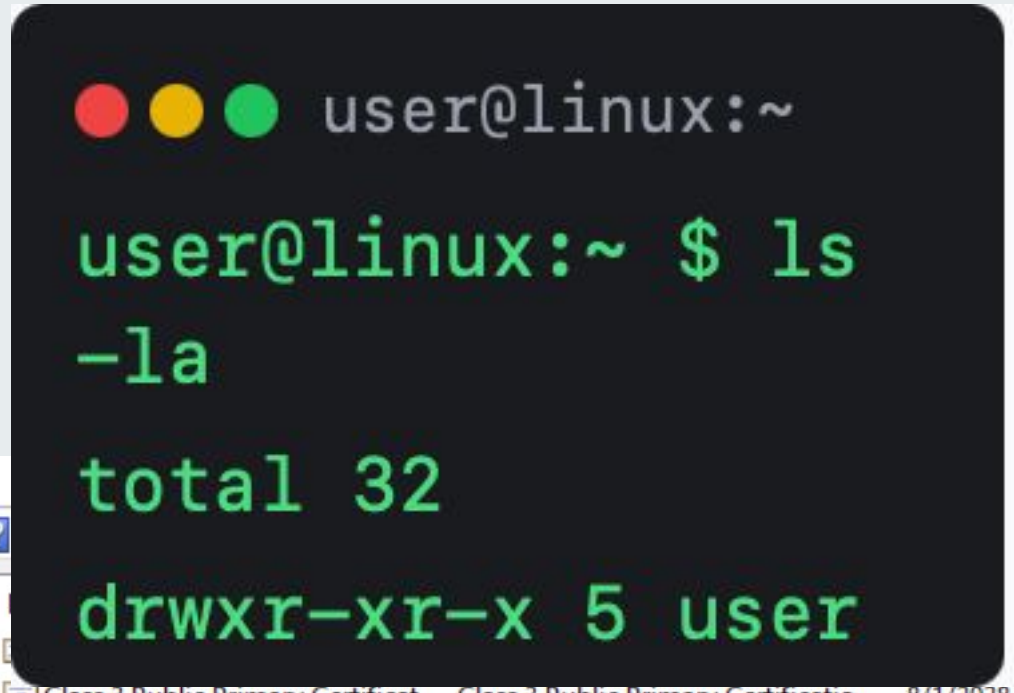
- Labs begin with simple tasks and build up in complexity—learners move from “create a key” to “link chain to CA” to “fix broken CRLs,” building expertise incrementally.

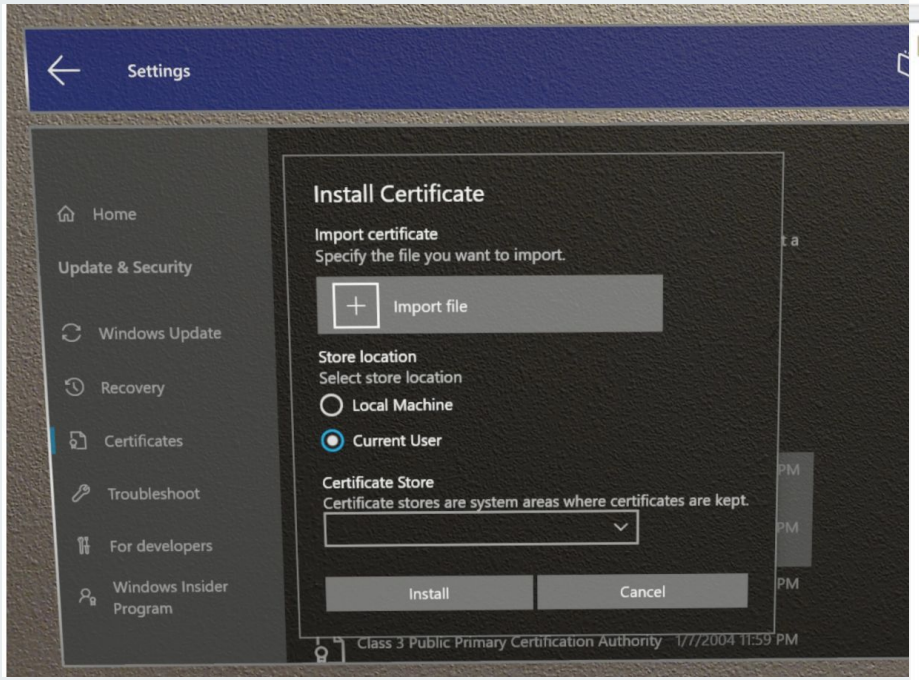
Solving Common Learning Gaps

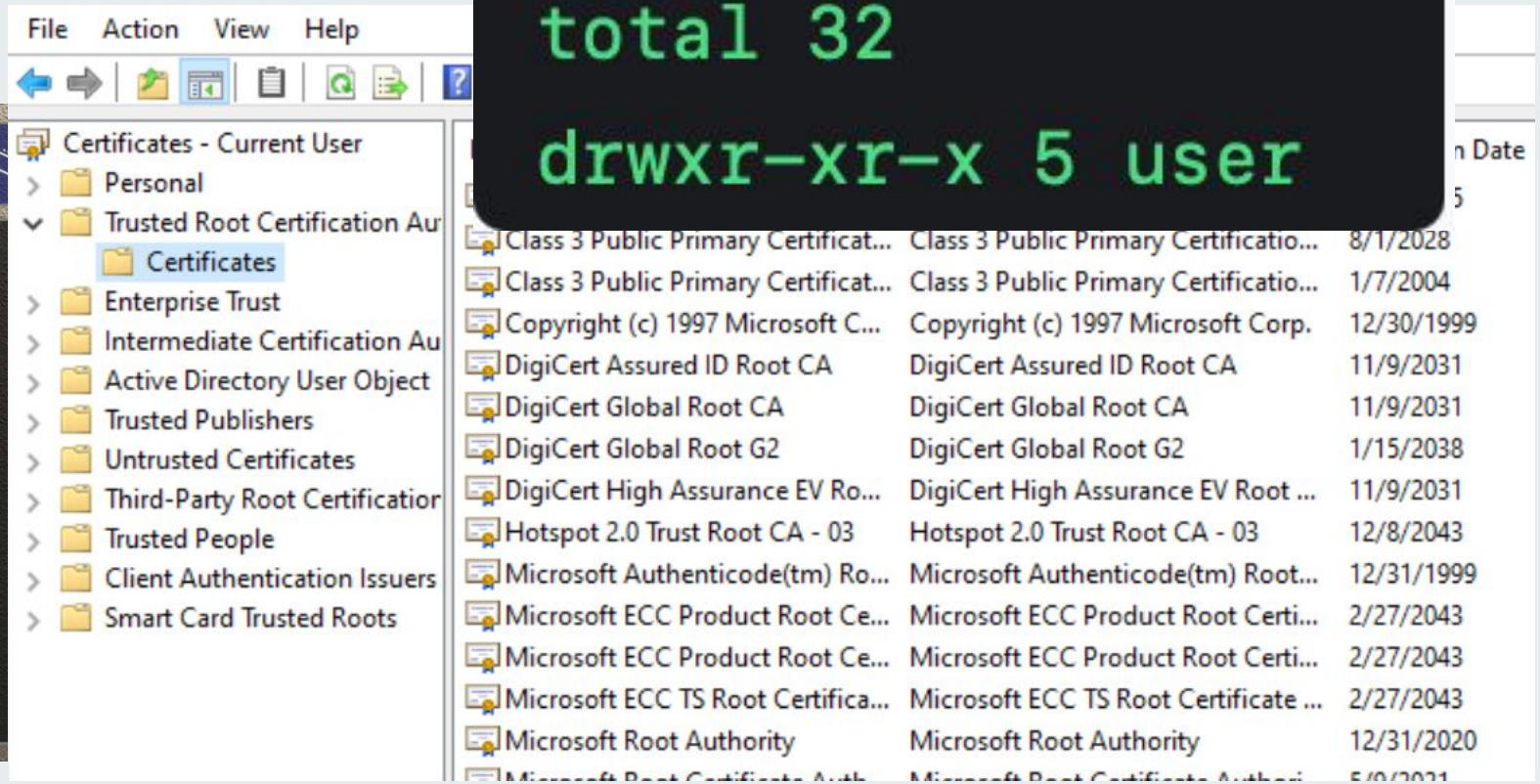
```
# 1. Generate a FIPS-compliant private key (cryptographic compliance, key management)

openssl genpkey -algorithm RSA -out privkey.pem -pkeyopt rsa_keygen_bits:2048

# 2. Inspect and verify the key format (PEM vs DER)

openssl pkey -in privkey.pem -text -noout

# 3. Create a valid CSR with subject and extensions (subject customization, CSR creation)

openssl req -new -key privkey.pem -out request.csr \
  -subj "/C=US/ST=California/L=San Francisco/O=ExampleCorp/CN=www.example.com/emailAddress=admin@example.com" \
  -addext "subjectAltName=DNS:www.example.com,DNS:example.com" \
  -config openssl.cnf


……..
……..
```

- Understanding **command syntax** and **flag usage**
- Properly generating and managing private keys
- **Creating valid** Certificate Signing Requests (CSRs)
- **Troubleshooting** key/CSR mismatches
- Building and **verifying** complete certificate **chains**
- Reading and **interpreting** OpenSSL **error messages**
- Differentiating between **certificate formats** (PEM, DER, etc.)
- Managing and **updating** Certificate Revocation Lists (**CRLs**)
- Using OpenSSL to **inspect** and parse **certificates**
- Working with subject string **customization** and extensions
- Automating routine tasks without introducing errors
- Ensuring **cryptographic compliance** (e.g., FIPS 140 support)
- Safely **using** OpenSSL **in production environments**

Which one are you?







Overview of Steps for obtaining and installing an ECA certificate

Step 1: Generate the Private Key

```
openssl genrsa -out private_key.pem 2048
```

Step 2: Verify the File Was Created

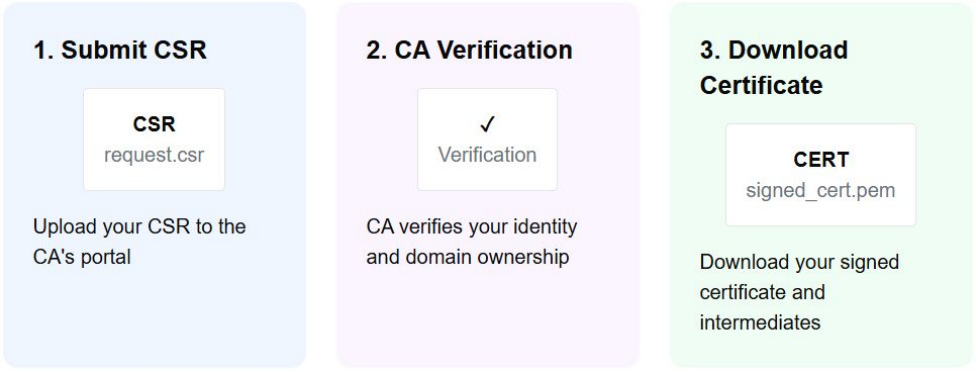
```
ls -l private_key.pem
```

Step 3: Generate the Certificate Signing Request

```
openssl req -new -key private_key.pem -out request.csr
```

Step 4: Submit CSR to the ECA

Step 5: Receive Certificate Files



Step 6: Combine Certificate Chain (If Required)

1. **Step 1: Build the chain**

```
cat signed_cert.pem intermediate.pem root.pem > full_chain.pem
```

This concatenates the files in the correct order to create a chain file.

2. **Step 2: Verify the chain**

```
openssl verify -CAfile full_chain.pem signed_cert.pem
```

This checks that the certificate chain is valid and trusted.

3. **Step 3: Inspect the chain**

```
openssl x509 -in full_chain.pem -text -noout
```

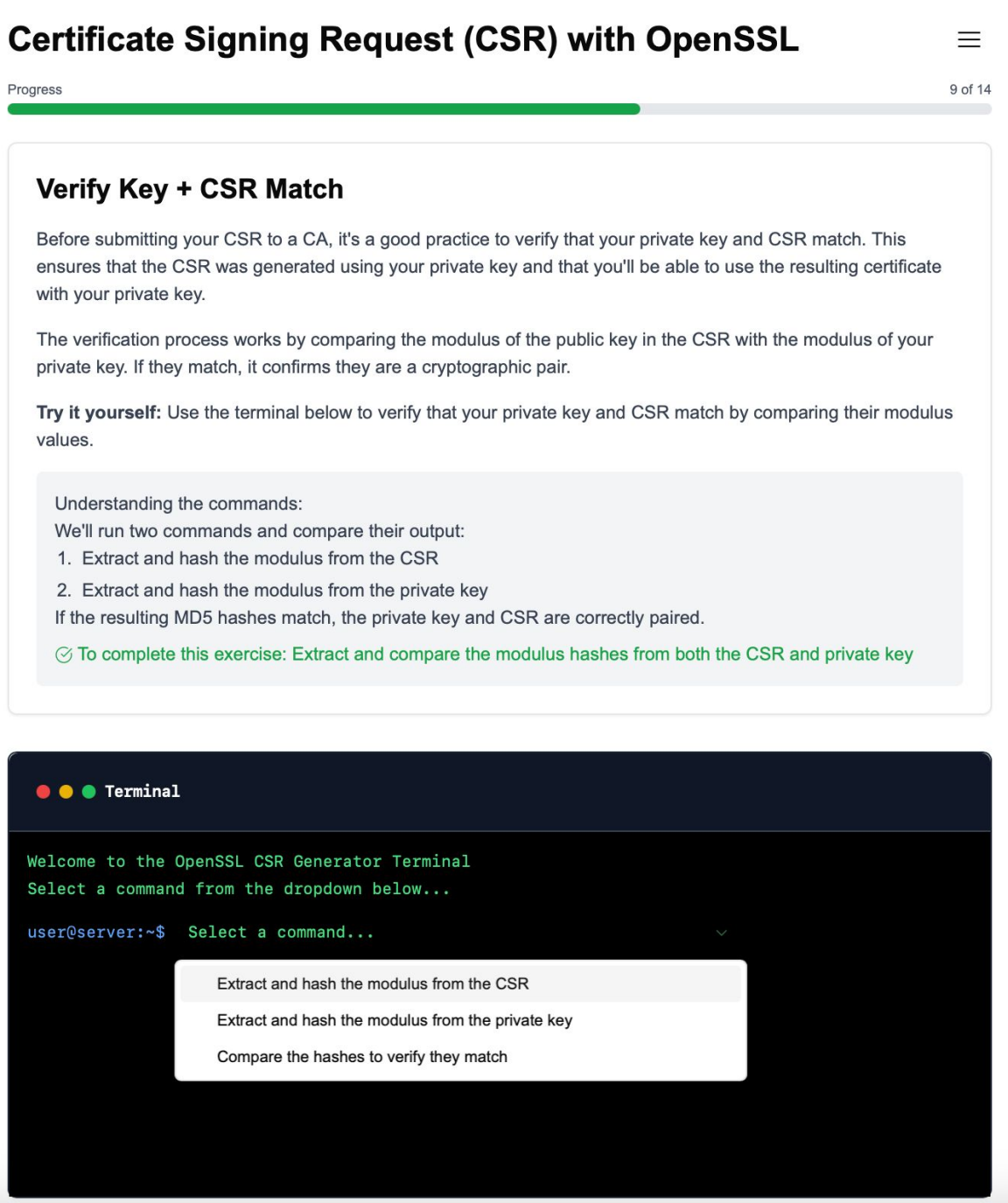
This displays the details of the certificate in the chain file.

Step 7: Install the Certificate on Your Server

Introducing the Learning Platform

- Browser-based, interactive labs for OpenSSL fundamentals
- Simulated terminal: run OpenSSL commands safely
- Guided exercises, built-in quizzes, and instant feedback

https://openssl-tutorial-uofa.netlify.app



Module Impact Analysis

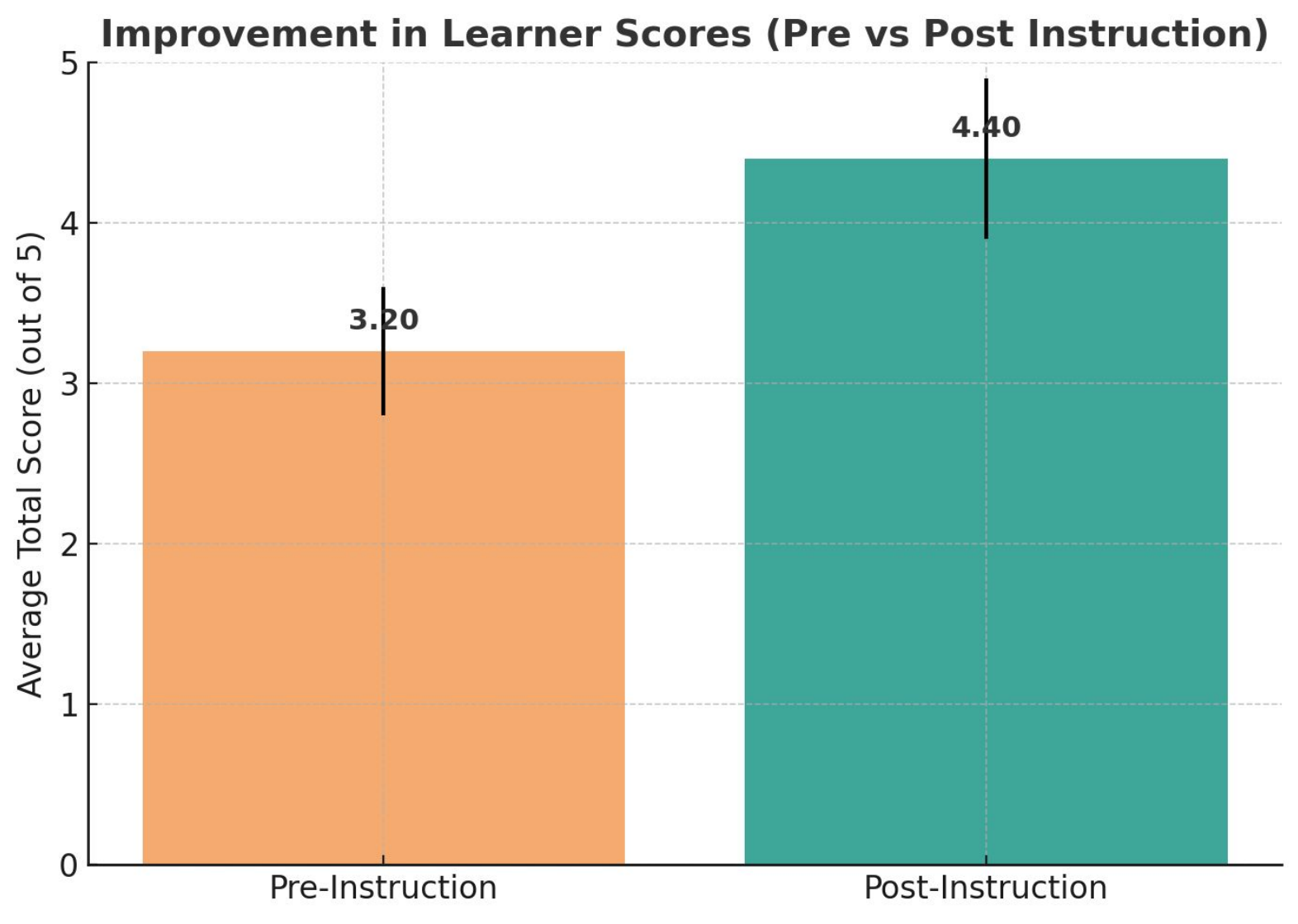
Methodology

- Self-rated knowledge (1–5); scores shrunk toward midpoint when confidence low; more impactful high confidence incorrect
    - Easy beginner questions; most people could get right
- Post-Module Test
    - New, specific questions; slightly more difficult to prevent copying

Statistical Analysis

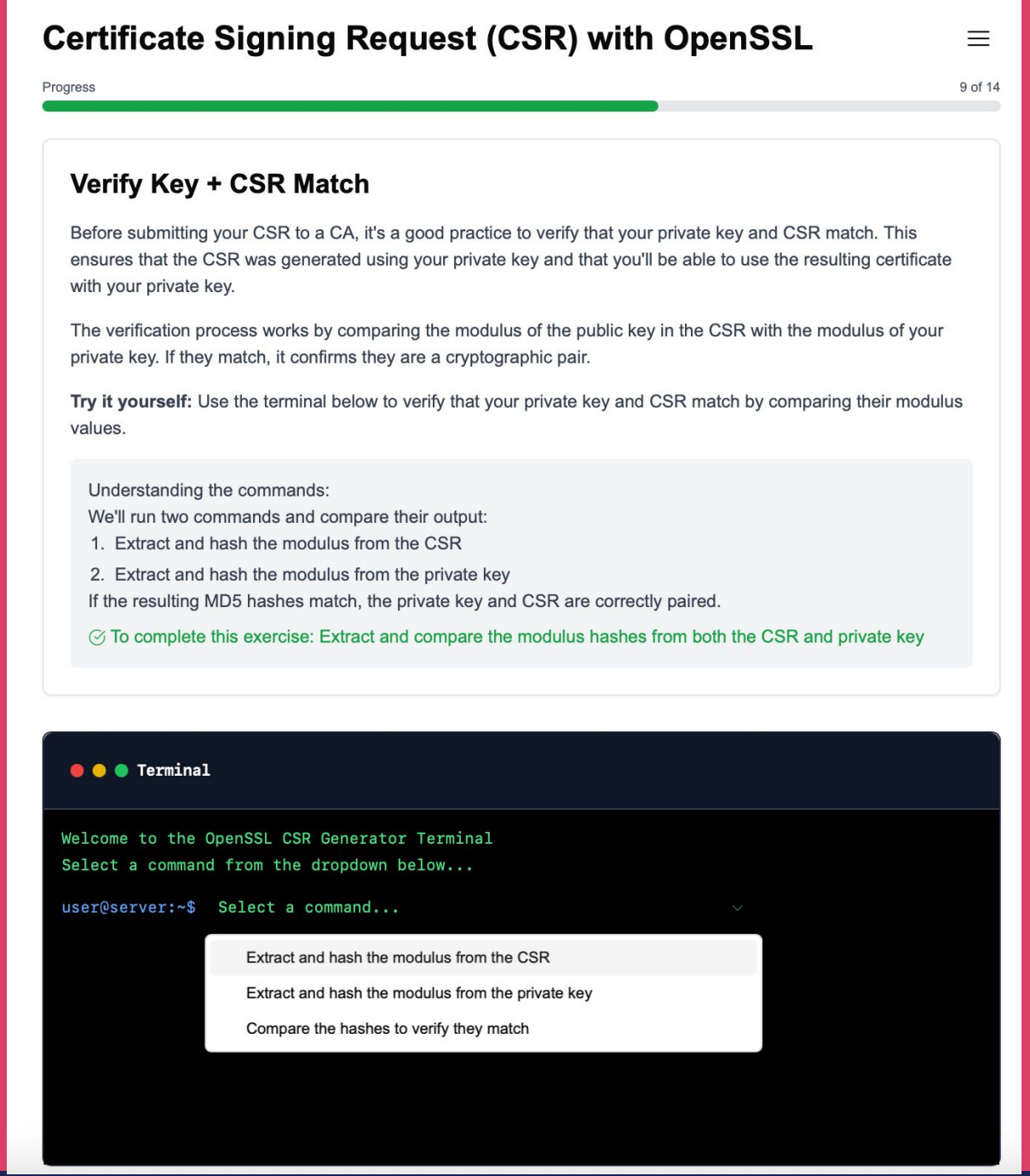
- Hypotheses
    - Ho: $\mu Pre - \mu Post \leq 0$
    - Ha: $\mu Pre - \mu Post > 0$
- P-value: 0.04
- 4% chance of observing this jump if module had no effect
- Cohen’s d: 0.94 (large effect size)

Post Learning-Module Results



Business Value and Enterprise Adoption

 **Audit Readiness & Compliance (FIPS)**	Deploying TAXII servers required validating crypto modules and reconfiguring OpenSSL builds.
 **Navigating Tool Limitations**	Subject string customization needed manual config edits and workarounds.
 **Learning to Automate**	Scripts + schedulers automated CRL updates, reducing downtime risk.
 **Learning Together**	Training improved learner confidence and directly reduced production errors.



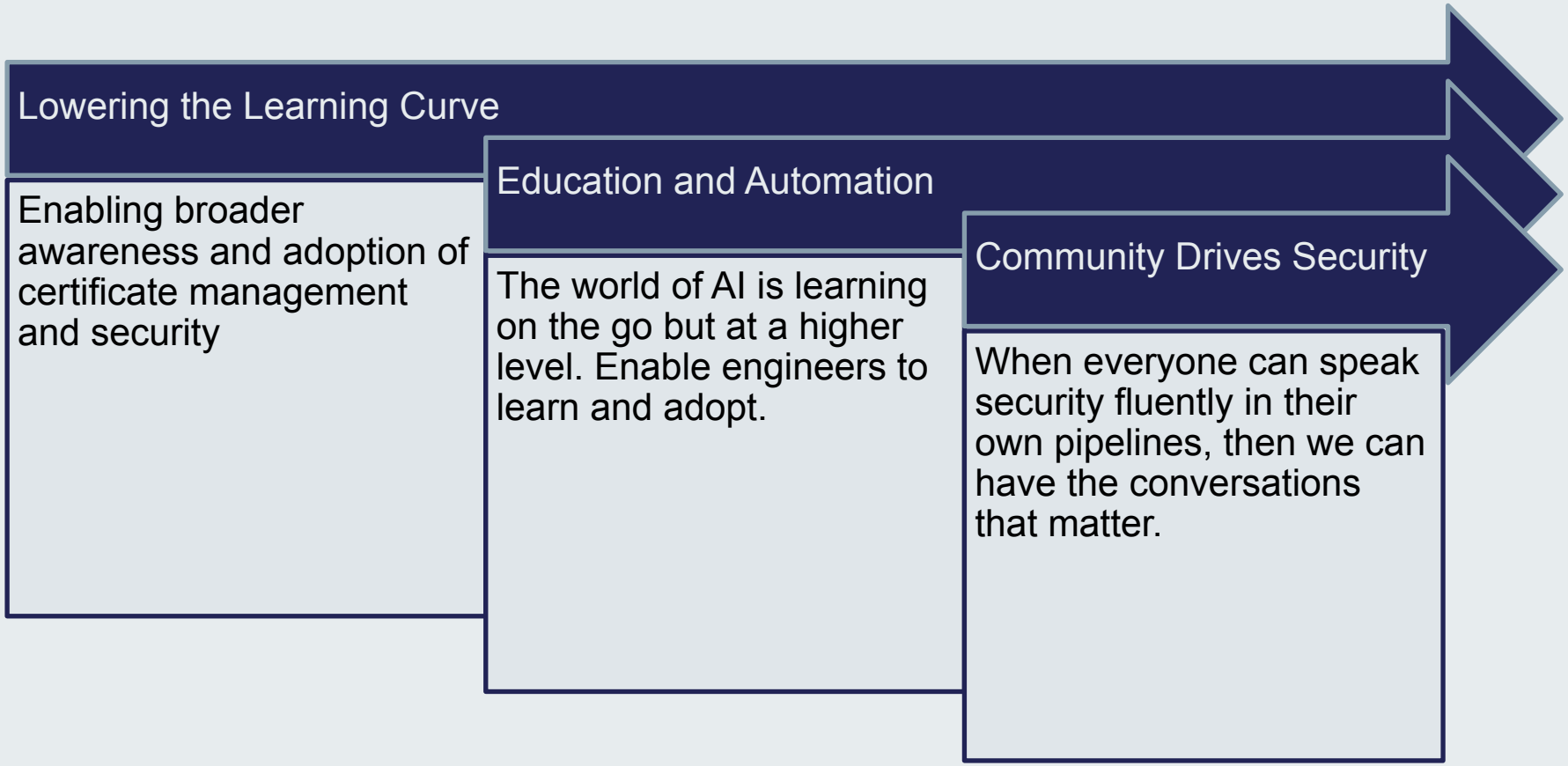
Future Learning Terminals and Automated AI driven Cert Managers

Why not make the terminal all the documentation and support you need?

Planned AI Assistant:

- 1) to support command usage
- 2) to support troubleshooting and chain validation, chain smithing, and self healing
- 3) Explainability! The ability to understand the current security posture and certificate management across your server.

Key Takeaways



Global Fingerprint - OpenSSL and NetSTAR Operations

Dedicated Support Team

Rapid Response Times

24/7 Availability

400 Global Employees

Multi-lingual Support Teams

Email / Telephone Web-API

Thank You