# Fuzzing-based Testing of Certificate Authority

Yunhan Qiao, Yen-Chun Huang, Ming Wei (group 9)

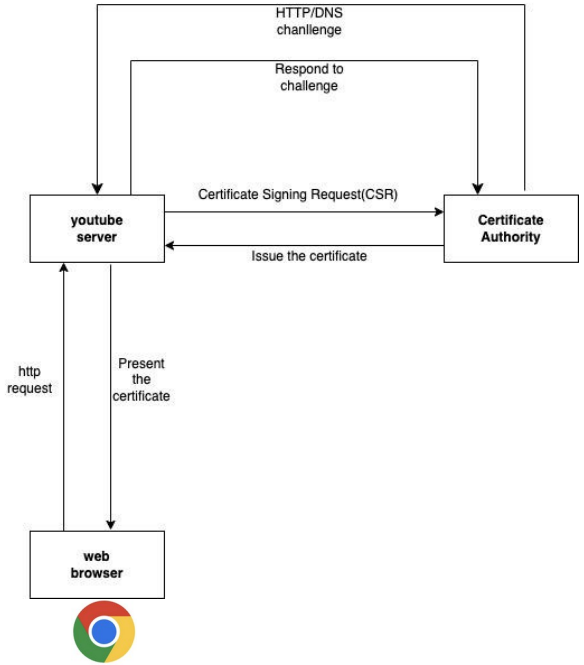
# Certificate Authority (CA)

A certificate authority is a company or organization that acts to validate the identities of entities (such as websites, email addresses, companies, or individual persons) and bind them to cryptographic keys through the issuance of electronic documents known as digital certificates.

*https://www.ssl.com/article/what-is-a-certificate-authority-ca/#:~:text=A%20certificate%20authority%20(CA)%2C,the%20issuance%20of%20electronic%20documents*

# How does it work?

1. Web browser sends http request
2. Server Requests a Certificate Signing Request (CSR)
3. CA Verifies the Server's Identity (verify the ownership of domain)
4. Server Completes the Challenge
5. CA Issues the Certificate
6. Secure Communication Between Server and Client



# Security incident of CA

## 2011 Comodo Security Breach Incident

Comodo, which accounts for 40% of global Internet certificate issuance.

Incident history:

In March 2011, Comodo was attacked by Iranian hackers.

Nine certificates were issued in error, affecting seven domains.

Handling and Impact:

Comodo quickly detected the security breach and revoked the improperly issued certificates.

The intermediate certificates affected over 85,000 domains.

Further impacted an additional 120,000 domains.

## 2011 DigiNotar Security Breach Incident

Incident history:

Fraudulent certificates were issued for several domains by Iranian hackers.

The vulnerability was hidden for two months until Iranian users discovered a fake certificate warning when accessing Google services using Google Chrome.

Impact:

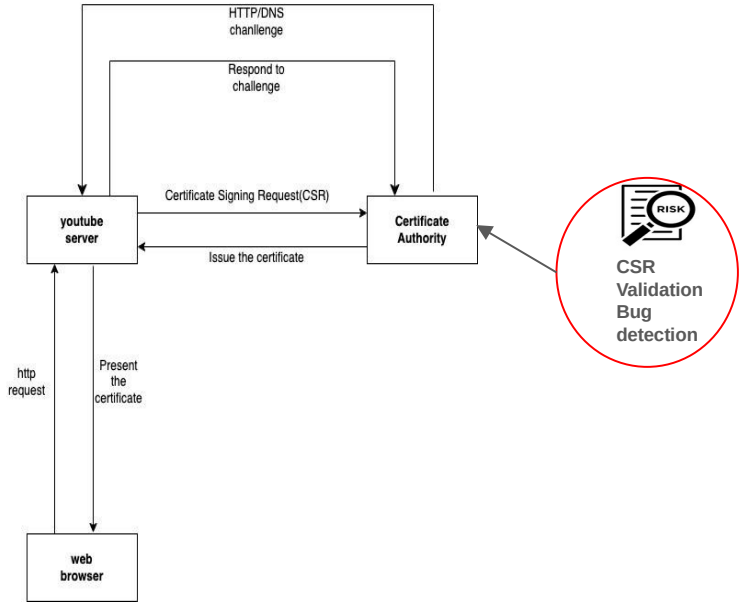
Led to a loss of trust in DigiNotar’s ability to securely issue certificates

The impact of the security breach was so widespread that the browser vendor revoked DigiNotar's certificates, ultimately putting the company out of business.

https://ieeexplore.ieee.org/document/8249081

# Project Idea - CATest

1. Detect CSR validation bugs
2. AFL++
3. Use grammar-based fuzzing approach to generate inputs
4. Run inputs on different CA libraries
5. Check if mutated CSR's certificate matches the original certificate



# Research Questions

1. How effective is CATest at detecting CSR validation bugs?
2. How can custom mutator improve the performance of a fuzzer?
3. How does CATest compare with the state-of-the-art tools?

# Evaluation-Success Criteria

1. Successfully identify potential CSR validation bugs in the CA.
2. Custom mutators can improve the performance of fuzzer.
   - Generate more inputs compared to AFL++ default strategy
1. CATest outperforms the other state-of-the-art tools.
   - less false positives, more true positives

# Planned Timeline

**Implement CATest and detect CSR validation bugs (Yunhan Qiao):**

Task 1: Implement an encoder and decoder tool. (week 4)

Task 2: Testing task1 output by manually. (week 4)

Task 3: Create a grammar based on seed inputs and RFC. (week 5)

Task 4: Create the custom mutator to fuzz CSR. (week 6)

Task 5: Find different CA libraries. (week 7)

Task 6: Run CATest on the CA dataset to find true and false positives. (week 8)

**Custom mutator improves the performance of fuzzer (Yen-Chun Huang):**

Task 1: Run the AFL++ without a custom mutator in the same dataset and time. (Week 4, 5)

Task 2: Check the true positives and false positives. (Week 6)

Task 3: Compare the true positives and false positives. (Week 7)

Task 4: Compare performance AFL++ when using the default strategy. (Week 8)

**Find the CSR validation issues from different CA library github issues and compare CATest with the other state-of-the-art tools (Ming Wei):**

Task 1: find the CSR validation issues from different CA library github issues (Week 4)

Task 2: find the existing tool(validate the CSR) (Week 5)

Task 3: run those tools on the same dataset (Week 6)

Task 4: compare true positives and false positives (Week 7-8)

Thank you for attention!