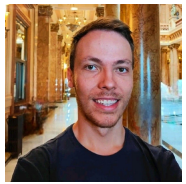
Breaking and Fixing the **OCPP** Electric Vehicle Charging Standard with **CheckOCPP** and **EmuOCPP**

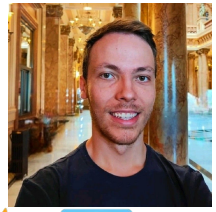
Daniele Antonioli
EURECOM (FR)



Victor Fresno Gómez
BASF (ES)

DEFCON SINGAPORE

Victor Fresno Gómez



- Cyber Security Engineer at BASF (Madrid, Spain☀️🏙️)
  - Vulnerability management, OT security, and system hardening
  - Dual Master’s at EURECOM (France) & UPM (Spain)
- Experience in Cybersecurity and Telematics
  - Protocol Security (OCPP)
  - Network Security & Testing

Daniele Antonioli

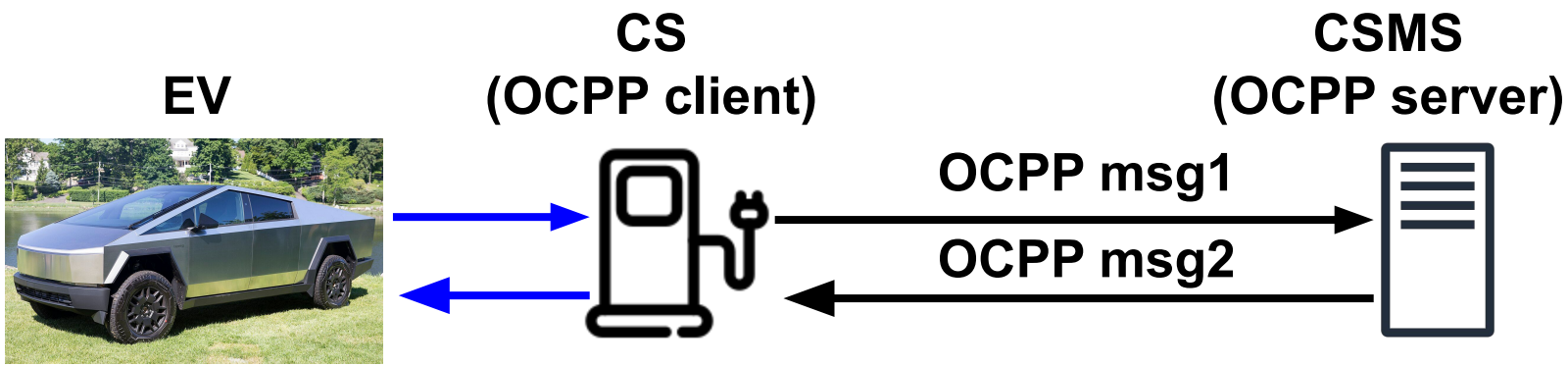
- Asst. Prof at EURECOM (French riviera, 🏖️, 🏔️)
  - Software and System Security (S3) group
  - PhD at SUTD in Singapore! 🇸🇬
- Research **system security and privacy**
  - Protocols (Bluetooth, FIDO2, OCPP, proprietary, …)
  - CPS (Vehicles, IoT, ICS, MiniCPS, …)
  - Threat Modeling (ADF)
  - More at https://francozappa.github.io

DEF CON Talk Outline

1. OCPP Introduction
2. OCPP Vulnerabilities and Attacks
3. CheckOCPP
4. EmuOCPP
5. Attack Demos
6. Evaluation, Mitigation
7. Disclosure & Takeaways

OCPP Introduction

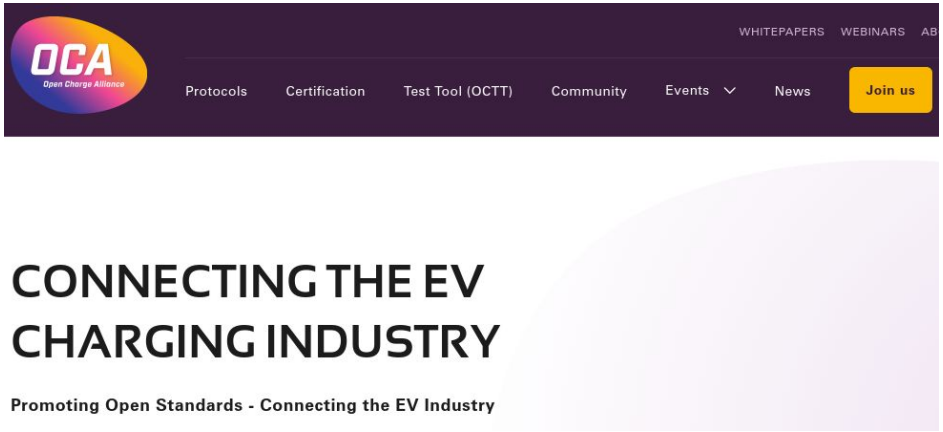
Open Charge Point Protocol (OCPP)



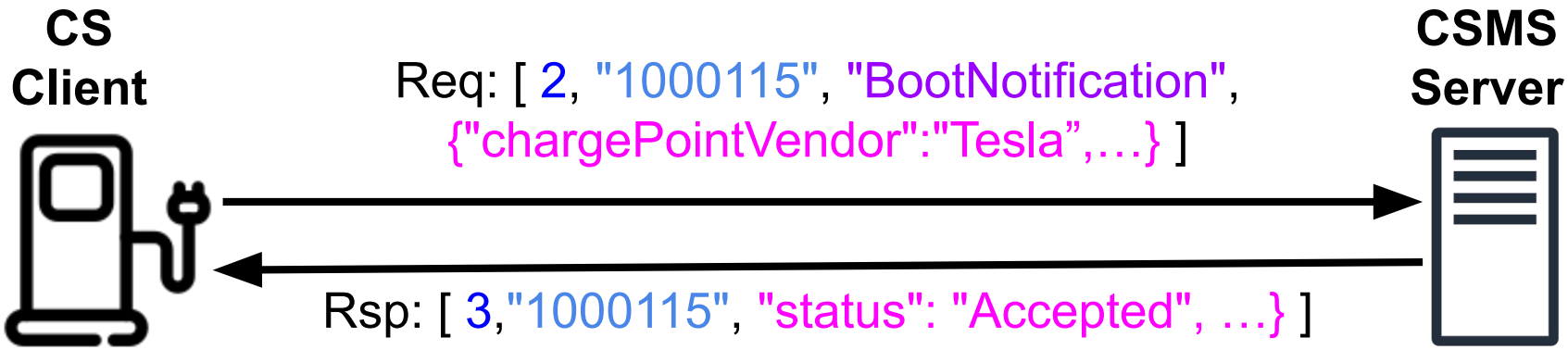
OCPP is a **standard EV charging protocol** built on **Web tech** (TCP, IP, HTTP, WebSocket)

OCPP is Pervasive

- *Vendor-agnostic* and *decentralized*
- Used *worldwide* in 130+ countries (ref)
- Millions of CS and thousands of CSMS
- Maintained by Open Charge Alliance (OCA)



OCPP BootNotification Req and Rsp



OCPP message is a **JSON array** over WebSocket.
Type (int), **UniqID** (str), **Action** (str), **Payload** (obj)

OCPP Bidirectional Messages over WS

- CS requests
  - BootNotification, Heartbeat, Status, ...
- CSMS requests
  - Remote control, Config, Security, CS fw update, Charging mgmt, Diagnostics, …
- CS typically open a new WS connection
  - if CS is offline, CSMS must wait for a BootNofication msg

OCPP Security Profiles

SP1: Basic HTTP Auth

- `ws://csms.example.com/ocpp/`
- user, pwd in plaintext (`Authorization` header)

SP2: TLS + Basic HTTP Auth

- `wss://csms.example.com/ocpp/`
- CSMS presents cert (self- or CA- signed)
- user, pwd tx encrypted over TLS

SP3: mTLS

- `wss://csms.example.com/ocpp/`
- Certs from CS and CMSM (mutual TLS, no phising)
- Required for ISO 15118 Plug & Charge (PnC)

OCPP Protocol Versions (ref)

- v1.5: 2012, legacy (SOAP/HTTP)
- **v1.6:** 2015, *most widely deployed, SPs backported*
- **v2.x:** Add SPs, incompatible with 1.6
  - **v2.0**: 2018, quickly deprecated (on paper)
  - **v2.0.1**: 2020
  - v2.1: 2025, not available during out tests

OCPP SP and Protocol Version Negotiations

- SP negotiation
  - SP1 vs SP2,SP3 using **URL scheme**
  - SP2 vs SP3 using **TLS handshake**
- Protocol version negotiation
  - WebSocket subprotocol header
  - CS sends a list wth priorities
  - CSMS picks one
- Negotiations are **orthogonal**
  - First SP then protocol version

Motivation

- OCPP is a **critical EV attack surface**
  - Complex protocol (downgrade, MitM, spoof, …)
  - Networked (remote attacks)
  - Bidirectional (attacks from client and server!)
- OCPP attacks affect **security, privacy and safety**
  - Stealing credentials, charging fraud, (D)DoS, session hijacking, malicious fw update, track EVs, …
- One **design vuln** in OCPP standard
  - Millions of vulnerable OCPP devices

Contributions

- OCPP spec security assessment
- **8 OCPP attacks (5 are new),** (MitM, Imp, DoS, Eave)
- **5 new OCPP design vulns** (SP upgr/downgr, …)
- Exploit **6 popular CS and CSMS** (MobHouse, SteVe, …)
- **CheckOCPP** for OCPP compliance checking (Lua)
- **EmuOCPP** for OCPP S&P testing (IPmininet)
- **5 effective attacks fixes**
- Responsible disclosure

Talk Covers Two Research Papers

CheckOCPP: Automatic OCPP Packet Dissection and Compliance Check

Soumaya Boussaha
SAP, EURECOM
Biot, France
soumaya.boussaha@sap.com

Victor Fresno Gómez
EURECOM, UPM
Madrid, Spain
victorfresno@live.com

Thomas Barber
SAP
Baden-Wurtemberg, Germany
thomas.barber@sap.com

Daniele Antonioli
EURECOM
Biot, France
daniele.antonioli@eurecom.fr

https://francozappa.github.io/publication/2025/checkocpp/

EmuOCPP: Effective and Scalable OCPP Security and Privacy Testing

Soumaya Boussaha
SAP, EURECOM, Biot, France
soumaya.boussaha@sap.com

Victor Fresno Gómez
EURECOM, UPM, Madrid, Spain
victorfresno@live.com

Thomas Barber
SAP SE, Walldorf, Germany
thomas.barber@sap.com

Daniele Antonioli
EURECOM, Biot, France
daniele.antonioli@eurecom.fr

https://francozappa.github.io/publication/2025/emuocpp/

Ack to co-authors and funding members!

Soumaya Boussaha
SAP, EURECOM
Biot, France
soumaya.boussaha@sap.com

Victor Fresno Gómez
EURECOM, UPM
Madrid, Spain
victorfresno@live.com

Thomas Barber
SAP
Baden-Wurtemberg, Germany
thomas.barber@sap.com

Daniele Antonioli
EURECOM
Biot, France
daniele.antonioli@eurecom.fr

ORSHIN

EURECOM
Sophia Antipolis

SAP

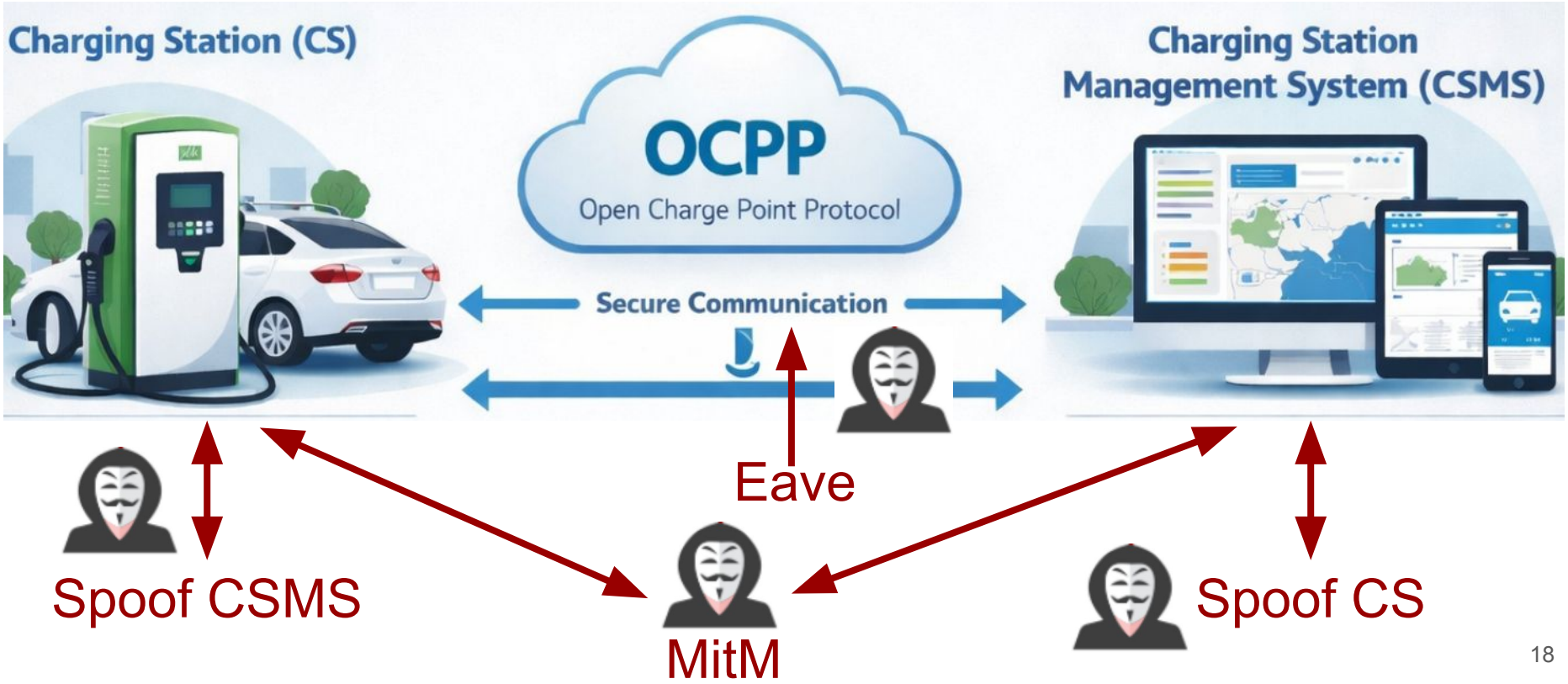
FRANCE 2030
PROGRAMME DE RECHERCHE
RÉSEAUX DU FUTUR

European Commission
Horizon Europe

UNIVERSIDAD POLITÉCNICA MADRID

OCPP Vulnerabilities and Attacks

OCPP Protocol-Level Threat Model



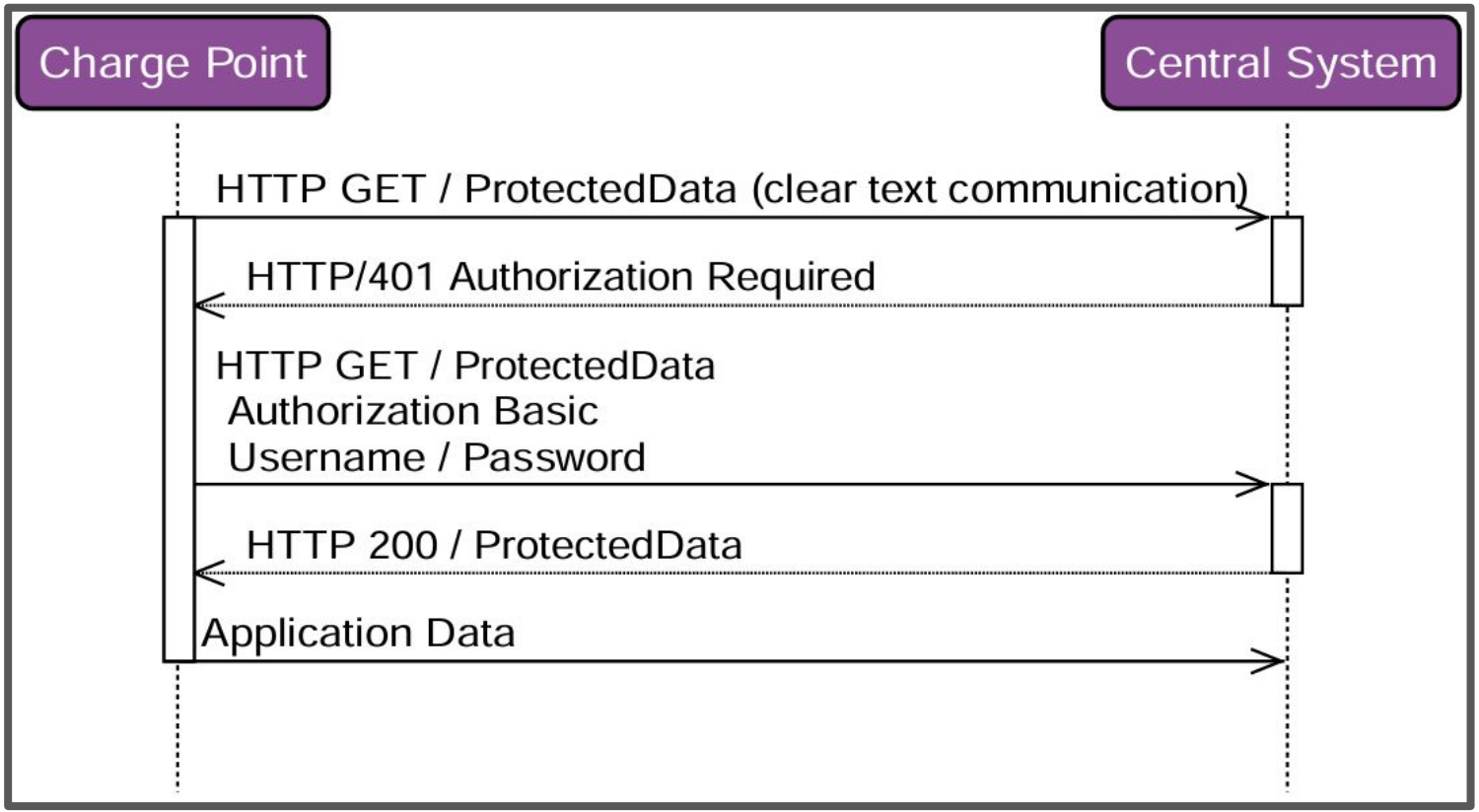
Eight Attacks exploiting OCPP Des and Impl Vulns

ID	Name	New	Type	OCPP SP	OCPP Ver.	Impact	Vulnerability
M1	MitM SP1	✗	Des	SP1	1.6, 2.0, 2.0.1	Sec, Pri	Weak SP1
M2	MitM SP2 Upgrade	✓	Des	SP2	1.6, 2.0, 2.0.1	Sec, Pri	Weak SP2 upgrade
M3	MitM SP3 Upgrade	✓	Des	SP3	1.6, 2.0, 2.0.1	Sec, Pri	Weak SP3 upgrade
M4	MitM SP Downgrade	✓	Imp	SP2, SP3	1.6, 2.0, 2.0.1	Sec, Pri	No SP down prot
E1	Eavesdrop	✗	Des	SP1	1.6, 2.0, 2.0.1	Sec, Pri	Weak SP1
D1	CSMS (D)DoS	✗	Des	SP1, SP2, SP3	1.6, 2.0, 2.0.1	Sec	Unauth CS boot
I1	CS Impersonation	✓	Des, Imp	SP1, SP2, SP3	1.6, 2.0, 2.0.1	Sec, Pri	CS ID undefined behaviour
I2	CS Impersonation	✓	Des, Imp	SP1, SP2, SP3	1.6, 2.0, 2.0.1	Sec, Pri	CS status trackable by anyone

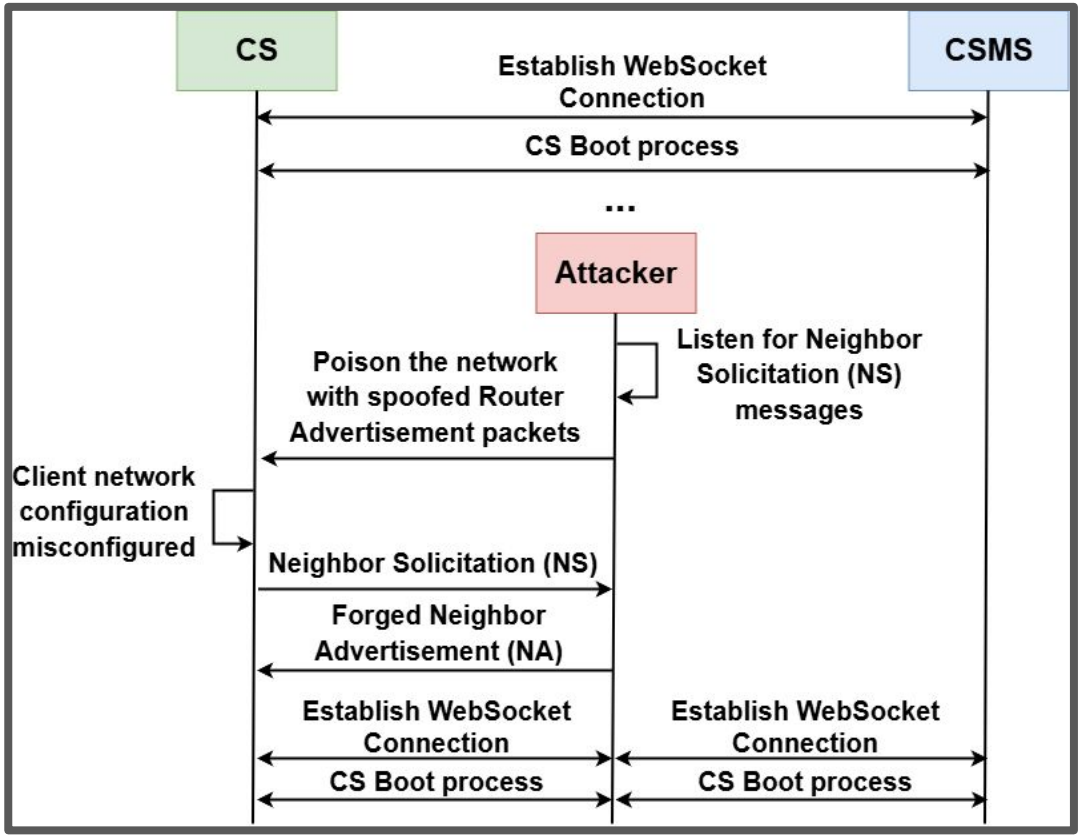
MitM, Impersonation, New Attack

Next we see **M1, M2, M3, M4, I1**

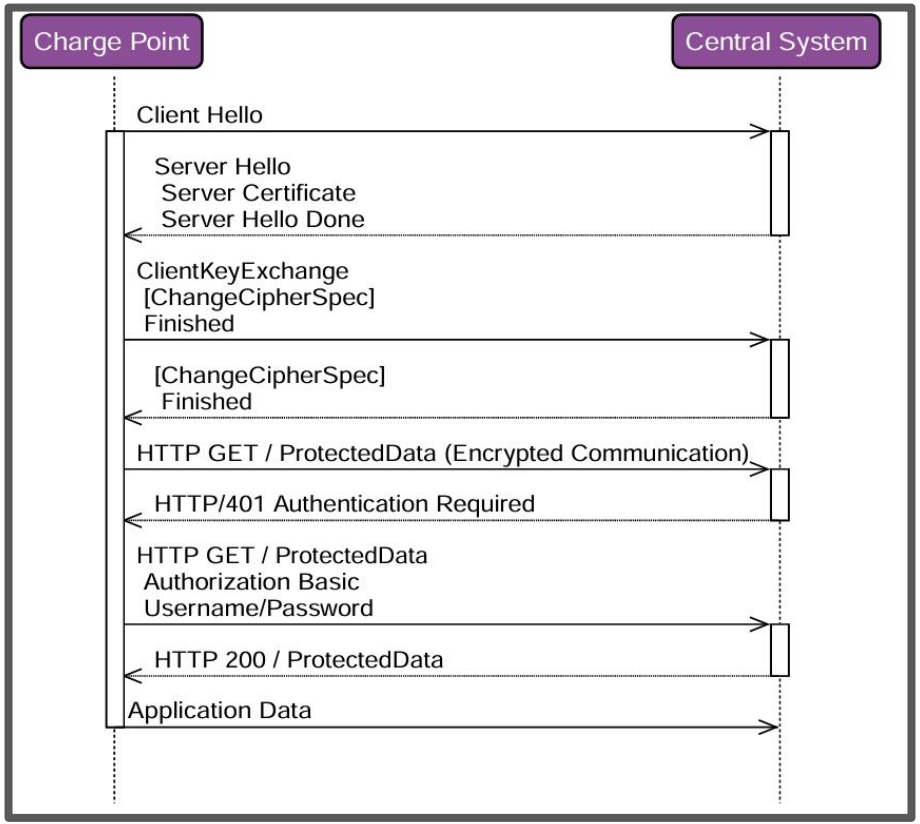
OCPP SP1: Basic Authentication



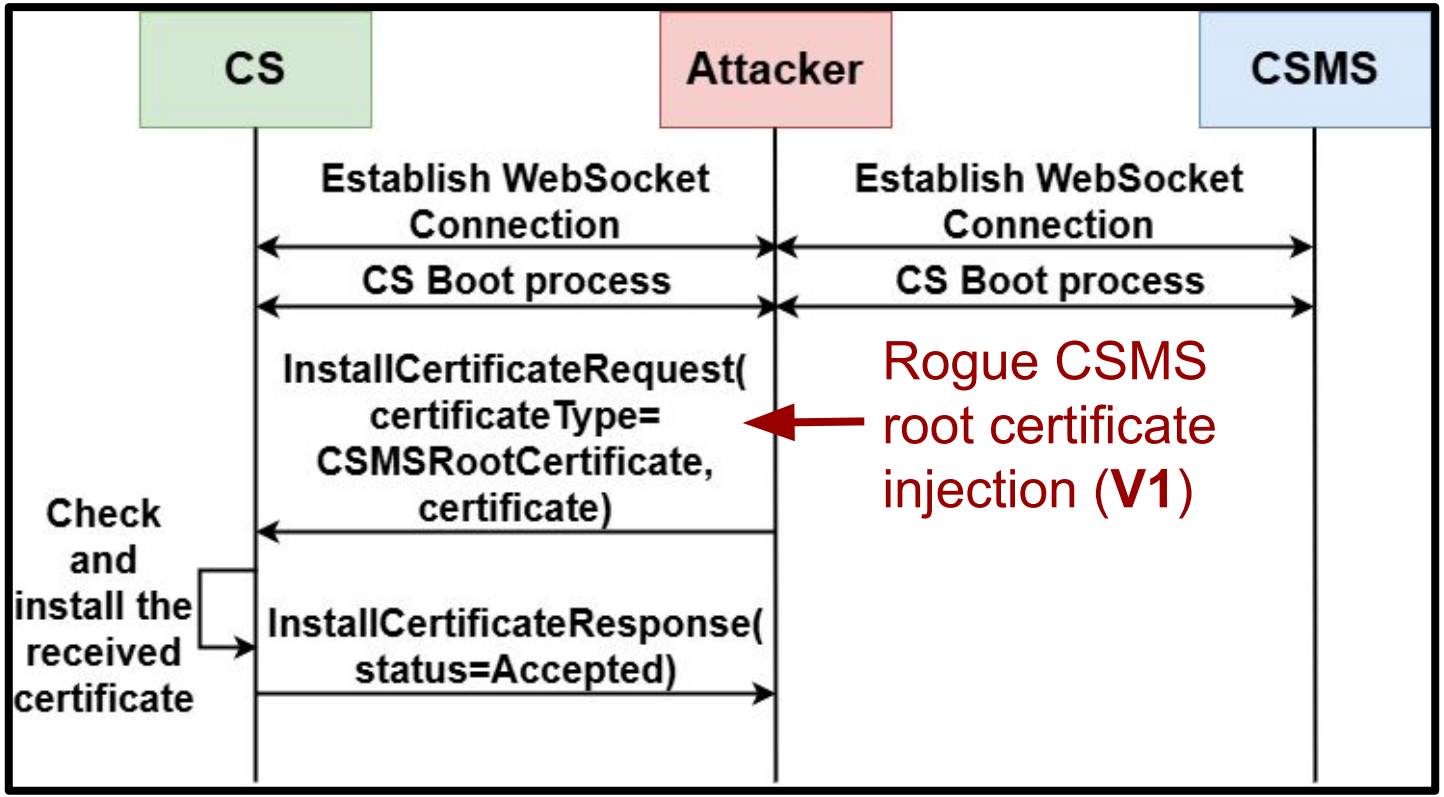
M1: MitM against Insecure SP1



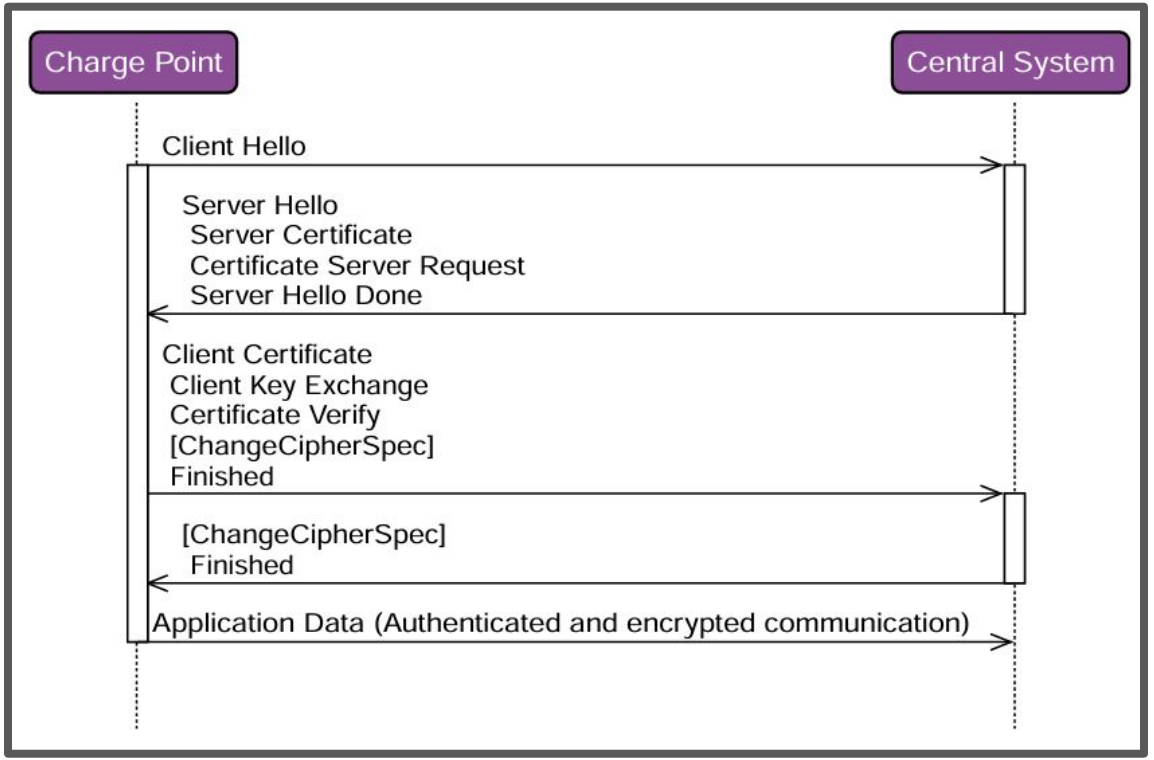
OCPP SP2: TLS with Basic Authentication



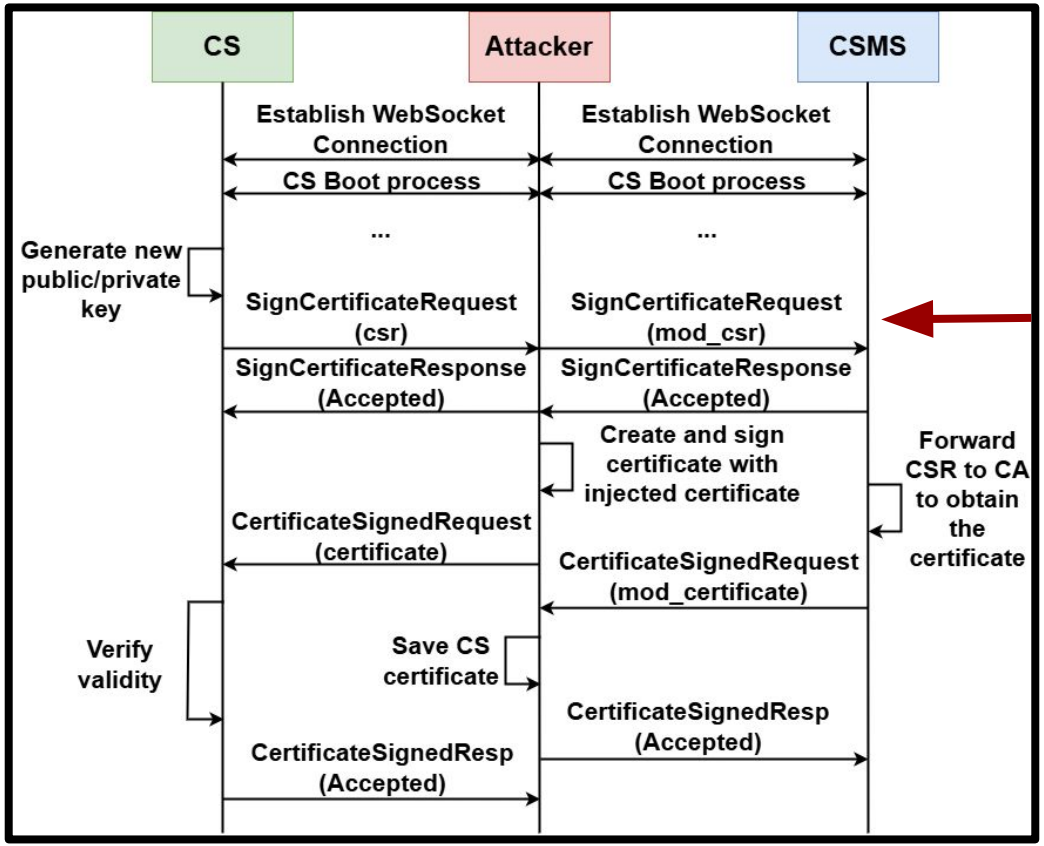
M2: MitM via Insecure SP2 Upgrade



OCPP SP3: TLS with Client Cert Auth

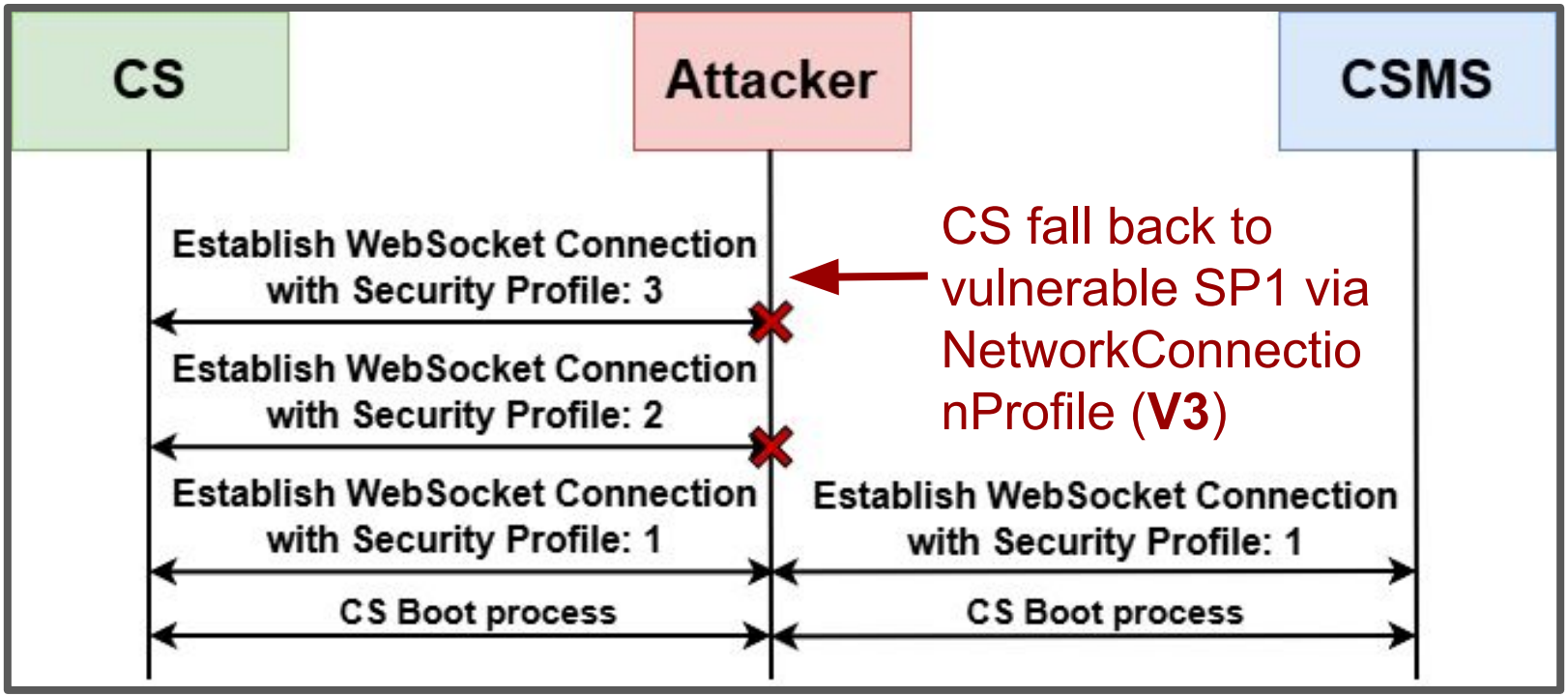


M3: MitM via Insecure SP3 Upgrade

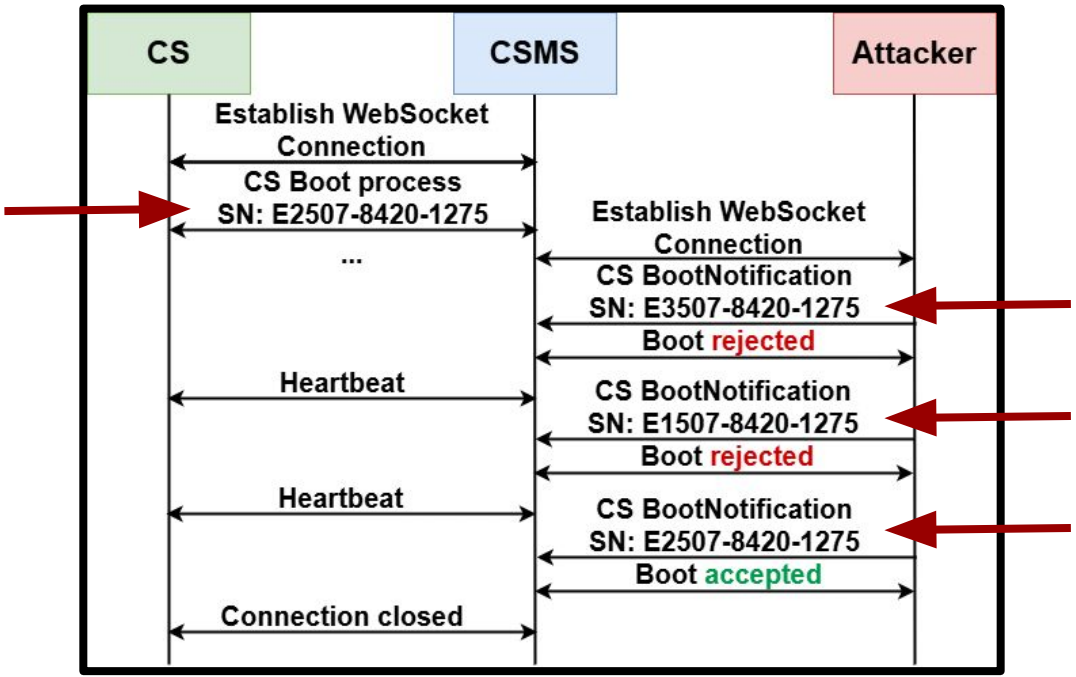


Rogue CSR injection and double client certificate (**V2**)

M4: MitM via SP Downgrades



I1: CS Impersonation



Duplicate SN in BootNotification is undef behaviour (**V4**)

Five OCPP Design Vulnerabilities

V1: SP2 upgrade CSMS root cert injection
V2: SP3 upgrade client CSR substitution
V3: Unprotected SP downgrade via legacy profiles
V4: Undefined behavior with duplicate SN
V5: Unauth CS connection status tracking (I2 attack)

Wide Impact on OCPP: **charging fraud, tracking EVs, DoS smart grid,** …

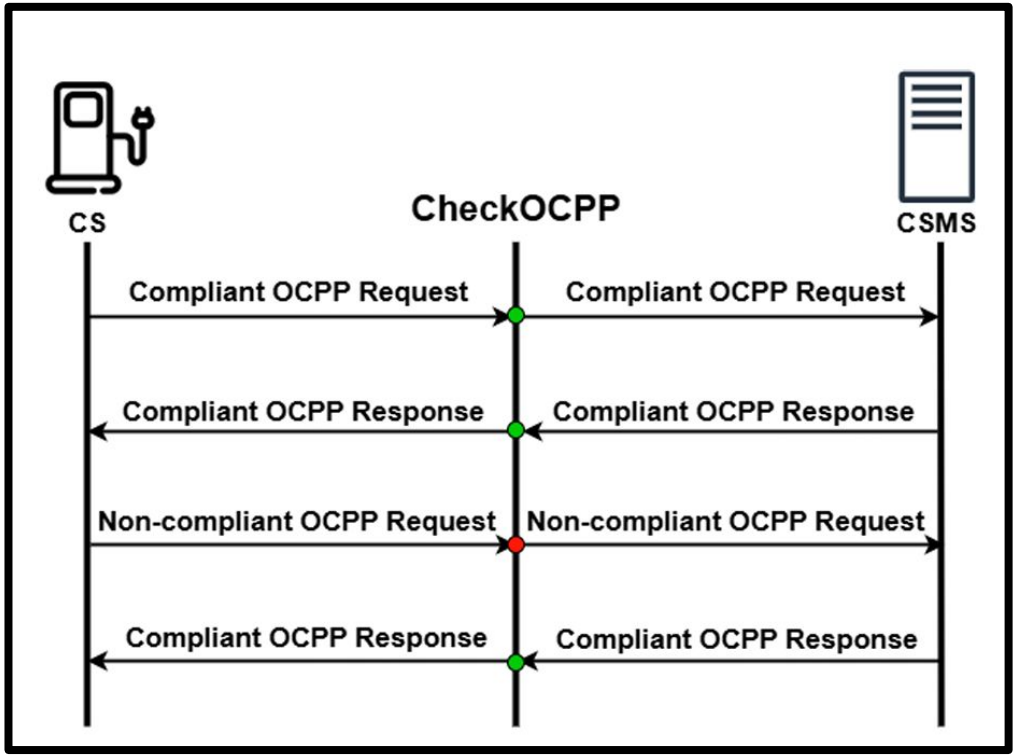
Finding OCPP Attacks and Vulnerabilities?

- **OCTT**
  - Official compliance test (no security)
  - Paywalled (20K USD), closed-source
- **OCPPStorm**
  - OCPP fuzzer (memory corruption)
  - No design and impl logic vulns
- We created **CheckOCPP** and **EmuOCPP**!
  - Open, low-cost, scalable
  - Security, privacy, safety
  - OCPP v1.6, v2.0.0, v2.0.1, SP1, SP2, SP3

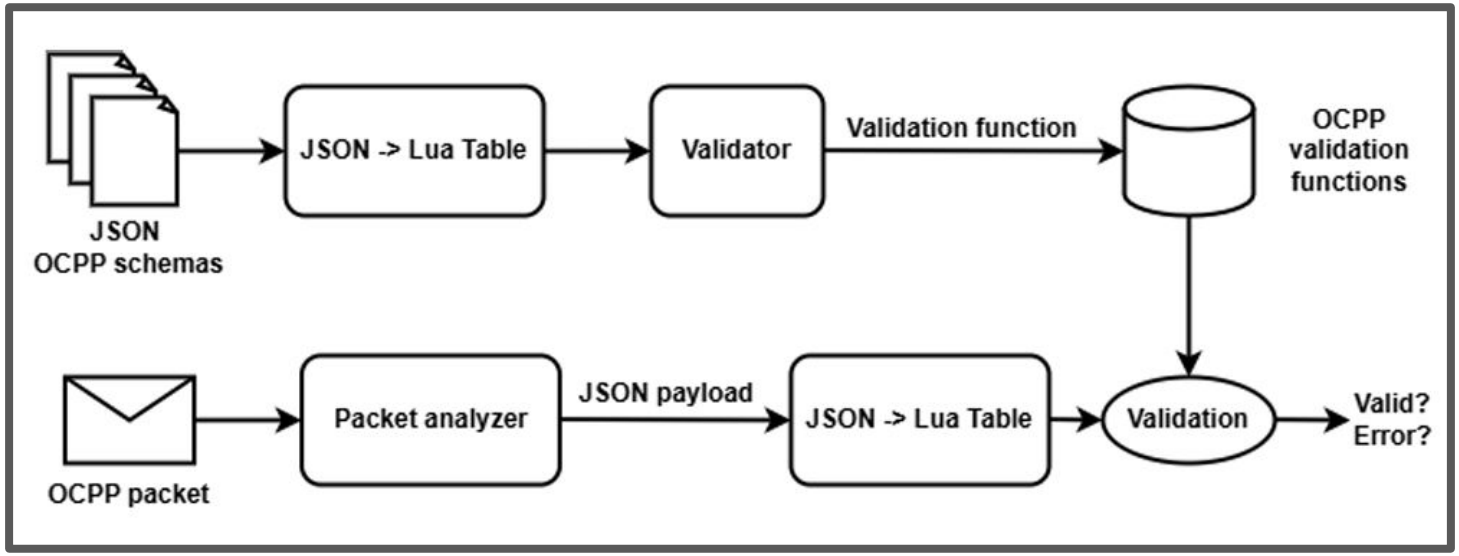
CheckOCPP

CheckOCPP

- Real-Time Interception
- Deep Extraction
- Dynamic Validation

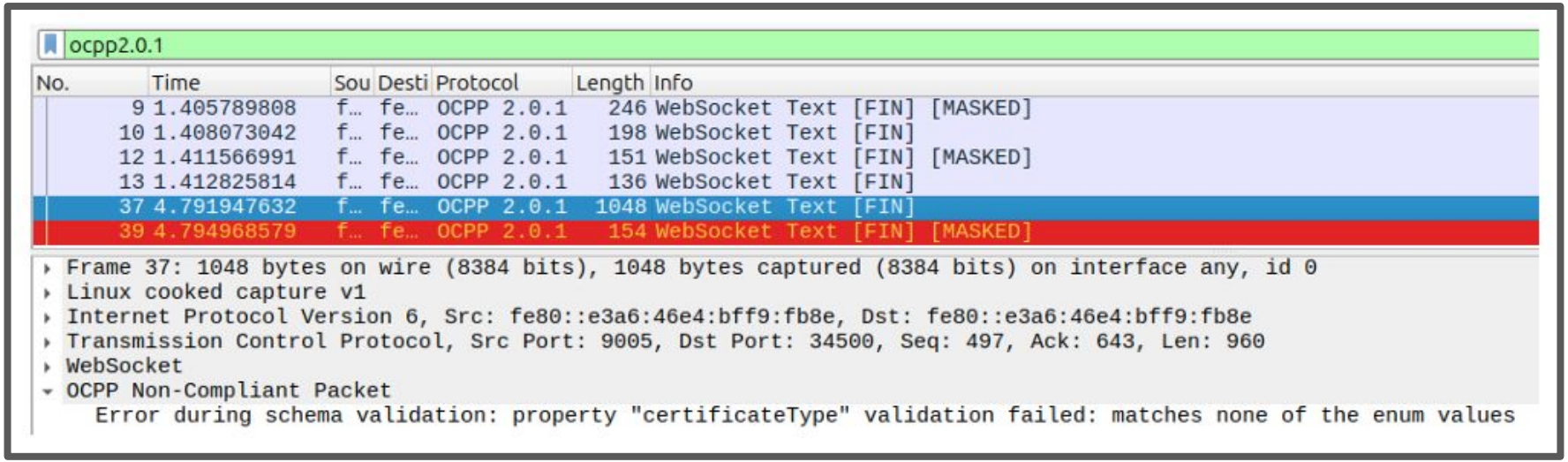


CheckOCPP - Architecture



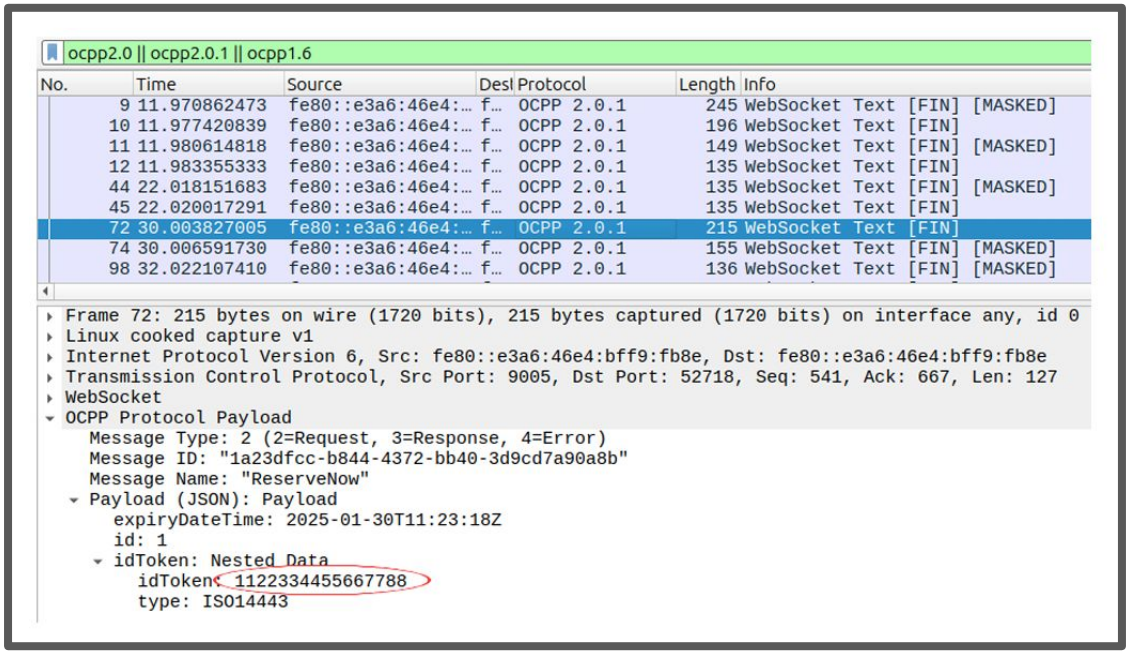
Lua+cjson parsing with structured OCPP tables
Schema validation + error highlighting in Wireshark

CheckOCPP - Validation error highlighting (2.0.1)



OCPP 2.0/2.0.1 parsing with mixed packets
Protocol violations detection (invalid fields)

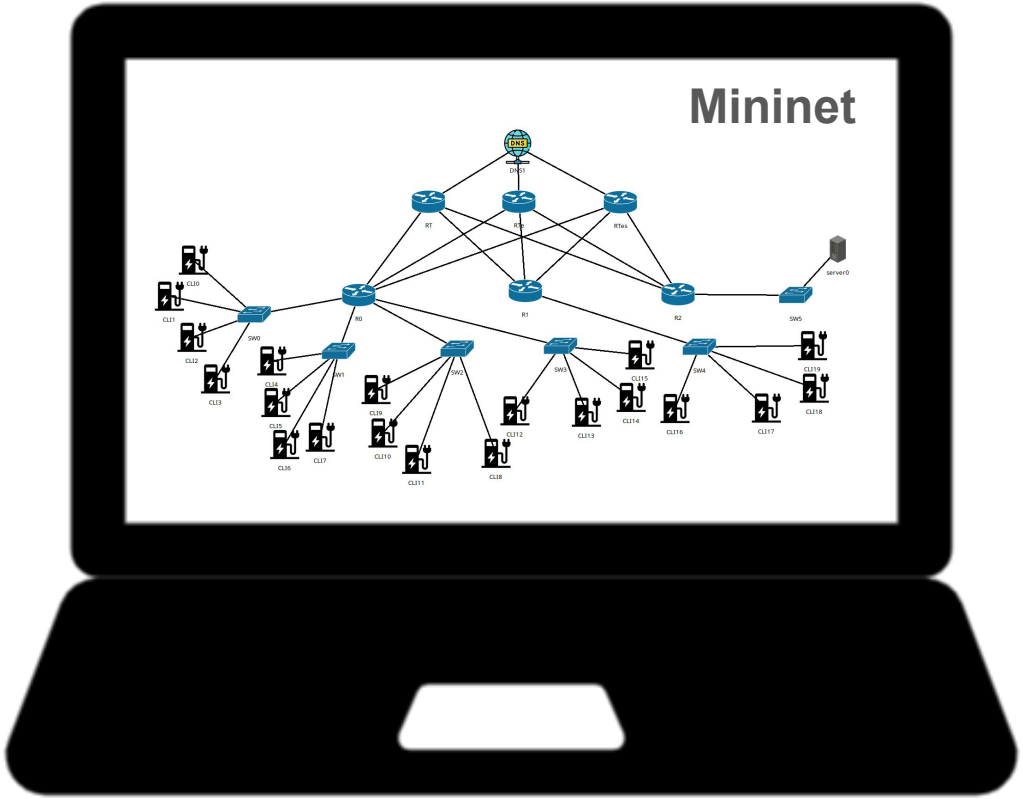
CheckOCPP - Sensitive data



Sensitive data display (e.g., idTokens)

EmuOCPP

EmuOCPP: A Charging Network in a Laptop

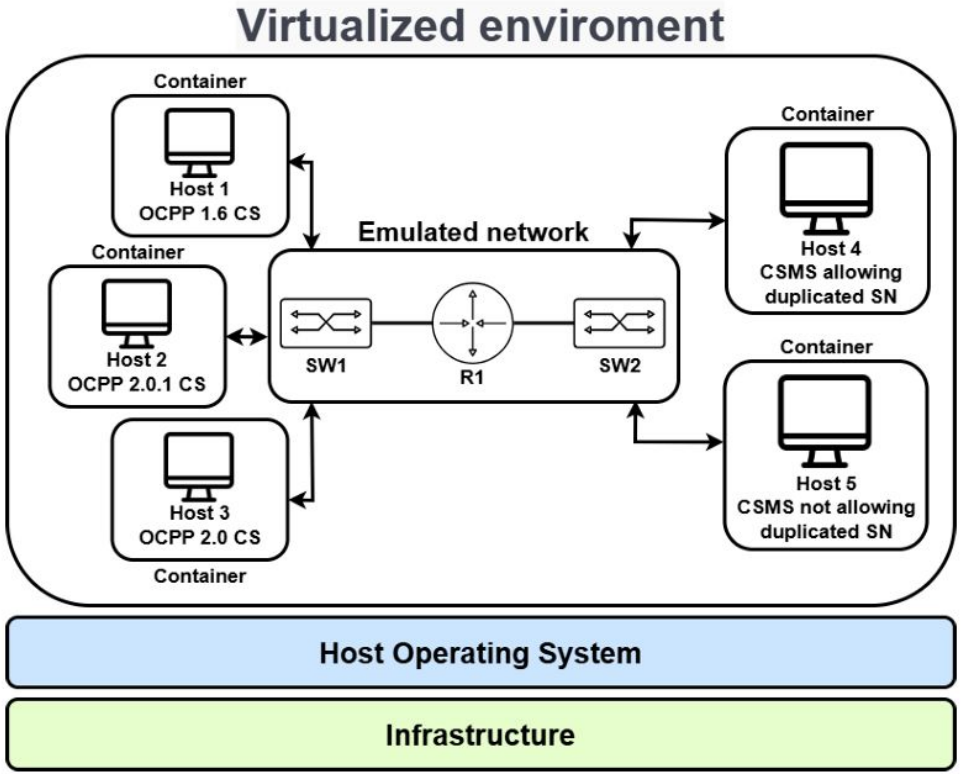


Mininet: An Instant Virtual Network on Your Laptop (or Other PC) - Mininet

EmuOCPP - Requirements

R1: OCPP 1.6, 2.0, 2.0.1
R2: OCPP SP1, SP2, and SP3
R3: Emulate OCPP network
R4: Test S&P
R5: Scalable
R6: Open and low-cost

EmuOCPP - Features



EmuOCPP Supports 17 OCPP Messages

Message	Version	Category
Heartbeat	1.6, 2.0, 2.0.1	Core
BootNotification	1.6, 2.0, 2.0.1	Core
Authorize	1.6, 2.0, 2.0.1	Core
StatusNotification	1.6, 2.0, 2.0.1	Core
TransactionEvent	2.0, 2.0.1	Core
Reset	2.0, 2.0.1	Core
SignCertificate	1.6, 2.0, 2.0.1	Security
InstallCertificate	1.6, 2.0, 2.0.1	Security
CertificateSigned	1.6, 2.0, 2.0.1	Security
GetVariables	2.0, 2.0.1	Config&Mgmt
SetVariables	2.0, 2.0.1	Config&Mgmt
SetNetworkProfile	2.0, 2.0.1	Config&Mgmt
GetConfiguration	1.6	Config&Mgmt
ChangeConfiguration	1.6	Config&Mgmt
TriggerMessage	2.0, 2.0.1	Remote control
ExtendedTriggerMessage	1.6	Remote control
ReserveNow	1.6, 2.0, 2.0.1	Reservation

Core

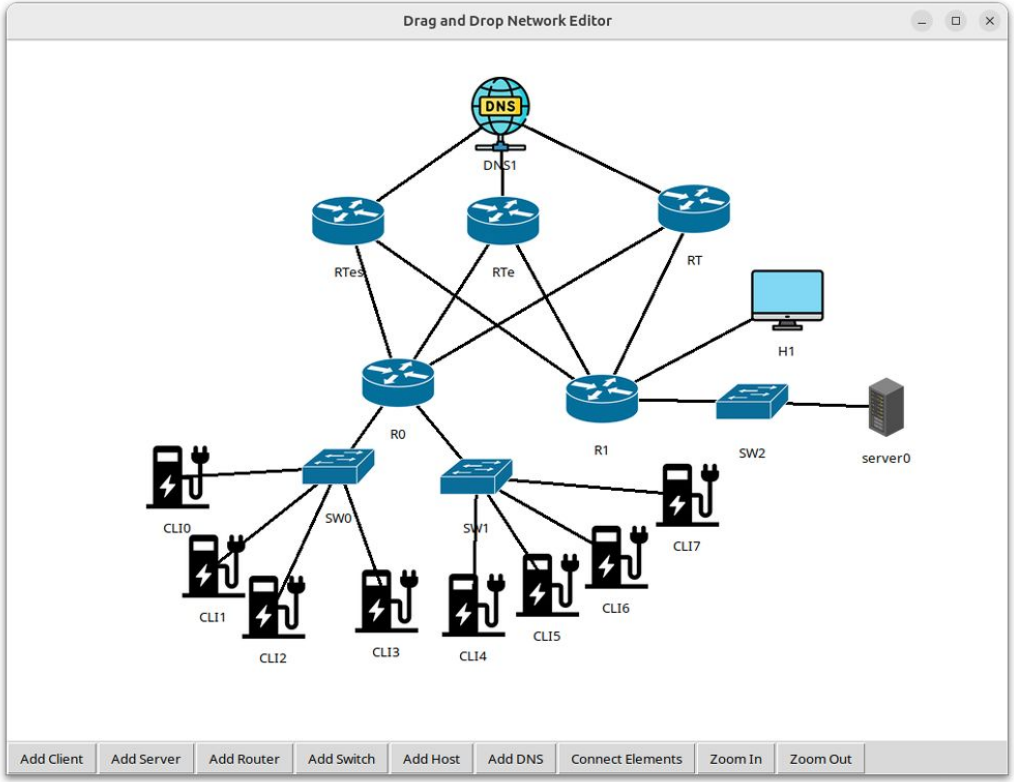
Security

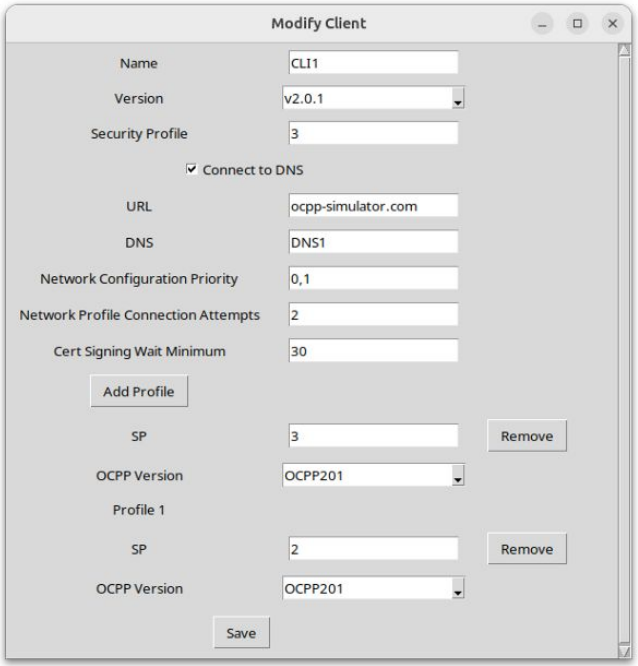
Config &Mgmt

Remote ctrl

Reservation

EmuOCPP GUI


Drag and Drop Network Editor
DNS
DNS1
RTes
RTe
RT
H1
R0
R1
SW2
server0
CLI0
SW0
SW1
CLI7
CLI1
CLI2
CLI3
CLI4
CLI5
CLI6
Add Client
Add Server
Add Router
Add Switch
Add Host
Add DNS
Connect Elements
Zoom In
Zoom Out

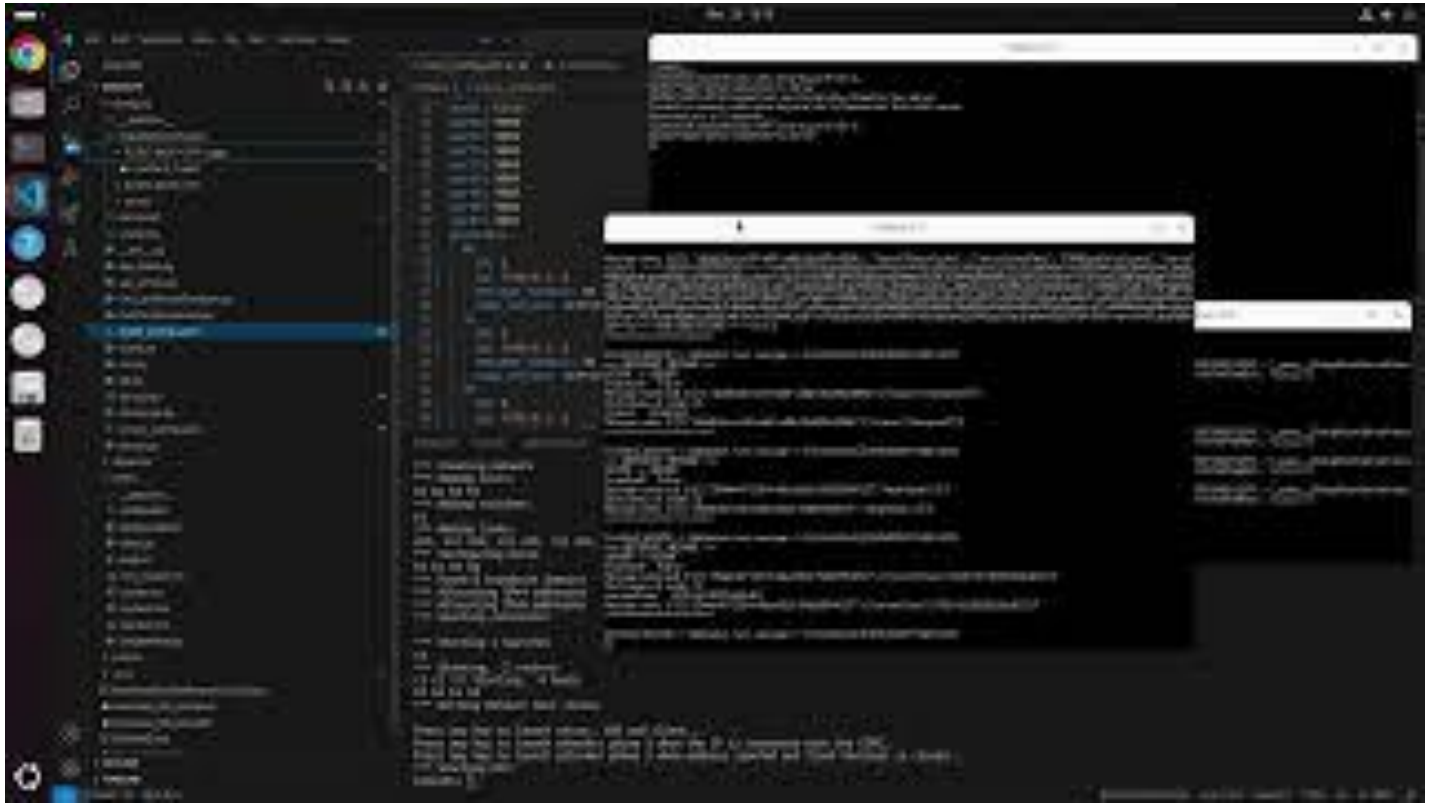

Modify Client
Name
CLI1
Version
v2.0.1
Security Profile
3
Connect to DNS
URL
ocpp-simulator.com
DNS
DNS1
Network Configuration Priority
0,1
Network Profile Connection Attempts
2
Cert Signing Wait Minimum
30
Add Profile
SP
3
Remove
OCPP Version
OCPP201
Profile 1
SP
2
Remove
OCPP Version
OCPP201
Save

EmuOCPP YAML Configuration

- Config CS version and SP
- Define topology: routers, CS, CSMS
- Auto-generate creds and certs
- Easy sharing
- One-click deployment

```yaml
clients:
  client0:
    SecProfile: 2
    attempts: 2
    dns: DNS1
    name: CLI0
    priority:
    - 0
    - 1
    profiles:
      0:
        SP: 2
        ocpp_version: OCPP201
      1:
        SP: 2
        ocpp_version: OCPP201
    url: ocpp-emulator.com
    version: v2.0.1
    wait: '30'
dns:
  dns1:
    name: DNS1
hosts:
  H1:
    name: Attacker
servers:
  server0:
    dns: DNS1
    multiple: 2
    name: server0
    url: ocpp-emulator.com
switches:
  switch0:
    name: SW0
```

EmuOCPP M1 M2 M3 Attacks Demo



Evaluation

Mobility House [ref]

- Python OCPP library
  - CS/CSMS framework
- Protocol implementation
  - Multi-version support
- Development toolkit
  - Extensible codebase

SteVE [ref]

- Open-source CSMS
  - Java-based
- OCPP 1.6 support
  - Single-version implementation
  - No security profiles
- Backend system
  - Central management

OpenEVSE [ref]

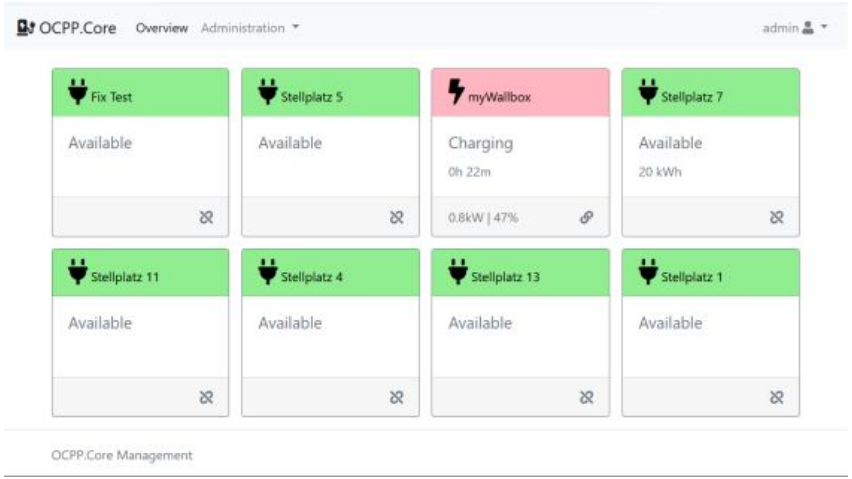
- Open-source EV charger
  - Hardware/software platform
  - OCPP 1.6
  - No security profiles
- Charging infrastructure
  - Real device
- Embedded system
  - Custom firmware

Open E-Mob CS [ref]

- Charging station simulator
  - Node.js application
- OCPP 1.6 support
  - CS emulation
  - No security profiles
- Testing tool
  - Simulated environment

OCPP.Core [ref]

- OS CSMS in .NET 8
- OCPP 1.6J and 2.0 (JSON/REST)
- Web UI managing CS and RFID
- Implements load mgmt and registration.



Evaluation Results (2025)

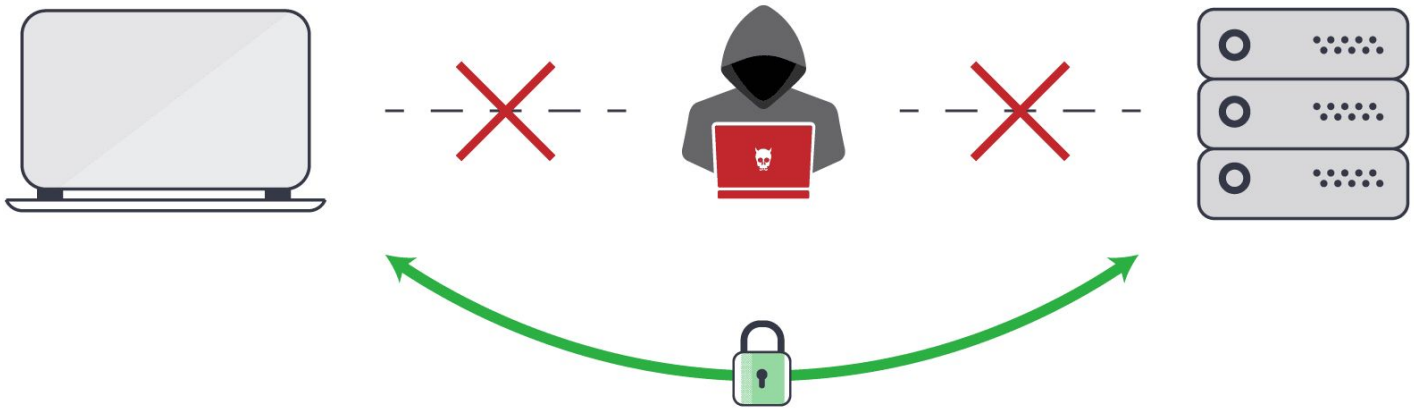
Target	Ver	SP	M1	M2	M3	M4	E1	D1	I1	I2
Mobility House	1.6	All	✓	✓	✓	✓	✓	✓	✓	✓
Mobility House	2.0	All	✓	✓	✓	✓	✓	✓	✓	✓
Mobility House	2.0.1	All	✓	✓	✓	✓	✓	✓	✓	✓
SteVe	1.6	None	✓	NA	NA	NA	✓	✓	✓	✓
Open E-Mob	1.6	None	✓	NA	NA	NA	✓	✓	✓	✓
OpenEVSE	1.6	None	✓	NA	NA	NA	✓	NA	✓	✓
OCPP.Core	1.6	None	✓	NA	NA	NA	✓	✓	✗	✓
OCPP.Core	2.0	None	✓	NA	NA	NA	✓	✓	✗	✓
Prod Network	1.6	SP2	NA	NA	NA	NA	NA	✓	✓	✓

1 CS, **4 CSMS**, **4 CS&CSMS**

✓ : **Vulnerable**, NA: Not Applicable, ✗ : Not vulnerable

Mitigations

Mitigations - MitM protections (M1, E1)



Enforce SP2/SP3 trusted certificates
Prevent LL MitM attacks

Mitigations - Certificate attacks (M2, M3)

Restrict trusted root certificates

Enforce certificate chain validation

Avoid permissive trust models

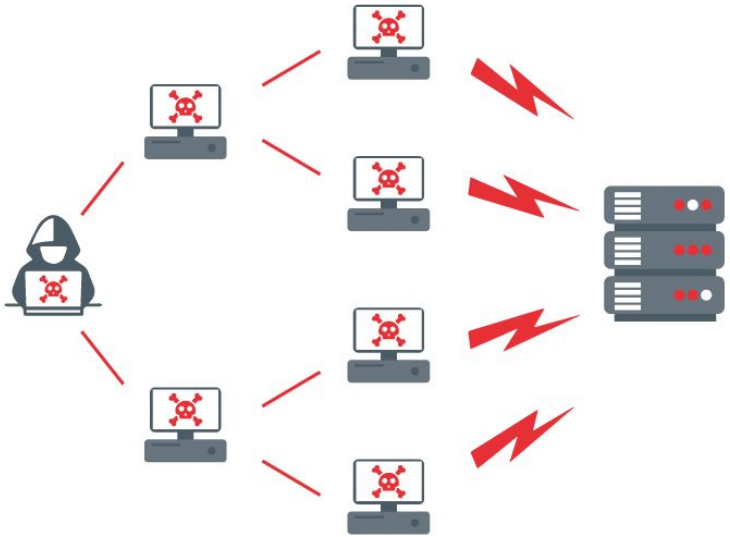
Mitigations - Downgrade attack (M4)

Enforce minimum required SP

Reject weaker security configurations

Enforce consistent SP usage

Mitigations - DoS/DDoS (D1)



Apply rate limiting protections
Use WAF and filtering

Mitigations - Identity & Leakage attacks (I1 & I2)

- Allowlist valid CS IDs
- Enforce single CS connection
- Hide status from unauth CS
- Detect anomalous reconnects

Disclosure and takeaways

Responsible Disclosure (2025)

- OCA answered quickly and organized a meeting
- Updated OCPP security certification tests
- New security white papers
- No technical internal meeting
- Asked to jointly present at VehicleSec
- No OCPP bug bounties

Takeaways

- OCPP 1.6 widely deployed, under-secured
- OCPP 2.x is an underexplored attack surfaces
- SP3 and SP2 can be bypassed in practice
- TLS does not mean OCPP security by desing
- Many real-world CS and CSMS are vulnerable to critical attacks (MitM, Impersonation, DoS)

Conclusions and Q&A

- OCPP spec security assessment
- **8 OCPP attacks (5 are new),** (MitM, Imp, DoS, Eave)
- **5 new OCPP design vulns** (SP upgr/downgr, …)
- Exploit **6 popular CS and CSMS** (MobHouse, SteVe, …)
- **5 effective attacks fixes**
- https://github.com/vfg27/CheckOCPP
- https://github.com/vfg27/EmuOCPP