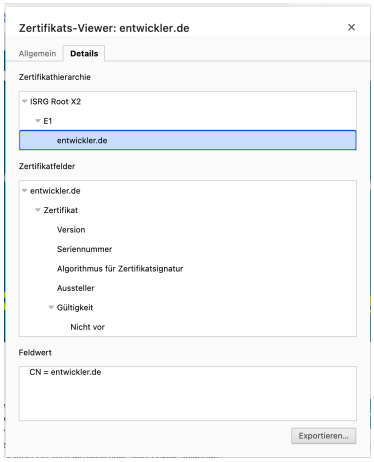
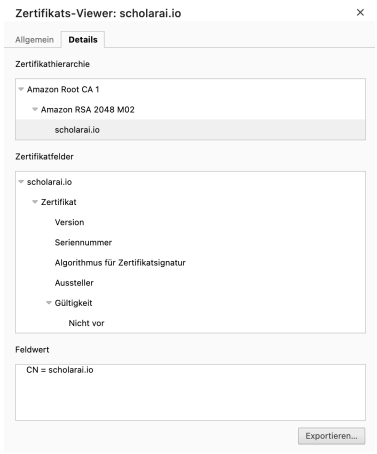
TLS/PKI

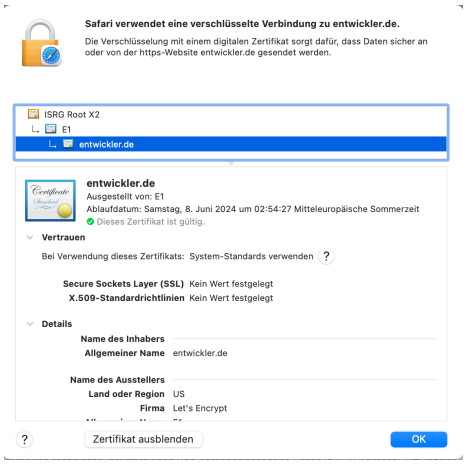
1) Open different https websites from at least 3 different browsers and take a close look at the certificates:
   - ParameterCertificate
   - Chain
   - Expiry date
   - Root CA
   - other fields

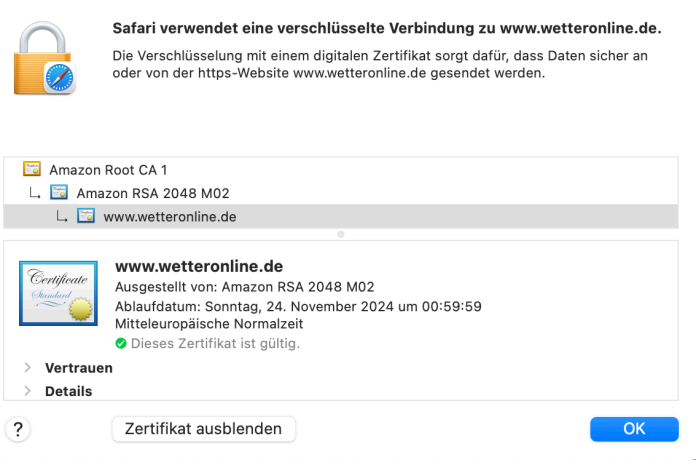
Chrome:



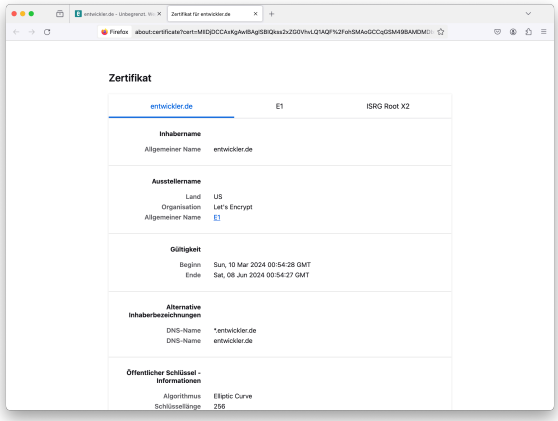


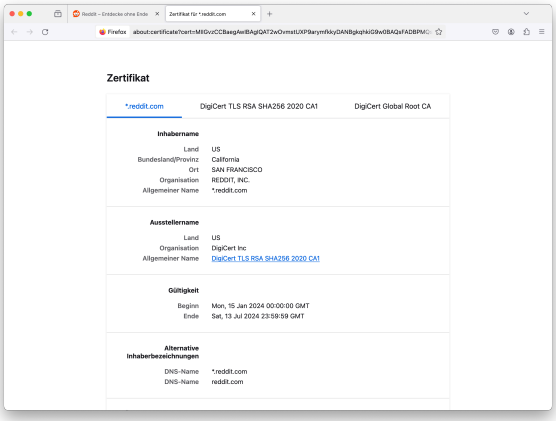
Safari:





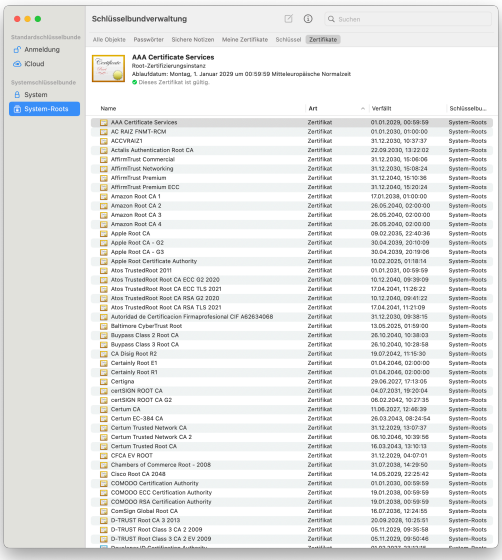
Firefox:

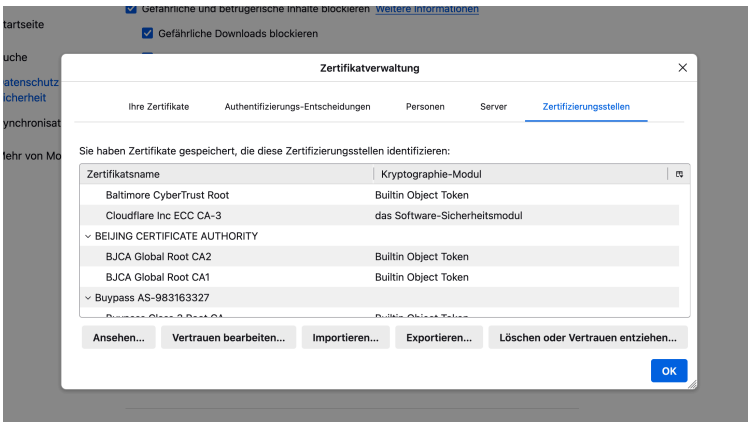




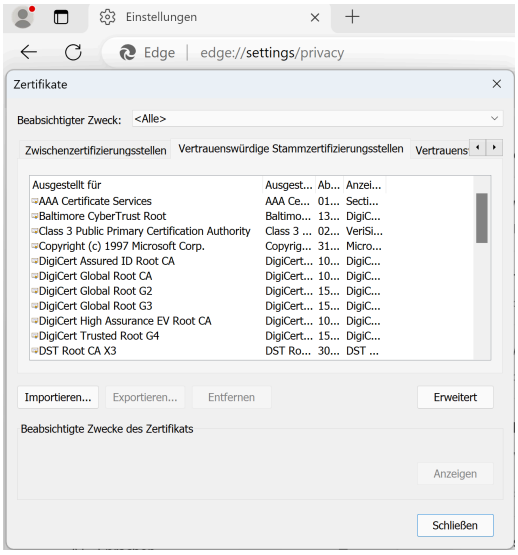
2) Open and see at least 3 different browsers on a total of at least 2 different operating systems, look at the installed root CAs, and compare the list.

Chrome is leading to the macOS keychain in my case:





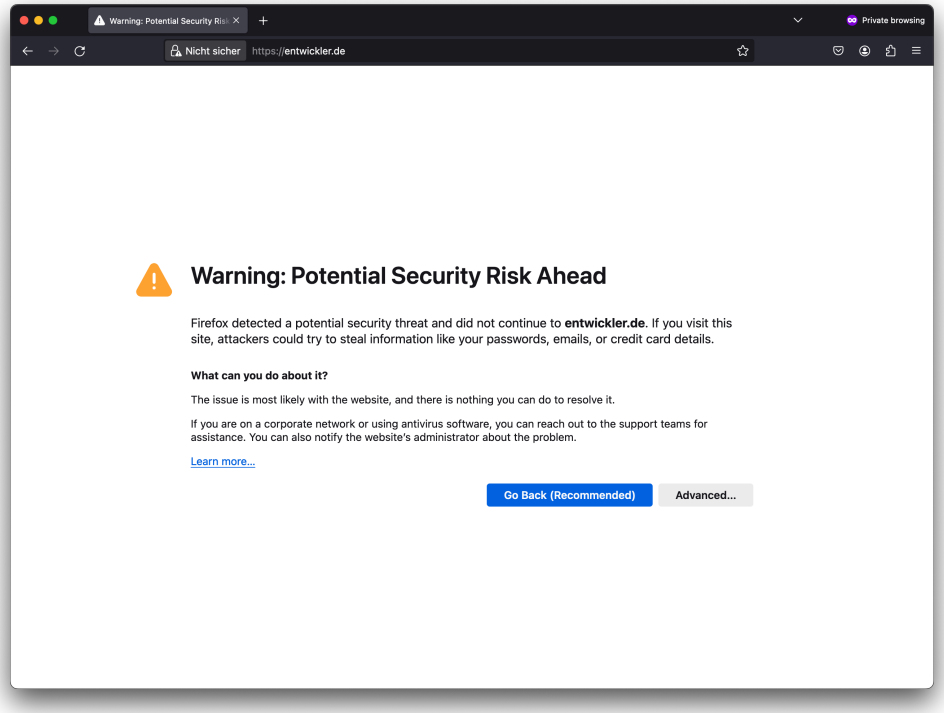
Windows:

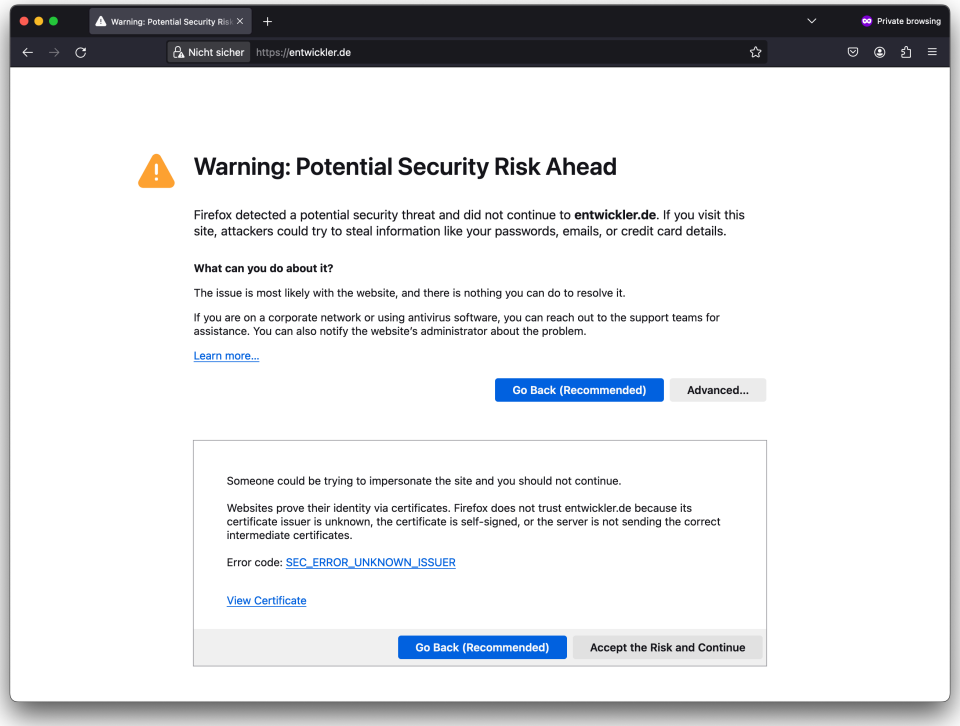


The root certificate authority lists from macOS and Windows, accessed via Chrome, Firefox or Edge, are not identical but have several CAs in common, such as Baltimore CyberTrust Root, Buypass, Amazon Root CAs, and DigiCert. This overlap ensures cross-platform security and trust, while the differences reflect each platform's unique trust ecosystem.

3) Install the Firefox browser, visit any https website, and look at the root CA of that website. Then withdraw trust in Firefox from this root CA and visit the website again. What error message do you get?

I received the error „SEC_ERROR_UNKNOWN_ISSUER. " It has to be noted that I had to open an incognito tab to provoke the error after trust withdrawal.





Diffie-Hellman Key Exchange

1)

```
class DiffieHellman {

    public static void main(String[] args) {
        BigInteger prime = new BigInteger( val: "17");
        BigInteger base = new BigInteger( val: "12");
        BigInteger secretA = new BigInteger( val: "88");
        BigInteger secretB = new BigInteger( val: "104");

        BigInteger publicA = base.modPow(secretA, prime);
        BigInteger publicB = base.modPow(secretB, prime);

        BigInteger sharedSecretA = publicB.modPow(secretA, prime);
        BigInteger sharedSecretB = publicA.modPow(secretB, prime);

        System.out.println("Public Key A: " + publicA);
        System.out.println("Public Key B: " + publicB);
        System.out.println("Shared Secret: " + sharedSecretA);
        System.out.println("Shared Secret the same? " + sharedSecretB.equals(sharedSecretA));
    }
}
```

```
ug  DiffieHellman
    Threads & Variables  Console
/Users/tim.herbst/Library/Java/JavaVirtualMachines/temurin-21.0.2/Contents/Home/bin/java -agentli
Connected to the target VM, address: '127.0.0.1:52036', transport: 'socket'
Public Key A: 16
Public Key B: 16
Shared Secret: 1
Shared Secret the same? true
Disconnected from the target VM, address: '127.0.0.1:52036', transport: 'socket'
```

2)

```
class DiffieHellman {

    public static void main(String[] args) {
        BigInteger prime = new BigInteger( val: "21");
        BigInteger base = new BigInteger( val: "16");
        BigInteger secretA = new BigInteger( val: "145");
        BigInteger secretB = new BigInteger( val: "389");

        BigInteger publicA = base.modPow(secretA, prime);
        BigInteger publicB = base.modPow(secretB, prime);

        BigInteger sharedSecretA = publicB.modPow(secretA, prime);
        BigInteger sharedSecretB = publicA.modPow(secretB, prime);

        System.out.println("Public Key A: " + publicA);
        System.out.println("Public Key B: " + publicB);
        System.out.println("Shared Secret: " + sharedSecretA);
        System.out.println("Shared Secret the same? " + sharedSecretB.equals(sharedSecretA));
    }
}
```

```
ug  DiffieHellman
    Threads & Variables  Console
/Users/tim.herbst/Library/Java/JavaVirtualMachines/temurin-21.0.2/Contents/Home/bin/java -agentli
Connected to the target VM, address: '127.0.0.1:52128', transport: 'socket'
Public Key A: 16
Public Key B: 4
Shared Secret: 4
Shared Secret the same? true
Disconnected from the target VM, address: '127.0.0.1:52128', transport: 'socket'
```