# 💔 SSL Trust - Missing "Authority Key Identifier"

## The problem

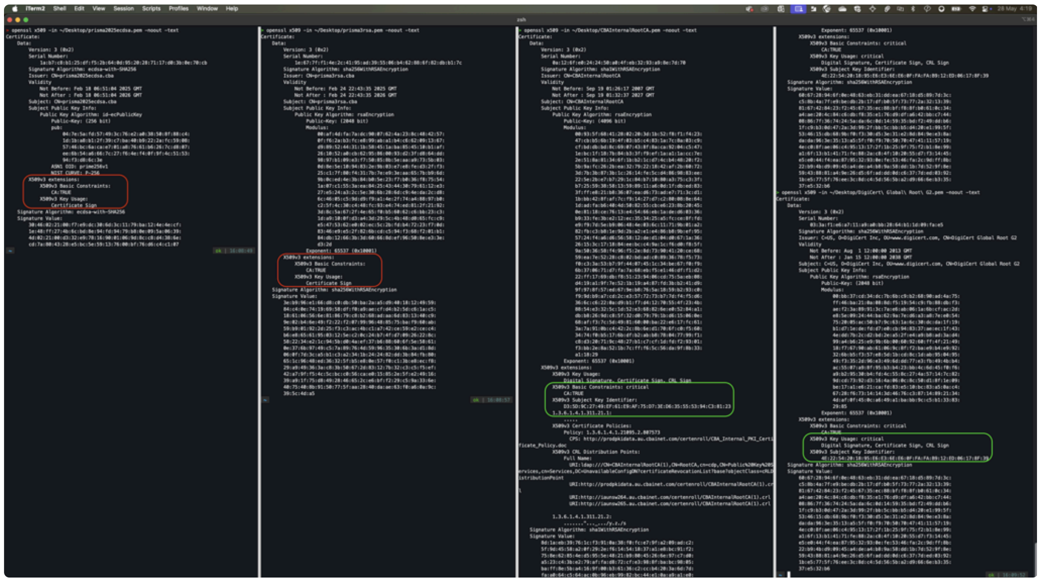
Some Engineers are experiencing issues with SSL Forward Inspections for ANY clients with `X509_STRICT` validation... For instance, `python 3.13` +

e.g. `python3 -c 'import requests; print(requests.get("https://github.com").status_code)'` would return `urllib3.exceptions.SSLError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: Missing Authority Key Identifier (_ssl.c:1028)`

The problem is **not** related to going throught a proxy or not. It is not either related to SSL certificate or whether or not "an internal Root CA is added to a pem file (CA Store)".
If your issue is related to Proxy or SSL Trust issues, see this : Proxy and SSL Certificate issues

The problem is, our SSL forward inspection Root CAs are not compliant with RFC 5280. Also we SSL Inspect most traffic, even the SSL traffic not going through proxies (This is easily verifyable with github.com which is direct (not proxied), and yet is SSL intercepted...

More specifically, both prisma3rsa and prisma2025ecdsa are missing that a x509v3 property or Subject Key Identifier as defined here : RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile



```
> openssl x509 -in ~/Desktop/prisma2025ecdsa.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            1a:b7:c8:b1:25:df:f5:2b:64:0d:95:20:28:71:17:d0:3b:0e:70:cb
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: CN=prisma2025ecdsa.cba
        Validity
            Not Before: Feb 18 06:51:04 2025 GMT
            Not After : Feb 18 06:51:04 2026 GMT
        Subject: CN=prisma2025ecdsa.cba
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:7e:5a:fd:57:49:3c:76:e2:a0:38:50:0f:88:c4:
                    1d:1b:a8:b1:2f:39:c7:ba:40:b0:22:2a:fc:59:00:
                    57:46:bc:6a:ca:e7:01:a0:76:61:b6:26:7c:d8:07:
                    ee:6b:54:a6:66:7c:27:f6:4e:f4:0f:9f:4c:51:53:
                    94:f3:d8:6c:3e
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:TRUE
            X509v3 Key Usage:
                Certificate Sign
    Signature Algorithm: ecdsa-with-SHA256
    Signature Value:
        30:46:02:21:00:f7:e9:dc:30:6d:3c:11:79:ba:12:4e:4e:cf:
        1e:48:ff:27:4b:6c:bd:0e:94:fd:94:79:b8:0e:09:5a:06:39:
        4d:02:21:00:d3:32:e9:78:16:90:01:06:9d:8c:c8:d4:30:6e:
        cd:7a:00:43:28:e5:bc:5e:59:13:76:00:bf:76:d6:c4:c1:07
~                                                        ok | 16:08:49
```

```
> openssl x509 -in ~/Desktop/prisma3rsa.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            1e:67:7f:f1:4e:2c:41:95:ad:39:55:06:b4:62:08:6f:02:db:b1:7c
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=prisma3rsa.cba
        Validity
            Not Before: Feb 24 22:43:35 2025 GMT
            Not After : Feb 24 22:43:35 2026 GMT
        Subject: CN=prisma3rsa.cba
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:af:4d:fa:7a:dc:90:07:62:4a:23:8c:48:42:57:
                    0f:f6:2a:b3:fc:e8:99:ab:a1:b4:c6:62:09:13:67:
                    d9:09:52:44:31:1b:58:45:1a:ba:85:45:10:b1:af:
                    26:10:52:a0:cb:62:95:06:00:93:d2:3f:d8:64:dd:
                    98:97:b1:09:e3:f7:50:85:0b:5e:aa:a9:73:5b:03:
                    0d:8e:5e:10:94:83:2e:9b:03:e7:e8:fe:d3:2f:f3:
                    25:c1:7f:00:f4:31:7b:7e:e9:3e:aa:65:7b:b9:6d:
                    9b:0c:ed:4e:3b:04:b0:5e:23:f7:b0:36:f8:75:54:
                    1a:07:c1:55:3a:ea:04:25:43:44:30:79:61:12:e3:
                    27:e5:24:a3:2c:5e:30:6b:28:6d:c9:4e:da:2c:d8:
                    6c:46:05:c5:9d:d9:f9:a1:4e:2f:74:a4:88:97:b0:
                    c2:5f:4c:30:c4:48:fc:93:e4:74:ed:81:2f:21:92:
                    3d:0c:5a:67:2f:4e:65:f0:b5:60:62:c6:bb:23:c3:
                    1d:a9:10:0f:d3:a4:3d:29:5c:4b:48:d0:65:fc:c9:
                    e5:47:53:62:e0:02:ec:5c:2b:fd:b4:72:23:f7:0d:
                    83:46:e9:e5:2f:82:6b:cd:c5:94:f3:68:f2:01:b1:
                    04:db:12:66:3b:3d:60:66:8d:ef:96:50:8e:e3:3e:
                    d3:2d
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:TRUE
            X509v3 Key Usage:
                Certificate Sign
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        3e:b9:96:e1:66:d8:c0:db:50:ba:2a:a5:d9:40:18:12:49:59:
        04:c4:0e:74:19:69:58:df:f0:a9:ae:cf:d4:b2:5d:c6:1a:c5:
        18:61:06:56:6e:81:06:79:c8:b2:68:ad:aa:6d:83:13:40:c9:
        9e:02:b4:6e:49:f2:22:f2:07:99:96:48:85:75:ba:f9:60:ab:
        59:b9:01:92:2d:25:f3:c3:ac:4b:c1:a7:42:ce:59:e2:ce:c4:
        b6:e8:65:61:95:03:12:5e:c2:0c:24:b7:4f:d7:09:26:22:8c:
        58:22:34:e2:1c:94:5b:d0:4a:ef:37:b6:88:60:6f:5e:58:61:
        0e:37:6b:97:49:c5:7a:09:76:4d:59:96:35:30:6b:3a:d1:0d:
        06:0f:7d:3c:a5:b1:c3:a2:34:1b:24:24:82:dd:3b:04:fb:00:
        65:1c:96:48:ed:36:32:5f:b5:e8:0e:57:f0:c1:3b:e8:ec:f8:
        29:a9:49:36:3a:c8:3b:50:67:2d:83:12:7b:32:c3:c5:f5:ef:
        42:a7:9f:f5:4c:5c:bc:c0:56:ca:e0:15:85:2e:5f:e2:49:16:
        39:a9:1f:75:d8:49:28:46:65:2c:e6:bf:f2:29:c5:9a:33:6e:
        40:75:40:0b:91:50:77:5f:aa:28:40:da:ae:63:f0:a6:0a:9c:
        39:5c:4d:a5
~                                                        ok | 16:08:57
```

```
> openssl x509 -in ~/Desktop/CBAInternalRootCA.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            0a:12:6f:e0:24:24:50:a0:4f:eb:32:93:a9:0e:7d:70
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: CN=CBAInternalRootCA
        Validity
            Not Before: Sep 19 01:26:17 2007 GMT
            Not After : Sep 19 01:32:37 2027 GMT
        Subject: CN=CBAInternalRootCA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:93:5f:68:41:28:02:20:3d:1b:52:f8:f1:f4:23:
                    47:cb:b5:6b:19:4f:03:b5:c4:83:1a:7c:0a:03:c9:
                    cf:bd:db:bd:0c:69:07:43:0f:0a:ca:92:04:c5:47:
                    1e:bc:1f:18:7b:04:b3:3f:f9:ef:1a:a1:1a:cc:7e:
                    2e:51:0a:01:34:6f:1b:b2:1c:d7:4c:b4:48:20:f2:
                    5b:9a:fc:26:2b:ea:32:79:22:18:42:af:2b:60:72:
                    3d:7b:3b:87:3b:1c:26:14:fe:5c:d4:06:98:83:ee:
                    22:5e:2b:e7:b7:29:1c:04:b7:10:00:a3:75:c3:3f:
                    b7:25:59:38:58:13:59:09:11:a6:0d:1f:db:ed:83:
                    3f:ff:e8:21:b8:36:07:ea:d6:73:ad:e7:71:3c:d1:
                    1b:bb:42:0f:af:7c:f9:14:27:d7:c2:00:08:0e:64:
                    1d:ad:fa:b6:40:4d:50:02:55:cb:e6:23:0b:20:45:
                    0e:81:18:ce:76:13:e4:54:66:eb:1a:de:d6:03:36:
                    b9:33:fe:3b:e2:12:ec:35:34:25:a5:fc:ce:8f:fd:
                    e9:f9:7d:5e:b9:06:48:4e:03:6c:11:71:9b:01:a2:
                    82:fb:c3:b8:1e:9d:2b:a2:e1:e4:06:b8:9b:ef:95:
                    57:24:f4:a6:d6:56:58:12:de:d1:04:d0:67:1a:36:
                    26:15:3c:17:18:04:ee:bc:c4:9a:1c:f6:d0:f8:5f:
                    9a:50:36:58:f4:96:f5:2e:8d:73:90:41:20:ce:68:
                    59:ea:7e:52:28:c8:02:bd:ad:c0:09:36:70:f5:73:
                    f0:c3:3a:53:b7:9f:44:07:45:1c:34:be:67:f0:f9:
                    6b:37:06:71:d7:fa:7a:68:eb:f5:e1:46:df:f1:d2:
                    22:ff:17:69:db:f8:51:23:94:06:cd:75:5a:eb:00:
                    d4:19:a1:9f:7e:52:1b:19:a4:87:fd:3b:b2:41:d9:
                    9f:97:8f:57:ed:67:9e:b8:76:5a:18:59:b2:93:c0:
                    f9:9d:b9:a7:cd:2c:e3:57:72:73:b7:7d:f4:f5:d6:
                    36:6c:c6:22:0a:d9:b1:f7:d4:12:70:55:4f:23:4b:
                    88:54:e3:32:5c:1d:52:e3:68:82:6e:e0:52:84:a1:
                    db:b8:26:9d:c8:5f:32:d0:79:79:1b:d6:15:06:0e:
                    68:af:f3:7c:5d:49:81:60:00:ba:22:e6:17:fc:61:
                    3a:7a:91:0b:c4:42:2c:8b:6e:d1:70:6f:c0:f5:60:
                    34:74:f0:b5:17:6b:df:b2:ab:b8:70:04:77:99:f1:
                    c8:d3:20:71:9c:48:27:b1:c7:cf:1d:fd:f2:93:01:
                    f3:bb:2e:8a:52:1b:7c:ff:f6:5c:56:da:9f:8b:33:
                    a1:18:29
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage:
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                D3:5D:9C:27:49:EF:61:E9:AF:75:D7:3E:D6:35:55:53:94:C3:81:23
            1.3.6.1.4.1.311.21.1:
                .....
            X509v3 Certificate Policies:
                Policy: 1.3.6.1.4.1.21095.2.007573
                  CPS: http://prodpkidata.au.cbainet.com/certenroll/CBA_Internal_PKI_Certi
ficate_Policy.doc
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:ldap:///CN=CBAInternalRootCA(1),CN=RootCA,cn=cdp,CN=Public%20Key%20S
ervices,cn=Services,DC=UnavailableConfigDN?certificateRevocationList?base?objectClass=cRLD
istributionPoint
                  URI:http://prodpkidata.au.cbainet.com/certenroll/CBAInternalRootCA(1).cr
l
                  URI:http://iaunow264.au.cbainet.com/certenroll/CBAInternalRootCA(1).crl
                  URI:http://iaunow265.au.cbainet.com/certenroll/CBAInternalRootCA(1).crl

            1.3.6.1.4.1.311.21.2:
                ........"..._..../y.z./s
    Signature Algorithm: sha1WithRSAEncryption
    Signature Value:
        8d:1a:eb:39:76:1c:f3:91:0a:38:f0:fc:e7:9f:a2:09:ad:c2:
        5f:9d:45:58:a2:0f:29:2e:f6:14:54:18:37:a1:e8:bc:91:f2:
        75:8e:62:85:4e:d5:95:5e:48:21:b9:80:45:26:6e:97:c7:d0:
        a5:23:c4:3b:e2:79:af:fa:d8:72:cf:e3:98:8f:ba:bc:98:05:
        ba:ff:8e:5b:a4:16:9f:00:b3:61:36:c2:cc:b4:20:3a:6d:7d:
        fa:a0:64:c5:64:ac:0b:96:eb:99:82:bc:44:e1:0a:a9:a1:e0:
```

```
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Subject Key Identifier:
                4E:22:54:20:18:95:E6:E3:6E:E6:0F:FA:FA:B9:12:ED:06:17:8F:39
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        60:67:28:94:6f:0e:48:63:eb:31:dd:ea:67:18:d5:89:7d:3c:
        c5:8b:4a:7f:e9:be:db:2b:17:df:b0:5f:73:77:2a:32:13:39:
        81:67:42:84:23:f2:45:67:35:ec:88:bf:f8:8f:b0:61:0c:34:
        a4:ae:20:4c:84:c6:db:f8:35:e1:76:d9:df:a6:42:bb:c7:44:
        08:86:7f:36:74:24:5a:da:6c:0d:14:59:35:bd:f2:49:dd:b6:
        1f:c9:b3:0d:47:2a:3d:99:2f:bb:5c:bb:b5:d4:20:e1:99:5f:
        53:46:15:db:68:9b:f0:f3:30:d5:3e:31:e2:8d:84:9e:e3:8a:
        da:da:96:3e:35:13:a5:5f:f0:f9:70:50:70:47:41:11:57:19:
        4e:c0:8f:ae:06:c4:95:13:17:2f:1b:25:9f:75:f2:b1:8e:99:
        a1:6f:13:b1:41:71:fe:88:2a:c8:4f:10:20:55:d7:f3:14:45:
        e5:e0:44:f4:ea:87:95:32:93:0e:fe:53:46:fa:2c:9d:ff:8b:
        22:b9:4b:d9:09:45:a4:de:a4:b8:9a:58:dd:1b:7d:52:9f:8e:
        59:43:88:81:a4:9e:26:d5:6f:ad:dd:0d:c6:37:7d:ed:03:92:
        1b:e5:77:5f:76:ee:3c:8d:c4:5d:56:5b:a2:d9:66:6e:b3:35:
        37:e5:32:b6
> openssl x509 -in ~/Desktop/DigiCert\ Global\ Root\ G2.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            03:3a:f1:e6:a7:11:a9:a0:bb:28:64:b1:1d:09:fa:e5
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
        Validity
            Not Before: Aug  1 12:00:00 2013 GMT
            Not After : Jan 15 12:00:00 2038 GMT
        Subject: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:bb:37:cd:34:dc:7b:6b:c9:b2:68:90:ad:4a:75:
                    ff:46:ba:21:0a:08:8d:f5:19:54:c9:fb:88:db:f3:
                    ae:f2:3a:89:91:3c:7a:e6:ab:06:1a:6b:cf:ac:2d:
                    e8:5e:09:24:44:ba:62:9a:7e:d6:a3:a8:7e:e0:54:
                    75:20:05:ac:50:b7:9c:63:1a:6c:30:dc:da:1f:19:
                    b1:d7:1e:de:fd:d7:e0:cb:94:83:37:ae:ec:1f:43:
                    4e:dd:7b:2c:d2:bd:2e:a5:2f:e4:a9:b8:ad:3a:d4:
                    99:a4:b6:25:e9:9b:6b:00:60:92:60:ff:4f:21:49:
                    18:f7:67:90:ab:61:06:9c:8f:f2:ba:e9:b4:e9:92:
                    32:6b:b5:f3:57:e8:5d:1b:cd:8c:1d:ab:95:04:95:
                    49:f3:35:2d:96:e3:49:6d:dd:77:e3:fb:49:4b:b4:
                    ac:55:07:a9:8f:95:b3:b4:23:bb:4c:6d:45:f0:f6:
                    a9:b2:95:30:b4:fd:4c:55:8c:27:4a:57:14:7c:82:
                    9d:cd:73:92:d3:16:4a:06:0c:8c:50:d1:8f:1e:09:
                    be:17:a1:e6:21:ca:fd:83:e5:10:bc:83:a5:0a:c4:
                    67:28:f6:73:14:14:3d:46:76:c3:87:14:89:21:34:
                    4d:af:0f:45:0c:a6:49:a1:ba:bb:9c:c5:b1:33:83:
                    29:85
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
            X509v3 Subject Key Identifier:
                4E:22:54:20:18:95:E6:E3:6E:E6:0F:FA:FA:B9:12:ED:06:17:8F:39
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        60:67:28:94:6f:0e:48:63:eb:31:dd:ea:67:18:d5:89:7d:3c:
        c5:8b:4a:7f:e9:be:db:2b:17:df:b0:5f:73:77:2a:32:13:39:
        81:67:42:84:23:f2:45:67:35:ec:88:bf:f8:8f:b0:61:0c:34:
        a4:ae:20:4c:84:c6:db:f8:35:e1:76:d9:df:a6:42:bb:c7:44:
        08:86:7f:36:74:24:5a:da:6c:0d:14:59:35:bd:f2:49:dd:b6:
        1f:c9:b3:0d:47:2a:3d:99:2f:bb:5c:bb:b5:d4:20:e1:99:5f:
        53:46:15:db:68:9b:f0:f3:30:d5:3e:31:e2:8d:84:9e:e3:8a:
        da:da:96:3e:35:13:a5:5f:f0:f9:70:50:70:47:41:11:57:19:
        4e:c0:8f:ae:06:c4:95:13:17:2f:1b:25:9f:75:f2:b1:8e:99:
        a1:6f:13:b1:41:71:fe:88:2a:c8:4f:10:20:55:d7:f3:14:45:
        e5:e0:44:f4:ea:87:95:32:93:0e:fe:53:46:fa:2c:9d:ff:8b:
        22:b9:4b:d9:09:45:a4:de:a4:b8:9a:58:dd:1b:7d:52:9f:8e:
        59:43:88:81:a4:9e:26:d5:6f:ad:dd:0d:c6:37:7d:ed:03:92:
        1b:e5:77:5f:76:ee:3c:8d:c4:5d:56:5b:a2:d9:66:6e:b3:35:
        37:e5:32:b6
~                                                        ok | 16:09:52
```

prisma3rsa.cba and prisma2025ecdsa are missing **Authority Key Identifier**
Note, CBAInternalRootCA is not affected…

The issue is only reproductible with clients or CLI using X509_STRICT validation.
So far that is only Python 3.13 and a few others but probably more to come…

❌ Now the real concern is if other clients like mainstream browsers (Chrome/Edge) perform strict x509 validation (like python 3.13), because then **all web requests** would fail…

## The solution

### Root-cause issue resolution

✅ Reissue Prisma Root CAs with the right x509 properties (**Authority Key Identifier**, and possible X509v3 Basic Constraints: "**critical**" as well): **Recommended** but the Palo Alto team will need to re-issue the RootCAs and the SOE teams (SCCM/JAMF) will need to push it to endpoint to trust it (Keychain/MMC Certs)

The issue has been escalated via multiple channels… Stay alert for an update!

- UAA-2472: Prisma Forward Trust Certificates are not compliant with Python 3.13+ OPEN
- Python 3.13 SSL Errors posted in ES-Python User Group / General Discussion on 28 May
- **Florian Bidabe on Viva Engage:** New Prisma Forward Inspection is partially broken… Posted in Global Protect on May 28, 2025 & **Florian Bidabe on Viva Engage:** We have an issue with the intercepting Root CA (prisma) - They lack x509v3 flag for Authority Key Identifier which isn't compliant with the RFC Current… Posted in Cyber Security team on May 30, 2025

### (Dirty) workarounds

#### Ignore strick validation and ssl veirication entirely

- Ignore **Strict** Checks: PAINFUL because that implies disruption and time spent on troubleshooting and finding what argument to pass or config file for a client to avoid x509 strict checks.
- Ignore SSL verification: e.g. `--no-verify` INSECURE ... likely what Engineers will resort to doing!

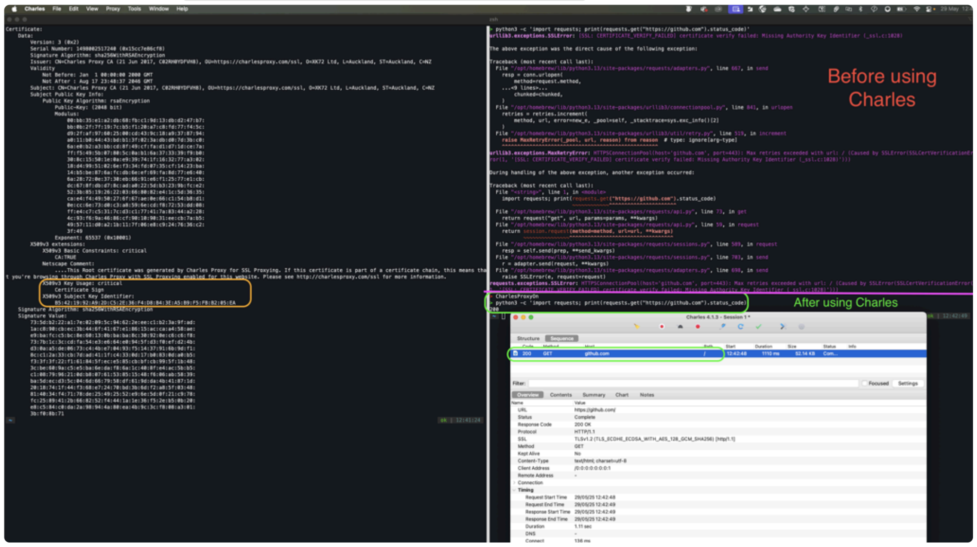
#### Downgrade or use another client

- Use older clients not doing strict validation: PAINFUL , downgrading often comes with a bunch of compatilibilty issues. Alternative clients might not exist or introduce other issues...

#### Do SSL interception yourself

##### MacOS

- Use a MITM Proxy like Charles (It’s on JAMF Self-Service and licensed) BEST TACTICAL SOLUTION but far from ideal: Web Debugging
  - Install Charles and trust its Root CA (It's on JAMF SelfService and Group licensed )
  - Pass Charles traffic to Alpaca/CNTLM ( `localhost:3128` ) or Prisma ( `cba.proxy.prismaaccess.com:8080` )
  - Set your proxy var to Charles so Python SDK knows where to go `export {all,http,https}_proxy=http://localhost:8888`
  - Set all this to auto-start or create Daemons so you don't have to run it manually everytime after reboot: Charles in the Background



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1498002517240 (0x15cc7e86cf8)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=Charles Proxy CA (21 Jun 2017, C02RH0YDFVH8), OU=https://charlesproxy.com/ssl, O=XK72 Ltd, L=Auckland, ST=Auckland, C=NZ
        Validity
            Not Before: Jan  1 00:00:00 2000 GMT
            Not After : Aug 17 23:48:37 2046 GMT
        Subject: CN=Charles Proxy CA (21 Jun 2017, C02RH0YDFVH8), OU=https://charlesproxy.com/ssl, O=XK72 Ltd, L=Auckland, ST=Auckland, C=NZ
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:bb:35:e1:a2:db:68:fb:c1:9d:13:db:d2:47:b7:
                    bb:0b:2f:7f:19:7c:b5:f1:20:a7:c8:fd:77:f4:5c:
                    d9:2f:af:97:60:25:00:cd:43:9c:18:a9:37:87:94:
                    a0:11:b0:44:43:bd:b1:3f:02:3a:db:d0:7d:3b:c0:
                    6a:e0:b2:a3:bb:cd:8f:49:cf:fa:d1:d7:1d:ce:7a:
                    ff:f5:49:5b:07:00:5c:0a:b1:6a:37:33:39:f9:b0:
                    30:8c:15:50:1e:0a:e9:39:74:1f:16:32:77:a3:02:
                    18:d4:99:51:02:6e:f3:34:fd:07:35:cf:14:23:ba:
                    14:b5:be:87:6a:fc:db:6e:ef:69:fa:8d:77:e6:40:
                    6a:28:72:0e:37:30:eb:66:91:e6:f1:25:77:e1:cb:
                    dc:67:8f:db:d7:8c:ad:a0:22:5d:b3:23:9b:fc:e2:
                    52:3b:85:19:26:22:03:66:80:82:e4:1c:5d:36:35:
                    ca:e4:f4:49:50:27:6f:67:ae:0e:66:c1:54:b8:d1:
                    0e:cc:6e:73:d0:c3:a8:59:6e:cd:f8:72:53:dd:08:
                    ff:e4:c7:c5:31:7c:d3:c1:77:41:7a:83:44:a2:28:
                    4c:93:f6:9a:46:86:cf:90:10:90:31:ee:cb:7a:b5:
                    49:57:11:d0:a2:1b:11:7f:06:e8:c9:24:76:36:c2:
                    3f:49
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            Netscape Comment:
                ....This Root certificate was generated by Charles Proxy for SSL Proxying. If this certificate is part of a certificate chain, this means tha
t you're browsing through Charles Proxy with SSL Proxying enabled for this website. Please see http://charlesproxy.com/ssl for more information.
            X509v3 Key Usage: critical
                Certificate Sign
            X509v3 Subject Key Identifier:
                85:42:19:92:A9:2D:C5:2E:36:F4:D8:B4:3E:A5:89:F5:F8:B2:05:EA
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        73:5d:b2:22:a1:7e:02:09:5c:94:62:2e:ee:c1:b2:3a:9f:ad:
        1a:c8:90:cb:ec:3b:44:6f:41:67:e1:86:15:ac:ca:a4:58:ae:
        e9:ba:fc:c5:bc:8e:60:13:0b:ba:ba:8c:30:92:0e:c6:c6:f8:
        73:7b:1c:3c:cd:fa:54:e3:e6:64:e0:94:5f:d3:f0:ef:d2:4b:
        d3:0a:a5:de:06:73:c4:4b:e7:04:93:f5:14:37:91:6b:9d:f1:
        8c:c1:2a:33:cb:7d:ad:41:1f:c4:33:0d:17:b0:83:0d:a0:b5:
        f3:3f:3f:22:f1:61:84:5f:ec:e5:85:cb:bf:cb:99:5f:1b:48:
        3c:be:60:9a:c5:e5:ba:6e:da:f8:6a:1c:40:8f:e4:ac:5b:b5:
        c1:08:79:96:21:0d:b8:07:61:53:85:15:48:f6:06:ab:58:39:
        ba:5d:ec:d3:5c:04:6d:66:79:58:df:61:9d:da:4b:41:87:1d:
        20:18:74:1f:44:f3:68:e7:24:70:bd:3b:6d:f2:a8:5f:03:48:
        81:40:34:f4:71:78:de:25:49:25:52:e9:6e:5d:0f:21:c9:78:
        fc:25:09:41:2b:66:82:52:f4:44:1a:1e:36:f5:2e:b5:0b:20:
        e8:c5:84:c0:da:2a:98:94:4a:80:ea:4b:9c:3c:f8:08:a3:01:
        3b:f0:8b:71
~                                                              ok | 12:41:24
```

```
> python3 -c 'import requests; print(requests.get("https://github.com").status_code)'
urllib3.exceptions.SSLError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: Missing Authority Key Identifier (_ssl.c:1028)

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "/opt/homebrew/lib/python3.13/site-packages/requests/adapters.py", line 667, in send
    resp = conn.urlopen(
        method=request.method,
    ...<9 lines>...
        chunked=chunked,
    )
  File "/opt/homebrew/lib/python3.13/site-packages/urllib3/connectionpool.py", line 841, in urlopen
    retries = retries.increment(
        method, url, error=new_e, _pool=self, _stacktrace=sys.exc_info()[2]
    )
  File "/opt/homebrew/lib/python3.13/site-packages/urllib3/util/retry.py", line 519, in increment
    raise MaxRetryError(_pool, url, reason) from reason  # type: ignore[arg-type]
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='github.com', port=443): Max retries exceeded with url: / (Caused by SSLError(SSLCertVerificationEr
ror(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: Missing Authority Key Identifier (_ssl.c:1028)')))

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "<string>", line 1, in <module>
    import requests; print(requests.get("https://github.com").status_code)
                           ~~~~~~~~~~~~^^^^^^^^^^^^^^^^^^^^^^
  File "/opt/homebrew/lib/python3.13/site-packages/requests/api.py", line 73, in get
    return request("get", url, params=params, **kwargs)
  File "/opt/homebrew/lib/python3.13/site-packages/requests/api.py", line 59, in request
    return session.request(method=method, url=url, **kwargs)
           ~~~~~~~~~~~~~~~^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/opt/homebrew/lib/python3.13/site-packages/requests/sessions.py", line 589, in request
    resp = self.send(prep, **send_kwargs)
  File "/opt/homebrew/lib/python3.13/site-packages/requests/sessions.py", line 703, in send
    r = adapter.send(request, **kwargs)
  File "/opt/homebrew/lib/python3.13/site-packages/requests/adapters.py", line 698, in send
    raise SSLError(e, request=request)
requests.exceptions.SSLError: HTTPSConnectionPool(host='github.com', port=443): Max retries exceeded with url: / (Caused by SSLError(SSLCertVerificationError(
1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: Missing Authority Key Identifier (_ssl.c:1028)')))
> CharlesProxyOn
> python3 -c 'import requests; print(requests.get("https://github.com").status_code)'
200
```

Charles 4.1.3 - Session 1 *

Structure | Sequence

| Code | Method | Host | Path | Start | Duration | Size | Status | Info |
|---|---|---|---|---|---|---|---|---|
| 200 | GET | github.com | / | 12:42:48 | 1110 ms | 52.14 KB | Com... | |

Filter: Focused Settings

Overview | Contents | Summary | Chart | Notes

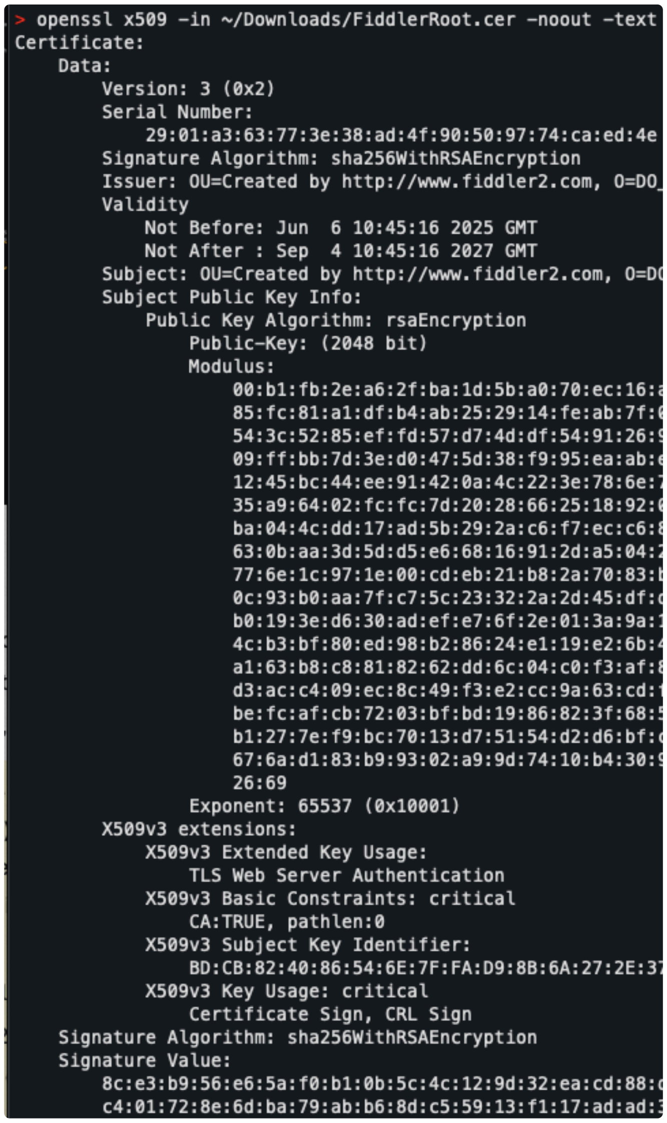
| Name | Value |
|---|---|
| URL | https://github.com/ |
| Status | Complete |
| Response Code | 200 OK |
| Protocol | HTTP/1.1 |
| SSL | TLSv1.2 (TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256) [http/1.1] |
| Method | GET |
| Kept Alive | No |
| Content-Type | text/html; charset=utf-8 |
| Client Address | /0:0:0:0:0:0:0:1 |
| Remote Address | - |
| Connection | |
| Timing | |
| Request Start Time | 29/05/25 12:42:48 |
| Request End Time | 29/05/25 12:42:49 |
| Response Start Time | 29/05/25 12:42:49 |
| Response End Time | 29/05/25 12:42:49 |
| Duration | 1.11 sec |
| DNS | - |
| Connect | 136 ms |

#### Windows

- Use a MITM Proxy like Fiddler (It’s on Software Center)
- In Options, Enable Decryption, and Trust the Root CA (requires local admin)
- On your problematic client, set the proxy to http://localhost:8888

⚠ Unfortunately, if you are a standard user, you're stuck in a tough spot. Software Center (SCCM) only offers Python 3.13 for installation. While Fiddler can be installed, enabling decryption requires local admin access on your Windows SOE. This is necessary for Fiddler to add its Signing Root CA certificate to the **System** Root certificate store to be trusted. As a result, you have a few options:

- Request local admin access to either:
  - Use Python 3.12 instead (e.g., in a virtual environment)
  - Trust the Fiddler CA so it can validate Python 3.13's RFC-compliant certificate
- Wait for the Prisma team to issue new RFC5280 compliant certificates
- Wait for the provisioning team (SCCM) to package Python 3.12 in the Software Center



```
> openssl x509 -in ~/Downloads/FiddlerRoot.cer -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            29:01:a3:63:77:3e:38:ad:4f:90:50:97:74:ca:ed:4e
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: OU=Created by http://www.fiddler2.com, O=DO_
        Validity
            Not Before: Jun  6 10:45:16 2025 GMT
            Not After : Sep  4 10:45:16 2027 GMT
        Subject: OU=Created by http://www.fiddler2.com, O=DO
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b1:fb:2e:a6:2f:ba:1d:5b:a0:70:ec:16:a
                    85:fc:81:a1:df:b4:ab:25:29:14:fe:ab:7f:0
                    54:3c:52:85:ef:fd:57:d7:4d:df:54:91:26:9
                    09:ff:bb:7d:3e:d0:47:5d:38:f9:95:ea:ab:e
                    12:45:bc:44:ee:91:42:0a:4c:22:3e:78:6e:7
                    35:a9:64:02:fc:fc:7d:20:28:66:25:18:92:0
                    ba:04:4c:dd:17:ad:5b:29:2a:c6:f7:ec:c6:8
                    63:0b:aa:3d:5d:d5:e6:68:16:91:2d:a5:04:2
                    77:6e:1c:97:1e:00:cd:eb:21:b8:2a:70:83:b
                    0c:93:b0:aa:7f:c7:5c:23:32:2a:2d:45:df:c
                    b0:19:3e:d6:30:ad:ef:e7:6f:2e:01:3a:9a:1
                    4c:b3:bf:80:ed:98:b2:86:24:e1:19:e2:6b:4
                    a1:63:b8:c8:81:82:62:dd:6c:04:c0:f3:af:8
                    d3:ac:c4:09:ec:8c:49:f3:e2:cc:9a:63:cd:1
                    be:fc:af:cb:72:03:bf:bd:19:86:82:3f:68:5
                    b1:27:7e:f9:bc:70:13:d7:51:54:d2:d6:bf:c
                    67:6a:d1:83:b9:93:02:a9:9d:74:10:b4:30:9
                    26:69
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Subject Key Identifier:
                BD:CB:82:40:86:54:6E:7F:FA:D9:8B:6A:27:2E:37
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        8c:e3:b9:56:e6:5a:f0:b1:0b:5c:4c:12:9d:32:ea:cd:88:c
        c4:01:72:8e:6d:ba:79:ab:b6:8d:c5:59:13:f1:17:ad:ad:3
```

As you can see, a Fiddle Root CA cert is also compliant.
It contains the X509 properties Strict validation requires.
But you need to be able to trust it in your system with your _adm

*Helpful? Drop me a thanks on [Achievers](#) ! And if you've got knowledge to share, don't hold back - we all grow when we learn from each other* 💡