Ricardo Yaben, rmyl@dtu.dk
Emmanouil Vasilomanolakis, emmva@dtu.dk

Rolling the DICE

A Device Identification and Classification Engine to detect vulnerable devices exposed to the Internet

Introduction

While the Internet is **flooded** with vulnerable devices, current identification methods **struggle to detect even basic vulnerabilities.**

Moreover, scanning results from different Internet measurements are **not comparable**, limiting their potential applications and advancements in the field [1].

Millions of Internet-facing devices are poorly maintained, obsolete, or already compromised.

DICE: The Engine

DICE is designed to **identify** and **classify** vulnerable devices facing the Internet. Our long-term goal is to streamline methods for **conducting** and **comparing** Internet measurements.

This engine opens new opportunities to **monitor** security events across the Internet, such as outages and systemic vulnerabilities.

How does it work?

```
$ dice scan --signatures mqtt,opcua
```

— First, DICE searches for the given signatures to load their modules and dependancies.

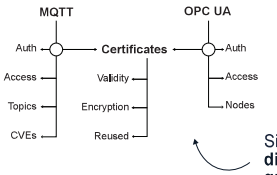
Modules are the signature's **processing units.** They either collect device information, identify devices, or detect vulnerabilities.

```
// opcua.dice
cls opcua-auth
cls opcua-access  (cls: opcua-auth)
cls opcua-nodes (cls: opcua-auth)
sig certificates (cls: opcua-auth)
// mqtt.dice
cls mqtt-auth
cls mqtt-access (cls: mqtt-auth)
cls mqtt-topics (cls: mqtt-auth)
cls mqtt-cve (cls: mqtt-auth)
sig certificates (cls: mqtt-auth)
// certificates.dice
cls x509-validity
cls x509-encryption
cls x509-reused
```

Signatures define **dependancies** between modules and other signatures.

— Then, links signatures and modules verifying there are no cycles.

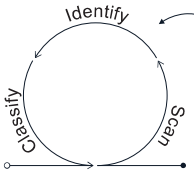
Signatures describe how and which modules are needed to identify and classify devices.



Signatures form **directional acyclic graphs (DAGs)** to sequence modules.

— Finally, sends addresses to its components to scan, identify and classify as needed.

Components orchestrate the **scanning**, **identification** and **classification** tasks.



Components work **independently** and **collaborate** to profile devices.

KEY FINDINGS

We implemented DICE signatures to identify vulnerabilities across 8 protocols commonly used in IoT and OT [2].

Our signatures revealed widespread access control and certificate management security issues.

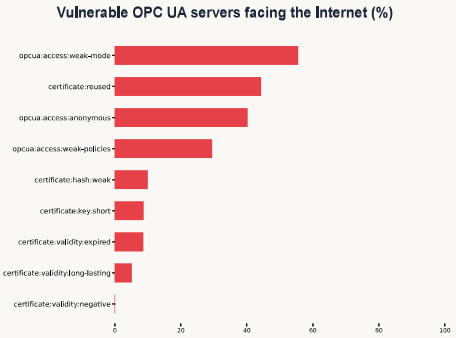


Fig: Distribution of vulnerabilities in OPC UA servers facing the Internet identifyied using DICE signatures for OPC UA and X.509 certificates. Dataset from February 2025.

USE CASES

- Longitudinal studies identifying trends and persistent issues.
- Artificial Intelligence trained of datasets classified using DICE.
- Improve IP exploitation techniques based on results from multiple neasurements.
- Conducting responsible disclosure campaigns to notify the affected owners.

DESIGN PRINCIPLES

- DICE is **modular**, allowing for a wide variety of measurements.
- Collects **measurement metrics** to evaluate its performance.
- Results are **comparable** between measurements.
- Supports most widely used network scanners (e.g., zmap and masscan).

References

[1] K. Claffy, D. Clark, J. Heidemann, F. Bustamante, M. Jonker, A. Schulman, and E. Zegura, "Workshop on overcoming measurement barriers to internet research (wombir 2021) final report," SIGCOMM Comput. Commun. Rev., vol. 51, no. 3, p. 33–40, Jul. 2021.

[2] R. Yaben and E. Vasilomanolakis, "Ricyaben/tma-2025-poster: Early implementation of DICE for the TMA conference (poster session)," May 2025. [Online]. Available: https://github.com/RicYaben/tma-2025-poster

INDEPENDENT RESEARCH FUND DENMARK

Want to learn more?
Extended abstract here