Ransomware Reconnaissance: Interrogating Certificates Towards Proactive Threat Mitigation

RUDD, Steph

Published version

RUDD, Steph (2024). Ransomware Reconnaissance: Interrogating Certificates Towards Proactive Threat Mitigation. In: KOBUSINSKA, Anna, JACOBSSON, Andreas and CHANG, Victor, (eds.) Proceedings of the 9th International Conference on Internet of Things, Big Data and Security IoTBDS. SCITEPRESS - Science and Technology Publications, 97-106. [Book Section]

Ransomware Reconnaissance: Interrogating Certificates Towards Proactive Threat Mitigation

Steph Rudd[a]
Centre of Excellence in Terrorism, Resilience, Intelligence and Organised Crime Research, CENTRIC,
Sheffield Hallam University, Sheffield, U.K.

Keywords: Security, Certificates, X.509, SSL, TLS, Ransomware, Remote Interrogation, CA, PKI, OpenSSL.

Abstract: "Got Root?" Presented herewith is an innovative approach to ransomware defence by interrogating the security certificate chain pertaining to modern website security. It is a proactive strategy to scrutinise the online resources prior to download for assessment of likelihood that ransomware may be present as a result of inconsistencies between the URL and its security certificate. OpenSSL is employed for interrogating certificate attributes, including characteristics such as domain mismatch and revocation status, through the systematic approach of certificate retrieval, parsing and validation. Whilst not a 'silver bullet solution' to the wider realm of ransomware attacks, this study presents a nuanced approach to suspicion detected under certificate-related vulnerabilities at a preemptive and reconnaissance stage of hazard - a necessary basis for any subsequent cyber security investigation.

1 INTRODUCTION

An investigation into the inspection of SSL/TLS X.509 security certificates towards ascertaining the possibility and probability of ransomware opportunities - and subsequent proposals towards safeguarding.

1.1 Context and Motivation

With ransomware becoming increasingly widespread, the study places focus on the pivotal role of SSL/TLS certificates in protecting information. Errors in these certificates can lead to major security breaches, yet they can be remotely analysed quickly and with reliable results. The urgency to fortify digital systems against such threats forms the core motivation of our study. Challenging the conventional reliance on tools such as Certbot as trusted Certificate Authorities (CA), in the context of ransomware exploit opportunities, provided insights into the areas requiring future safeguarding, and proposed recommendations of improvements beyond the remit of this study.

1.2 Novelties and Contributions

Listed non exhaustively are the key observations deduced from the study:

- **Overlooked Vulnerabilities in Certbot.** Whilst the trusted CA, Certbot, provides robust and free security certificates, this study presents evidence that critical misconfigurations such as lacking integrity in certificate chains, and revoked certificates, seemingly valid, are subject to manipulation.
- **Browser Validation Limitations.** Proof that browsers cannot detect incomplete certificate chains or revoked certificates, advantageous for ransomware deception
- **Alternative Subject Analysis.** Inappropriate certificate labelling also evades detection, allowing malicious domains to appear trustworthy.
- **Revocation.** Revoked certificates can remain active yet undetected, allowing loopholes. Error testing: Positioning of context between SSL/TLS vulnerabilities against the advance and prevention of ransomware opportunities.

[a] https://orcid.org/0009-0004-8503-373X

Rudd, S.
Ransomware Reconnaissance: Interrogating Certificates Towards Proactive Threat Mitigation.
DOI: 10.5220/0012710600003705

In *Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS 2024)*, pages 97-106
ISBN: 978-989-758-699-6; ISSN: 2184-4976

1.3 Structure

The balance of this paper describes the background problem, introducing the significance of ransomware as a topical cyber threat, subsequently detailing pitfalls of certificate vulnerabilities leading to attack opportunities. The most concerning pitfalls are carried forward to testing, in which live nested subdomains with deliberate errors are deployed using a trusted CA and tested as they would be in industry. Conclusions are drawn from these concerns and future work is proposed to mitigate vulnerabilities discovered.

2 PROBLEM BACKGROUND

Ransomware, a rapidly evolving cyber threat, leverages various attack vectors to infiltrate systems, encrypt data, and demand ransom. A critical and often underestimated avenue for these attacks is the exploitation of X.509 certificate vulnerabilities. The security of these certificates is paramount as they authenticate and secure communications for various sensitive online processes such as finance, healthcare and other personal or sensitive data. It is a major concern that vulnerabilities such as expired certificates, weak encryption algorithms, or misconfigurations can provide a conduit for ransomware attacks. Such vulnerabilities not only weaken encryption, but also erode trust mechanisms as a holistic entity - those integral to digital security.

A further challenge towards mitigating ransomware risks associated with certificate vulnerabilities is user naivety. The average user often lacks the expertise to understand the nuances surrounding Transport Layer Security (TLS) security in general - let alone certificate inspection within or even beyond the browser interface. Ignorance of browser warning regarding safety of expired and untrusted certificates however is fairly obvious. The type of person who would happily disregard the safety of a browser on their private device is unlikely to possess the wherewithal to proceed beyond that warning, as the process is deliberately obfuscated - and most organisational firewalls will prevent that proceeding for users who can. More poignantly is the fact that such vulnerabilities are permitted to launch in the first instance - and these are the errors on which this study is based.

How possible is it to manipulate, mimic, and mock a trusted CA? Having achieved a number of deliberate errors into a server, how well can they be hidden, by the casual browser, or by closer inspection of a designated tool such as OpenSSL (Tuleuov and Ospanova, 2024)?

The answers to these questions will inform us of the realities of certificate vulnerabilities, but more importantly bring concrete indicators to the next generation of CA infrastructures and safety requirements.

3 VULNERABILITIES

Presented below are details of the tool used for testing, and which vulnerabilities are suitable for testing given the behaviour and automated safety settings of such.

3.1 Related Work

There are several recent studies in detection of malicious software in IoT (Tamás et al., 2021), Machine Learning (ML) neural models (Obetta and Moldovan, 2023), Artificial Intelligence (AI) (Lu and Thing, 2022), and advanced malicious binaries existing within system files (Chukka and Devi, 2021), there is little coverage on direct and simplistic preventative tactics such as addressing publicly-available security information directly. Whilst such advanced technologies are indeed necessary and with great success, this study places focus on how vulnerabilities in remote servers can indicate the possibilities of malicious attempt - thus providing advanced warning, and ideally, preventing the ransomware connection.

3.2 Obvious Indications

Many certificate vulnerabilities will flag in the browser automatically, warning them that the site is insecure and creating an extra layer of difficulty to proceed any further. This is often enough to scare users from venturing any further, and therefore certificate vulnerabilities that provide this function need not be considered. For less obvious certificate vulnerabilities, ie, those that do not trigger such warnings, interaction with Certbot is considered. Certbot is a trusted and open-source Certificate Authority (CA), used to issue X.509 certificates, ensuring safety throughout. Ideally, Certbot will prevent any artificial vulnerability implementation, representative of resistance against ransomware.

3.3 Potential Pitfalls

These are the multiple vulnerabilities pertaining to X.509 certificates (Zulfiqar et al., 2022), and their underlying centralised management system Public Key Infrastructure (PKI), (Vlad et al., ; Housley and Polk, 2001):

1. **Expired Certificates.** Expired certificates, or very short validity periods, cannot be set in Certbot, but auto-renewal can be disabled to create the same in time. Expired certificates, while not directly linked to ransomware, raise red flags in browsers, and therefore require no further consideration.
2. **Short Key Lengths.** Certificates with short cryptographic key lengths are more vulnerable to brute-force attacks, but fortunately, Certbot does not support modification of key lengths - and so they will always abide by the latest TLS standard.
3. **Misconfigured Certificates.** Misconfigurations related to the setup of the certificate (such as enabled cipher suites and protocols) can be examined using OpenSSL . This is a broad turn of phrase, largely referring to the misconfiguration of cipher suites and SSL or TLS protocols, which could potentially be modified in the generated configuration files after certificate issuance.
4. **Self-Signed Certificates.** Certificates generated without a CA flag up on browsers and are amongst the most obvious of all dangerous website indicators. Certbot is a trusted CA, and thus does not produce self-signed certificates.
5. **Revoked Certificates.** These are valid certificates that have been withdrawn for whatever reason. If they remain unidentified, they can be exploited for ransomware attacks by mimicking trustworthy sites and misleading users into downloading malicious content. This is an important vulnerability to investigate using Certbot.
6. **Domain Mismatch Certificates.** Generate certificates where the Common Name (CN) or Subject Alternative Name (SAN) doesn't match the subdomain. Although not a direct vector for ransomware, it can often play a part in sophisticated phishing schemes, leveraging the naivety of the user, and often opens the possibilities of Man-in-the-Middle (MitM) attacks.
7. **Certificates with Invalid Signatures.** These are certificates where the signature verification fails, indicating potential tampering or corruption, but will also flag immediately in the browser.
8. **Certificates with Incorrect Usage Flags.** Certificates that are not flagged correctly for their intended use, such as a certificate meant for securing email being used for a web server.
9. **Wildcard Certificates for Suspicious Domains.** While wildcard certificates are legitimate, their use in certain contexts (like a brand-new or low-reputation domains) can be suspicious. Certbot insists on domain validation within the DNS records for issuance of such certificates, and will therefore always be valid on completion.
10. **Certificates with Very Long Validity Periods.** Abnormally long validity periods might be a sign of non-compliance with best practices or an attempt to avoid frequent renewal scrutiny, which is why Certbot will provide a set 90-day validity period with auto-renewal so this cannot occur.
11. **Certificates with Incomplete Chain of Trust.** These certificates do not have a complete path to a trusted root CA, often because of missing intermediate certificates. This is a vulnerability to be tested, as modification of configuration files may be available using certificate paths.
12. **Certificates with Suspicious Subject Information.** This includes certificates with vague or misleading information in the subject field, like generic names or placeholder details. Certbot populates these details automatically from within the domain path and therefore will not occur.
13. **Overly Broad Subject Alternative Names (SANs).** Certificates that cover an unusually wide range of domain names or include unrelated domains can be suspicious. This will be investigated amongst other nested subdomains, subdomains, or domains using configured DNS 'A' records.
14. **Certificates Issued in the Future.** Certificates with a 'Not Before' date set in the future can be indicative of system misconfigurations or malicious intent. Certbot will not allow this behaviour as auto-renew is available for the 90-day validity period.

3.4 Concerns to Be Carried Forward to Testing

Whilst it is understood that browsers will flag up the most obvious of certificate non-compliance and suspicion, and Certbot can prevent the majority of modifications towards certificate detriment, there still remain a small number of item in abeyance:

1. **Incomplete Certificate Chains.** Maintaining a chain can be disturbed such as from changes in the CA's intermediate certificates.
2. **Subject Alternative Names (SANs).** Unrelated subjects may be legitimate, but can pose risks if mismanaged.
3. **Deprecated Cipher Suites.** Default configurations may be subject to modification or change over a long period of time or through auto-renewal using the same parameters.

4. **Deprecated Protocols.** As with cipher suites.
5. **Domain Mismatch.** The configuration of server files are beyond the remit of Certbot's direct control, presenting a challenge in consistency between certificate and server setup.
6. **Revocation.** The discovery of revocation may or may not be detected, depending on additional infrastructures and recognition, or as intermediaries change over time.

4 TESTING METHODOLOGY AND RESULTS

Firstly, risk rankings are assigned by severity, to the attributes which will undertake testing - followed by an introduction to an appropriate testing setup, and how to invoke such to effectively make use of OpenSSL in a native environment from remote and secure access to the server. The errors will then be implemented onto separate websites, and interrogated remotely using OpenSSL. This will demonstrate how publicly available security information can return low, medium, high and critical vulnerability indicators.

4.1 Assign Severity Ratings

It is understood within the wider consensus of security practice, that assigning severity scores to vulnerabilities is an objective method of user-oriented decision making (Rudd and Cunningham, 2022). Generally, low-medium warnings are acceptable, and high-critical warnings are not. With reference to the risk rating methodology by the Open Web Application Security Project ((OWASP),), the assignments are presented as:

1. **Low.** Minor risks, negligible direct exploitation threat.
2. **Medium.** Moderate risks, conditional exploitation potential.
3. **High.** Significant risks, high exploitation likelihood.
4. **Critical.** Severe risks, immediate and widespread damage potential.

1. **ICC, Incomplete Certificate Chain.** This error is usually rated with low to medium severity and occurs when a server does not provide the full chain of certificates. It can lead to trust issues but is less directly related to severe attacks like ransomware. It can also be indicative of poor security practices - a low warning, rated 2.
2. **SAN, Overly Broad of Inappropriate Subject Alternative Names (SANs).** Involves certificates with subject alternative names that don't align with the domain's typical use. Rated medium, in severity, this error could indicate broader security governance issues and open vectors for attacks like phishing, a common ransomware delivery method, rated 2.
3. **WCS, Using Weak Cipher Suites.** Occurs when a server supports outdated or weak cipher suites, making encrypted communications vulnerable to interception, and therefore a critical concern since it can lead to data breaches or credential theft, rated 4.
4. **ODP, Outdated SSL/TLS Protocols.** Employing deprecated protocols like TLS 1.0 or SSLv3 significantly increases the risk of cyber attacks, including ransomware incidents, as protocols contain known vulnerabilities that can be exploited by attackers, thus marked as critical, 4.
5. **DMM, Domain Mismatch.** Rated high in severity, domain mismatches can indicate a Man-in-the-Middle attack, often used for initial reconnaissance, infiltration, or delivering ransomware - a prerequisite to attack, rated 3.
6. **RVC, Revoked Certificates.** Also a high-severity issue, a revoked certificate often signals a security compromise. If a certificate has been revoked due to key compromise, it could potentially be used by attackers to impersonate legitimate services, leading to ransomware deployment, rated 4.

4.2 Setup the Environment

To test and analyse the six deliberate SSL/TLS certificate errors (ICC, SAN, WCS, ODP, DMM, RVC), an environment was established encompassing nested subdomains on a Linux Virtual Private Server (VPS), to accommodate the Linux-native issuance of certificates, by entering the remote server using the Secure Shell (SSH), through a Linux terminal Command Line Interface (CLI).

1. **Nested Subdomains for Browser View.** Each error type is assigned a unique nested subdomain, allowing for isolated testing and clear visualisation in a web browser. This structure facilitates the demonstration of how browsers react to each specific SSL/TLS error.
2. **Linux Environment for OpenSSL.** OpenSSL commands are used to simulate and test the certificate errors, providing a robust platform for manipulation and analysis.

3. **Creation of Certificates for Each Nested Subdomain.** Individual SSL/TLS certificates are created for each nested subdomain using Let's Encrypt's Certbot. These certificates are issued from a recognised CA, and then modified to introduce the specific errors.
4. **Creation of Each of the Six Errors.** Individually created post-certificate issue:
   (a) **ICC (Incomplete Certificate Chain).** Configured by omitting intermediate certificates from the server's certificate chain, leading to trust warnings in browsers.
   (b) **SAN (Overly Broad or Inappropriate Subject Alternative Names).** Implemented by generating certificates with SANs that do not match the intended domain's typical usage, potentially causing validation issues.
   (c) **WCS (Using Weak Cipher Suites).** Established by configuring the server to prefer weak or deprecated cipher suites, detectable through SSL/TLS handshake analysis.
   (d) **ODP (Outdated SSL/TLS Protocols).** Enacted by setting the server to use outdated protocols like TLS 1.0 or SSLv3, exposing known vulnerabilities.
   (e) **DMM (Domain Mismatch).** Achieved by using a certificate issued for one domain on a different subdomain, leading to mismatch errors in browsers.
   (f) **RVC (Revoked Certificates).** Introduced by revoking a certificate post-issuance, then verifying its status using OCSP or CRL checks.

The Linux, Apache, MySQL and PHP (LAMP) stack was setup on the global server to accommodate the dynamic websites, and the following files were added for deployment:

- **DNS 'A' Record.** Added to the root domain in the format of 'subsubdomain.subdomain.domain.co.uk'.
- **/var/www.** Folder for mirrorXX.ransomrecon.domain.co.uk, containing index.php so to distinguish which error
- **/etc/apache2/sites-available.** mirrorXX.ransomrecon.domain.co.uk.conf, containing the VirtualHost port 80 (insecure HTTP), server name, document root, directory settings, log paths, and SSL certificate paths, which will automatically populate in their own conf file once they are installed.
- **/etc/apache2/sites-enabled.** These are symbolic links that automatically populate once the nested subdomains are deployed using the command 'a2ensite mirrorXX.ransomrecon.domain.co.uk.conf', followed by restarting the server using 'systemctl apache2 restart'. Such commands require sudo or admin privileges.

4.3 Implementation of Specific Error Types

Following issuance of a valid certificate on each nested subdomain, each will contain the entire certificate chain (fullchain.pem), the servers own certificate for that specific nested subdomain (cert.pem), and the private key (privkey.pem) of the server, or nested subdomain, which is used to decrypt information encoded using the server's public key and should be kept secret from everything.

4.3.1 Issue the Only X.509 Certificate that Will Remain Valid

Issue and retain one valid, error-free X.509 certificate to use a benchmark for errors.

```
~$ sudo certbot --apache -d
  mirror01.ransomrecon.domain.co.uk
~$ sudo systemctl restart apache2
~$ sudo openssl s_client -connect
mirror01.ransomrecon.domain.co.uk:443
-servername
mirror01.ransomrecon.domain.co.uk
```

4.3.2 ICC Incomplete Certificate Chain: Severity 2

Configure the server to utilise the certificate only (cert.pem), as opposed to the complete chain (fullchain.pem), thus negating the integrity throughout. This can appear valid from the browser, but should show issues on deep inspection accessible using OpenSSL.

1. **Issue Certificate, Restart the Server, Validate the Certificate on Port 443, HTTPS:**

```
~$ sudo certbot --apache -d
mirror02.ransomrecon.domain.co.uk
~$ sudo systemctl restart apache2
~$ sudo openssl s_client -connect
mirror02.ransomrecon.domain.co.uk:443
-servername
mirror02.ransomrecon.domain.co.uk
```

2. **Identify the Certificate File Name:**

```
~$  nano /etc/sites-available/mirror02.
ransomrecon.domain.co.uk-le-ssl.conf
```

3. **Modify the File to Reflect the Path of Cert.Pem and Restart the Server - Change From:**

```
SSLCertificateFile /etc/letsencrypt/
live/mirror02.ransomrecon.domain.
co.uk/fullchain.pem
```

to:

```
SSLCertificateFile /etc/letsencrypt/
live/mirror02.ransomrecon.domain.
co.uk/cert.pem
~$ sudo systemctl restart apache2
```

4. **Test the Configuration:**

```
~$ sudo openssl s\_client -connect
mirror02.ransomrecon.domain.co.uk:443
-servername
mirror02.ransomrecon.domain.co.uk
```

4.3.3 SAN Overly Broad of Inappropriate Subject Alternative Names (SANs): Severity 2

Request a certificate with an extra SAN, a domain beyond the remit of the nested subdomain.

1. **Issue Certificate, Restart the Server, Validate the Certificate, as Before.**
2. **Request Certificates From Two Domains, One the Mirror, and One Unrelated:**

```
~$  sudo certbot certonly --webroot
-w /var/www/subdomain.unrelated-
domain.co.uk -d mirror03.
ransomrecon.domain.co.uk -d
geocash.domain.co.uk
```

(choose option 2 for extend domains)

3. **List Certificates to View SANs:**

```
~$  sudo certbot certificates
```

4.3.4 WCS Using Weak Cipher Suites: Severity 4

Change the configuration file to use deprecated cipher suites such as MD5 rather than SHA256.

1. **Issue Certificate, Restart the Server, Validate the Certificate, as Before.**
2. **Change the Certbot-Generated Configuration File to Include Weak Cipher Suites:** HIGH, MEDIUM, LOW will allow cipher suites of such varying strengths, whilst an exclamation mark will disallow the subsequent attribute; !aNULL will disallow anonymous authentication. The weaker suites can then be queried in OpenSSL.

```
~$  nano /etc/apache2/sites-available/
mirror04.ransomrecon.domain.co.uk
-le-ssl.conf
```

To:

```
~$  SSLCipherSuite
HIGH:MEDIUM:LOW!aNULL
```

3. **Test the Configuration:**

```
~$  openssl s_client -connect
mirror04.ransomrecon.domain.
co.uk:443 -cipher MD5
```

4.3.5 ODP Outdated SSL/TLS Protocols: Severity 4

Change the configuration file to use deprecated SSL/TLS protocols such as SSL v3 or TLS v1.0.

1. **Issue Certificate, Restart the Server, Validate the Certificate, as Before.**
2. **Change the Certbot-Generated Configuration File to Include Weak Cipher Suites:**

```
~$  SSLProtocol all -SSLv3 -TLSv1 - TLSv1.1
```

3. **Test the Configuration:**

```
~$  openssl s_client -connect mirror05.
ransomrecon.domain.co.uk:443 -ssl3
~$  openssl s_client -connect mirror05.
ransomrecon.domain.co.uk:443 -tls1
~$  openssl s_client -connect mirror05.
ransomrecon.domain.co.uk:443 -tls1_1
```

4.3.6 DMM Domain Mismatch: Severity 3

Obtain a valid certificate for a specific nested subdomain, then apply it to a separate subdomain. In this scenario, mirror05 is valid, and used as support to error test mirror06, which is invalid.

1. **Issue Certificate, Restart the Server, Validate the Certificate, as Before.**

2. **Repeat the Process for Another Nested Subdomain, mirror06.**
3. **Reconfigure the le-ssl.conf File for mirror06 to Point to the Paths for mirror05:**

```
~$ sudo nano mirror06.ransomrecon.domain
.co.uk-le-ssl.conf
```

From:

```
SSLCertificateFile /etc/letsencrypt/
live/mirror06.ransomrecon.domain.co.uk/
fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/
live/mirror06.ransomrecon.domain.co.uk/
privkey.pem
```

To:

```
SSLCertificateFile /etc/letsencrypt/
live/mirror05.ransomrecon.domain.co.uk/
fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/
live/mirror05.ransomrecon.domain.co.uk/
privkey.pem
```

4. **Test the Configuration:**

```
~$ sudo openssl s_client -connect
mirror06.ransomrecon.domain.co.uk:443
-servername mirror06.ransomrecon.
domain.co.uk
```

4.3.7 RVC Revoked Certificates: Severity 4

Obtain a valid certificate and then revoke it. The certificate should initially appear valid but should show as revoked upon checking its revocation status.

1. **Issue Certificate, Restart the Server, Validate the Certificate, as Before.**
2. **Locate the Certificate and Full Chain Paths From Checking the Certificate-Generated Configuration File Using Global Regular Expression Print (GREP) Command, Where:**
   - **-R or -r:** Recursively search files in the specified directory and all sub directories.
   - **-i:** Ignore case distinctions in both the pattern and the input files.
   - **-l:** (Lowercase 'L') List only the names of files with matching lines, once for each file.

```
~$  grep -Ril "SSLCertificateFile.
*mirror07.ransomrecon.domain.co.uk"
/etc/apache2/sites-available/
```

3. **Revoke cert.pem and Validate the Certificate:**

```
~$ certbot revoke --cert-path
/etc/letsencrypt/live/mirror07.
ransomrecon.domain.co.uk/cert.pem
~$ sudo openssl s_client -connect
mirror07.ransomrecon.domain.co.uk:443
-servername mirror07.ransomrecon.
domain.co.uk

Congratulations! You have
successfully revoked the certificate
that was located at /etc/letsencrypt/
live/mirror07.ransomrecon.domain.co.uk/
cert.pem.
```

4. **Revoke fullchain.pem and Validate the Certificate:**

```
~$ certbot revoke - -cert-path
/etc/letsencrypt/live/mirror08.
ransomrecon.domain.co.uk/fullchain.pem
~$ sudo openssl s_client -connect
mirror08.ransomrecon.domain.co.uk:443
-servername mirror08.ransomrecon.
domain.co.uk

Congratulations! You have
successfully revoked the certificate
that was located at /etc/letsencrypt/
live/mirror08.ransomrecon.domain.co.uk/
fullchain.pem.
```

5 DISCUSSION

We present the findings from the only valid certificate, and the six deliberate errors, summarised below initially, and subsequently expanded:

1. **ICC, Incomplete Certificate Chain.** Medium severity. Showed valid on the browser, detected issues.
2. **SAN, Overly broad of inappropriate Subject Alternative Names (SANs).** Medium severity. Showed valid on the browser, detected issues.
3. **WCS, Using Weak Cipher Suites.** Critical severity. Seemingly possible by modification, but was not.
4. **ODP, Outdated SSL/TLS Protocols.** High severity. Seemingly possible by modification, but was not.
5. **DMM, Domain Mismatch.** High severity. Flagged as a browser safety error, but no issues detected.
6. **RVC, Revoked Certificates.** Critical Severity. Showed valid on the browser, detected issues.

5.1 Issue the Only X.509 Certificate that Will Remain Valid

Various details about the certificate and the TLS connection, the output summary is:

1. **Certificate Issuer and Subject Information.** The issuer details of the certificate, typically Let's Encrypt for a Certbot-issued certificate, and subject information, including the domain name for which

the certificate was issued - in this case, mirror01.ransomrecon.domain.co.uk

2. **Certificate Validity Dates.** The date from which the certificate is valid, expiration date of the certificate.
3. **Public Key Information.** Type of key; RSA, ECDSA etc, length; 2048 bits, and the public key itself in a human-readable format.
4. **Certificate Chain:** List of certificates in the chain, from the server certificate up to the root certificate, and validation status of the chain.
5. **SSL/TLS Version.** SSL or TLS version used for the connection; generally TLSv1.2 or TLSv1.3.
6. **Cipher Suite.** The cipher suite used for establishing the secure connection; more recently ECDHE-RSA-AES256-GCM-SHA384.
7. **Session Data.** Session ID, if any, and key exchange, encryption, and hash algorithms used in the session.
8. **Handshake and Encryption Details.** SSL/TLS handshake process, and any encryption parameters that were negotiated during the handshake.
9. **Verification Outcome.** Success or failure of the SSL/TLS certificate, details about any errors encountered during the verification process.
10. **Additional SSL/TLS Information.** Any extensions or additional features supported in the SSL/TLS protocol.

This is how a certificate output should look. It is a confirmation that Certbot has set up the attributes and completed the handshake successfully, and by visiting the nested subdomain URL, it can be observed that the HTTPS status is present and details of the browser security is provided.

5.2 Errors 1-6

Presented below are explorations into the results of deliberate certificate errors 1-6:

5.2.1 ICC Incomplete Certificate Chain: Severity 2

There are errors indicating problems with the SSL certificate chain, cited in the output file content:

1. **Error 20.** Unable to get the local issuer certificate: This error occurs when OpenSSL cannot find the intermediate or root certificate necessary to validate the chain of trust. It suggests that the server may not be correctly configured to send the intermediate certificates, which are essential for the client to validate the server's SSL certificate.
2. **Error 21.** Unable to verify the first certificate: This error is related to Error 20 and occurs when the SSL client is unable to verify the server's certificate in the absence of the necessary intermediate certificates.

OpenSSL has correctly ascertained the certificate issuer and first certificate to be questionable. The trust mechanism is compromised, contributing to weakness and misdirection of an original domain location.

5.2.2 SAN Overly Vroad of Inappropriate Subject Alternative Names (SANs): Severity 2

OpenSSL discovered a SAN entry for the unrelated domain, and so the script was successful. The browser did not show any indication of questionable trust, and the HTTPS with verification details were sound. As a result, the conclusion was that even though the site appeared to be valid through the URL's browser, there was potential for vulnerabilities and issues pertaining to integrity and trustworthiness:

```
- - - - - - - - - - - - - - - - - - - - - -
Certificate Name:  mirror03.ransomrecon.
domain.co.uk
Serial Number:
4c23fc4e840ded31893085c20de0e34553e
Key Type:  RSA
Domains:  mirror03.ransomrecon.domain.
co.uk  subdomain.unrelated-domain.co.uk
Expiry Date: 2024-04-17 15:49:53+00:00
(VALID: 89 days)
Certificate Path: /etc/letsencrypt/live/
mirror03.ransomrecon.domain.co.uk/
fullchain.pem
Private Key Path: /etc/letsencrypt/live/
mirror03.ransomrecon.domain.co.uk/
privkey.pem
- - - - - - - - - - - - - - - - - - - - - -
```

5.2.3 WCS Using Weak Cipher Suites: Severity 4

Upon OpenSSL interrogation, there was no report of weak cipher suites in use. It was discovered that Certbot does not allow modification towards deprecated and weak ciphers in any way, and even for educational or testing purposes that this safety feature cannot be overridden - which is very reassuring, considering this would otherwise be a critical-severity vulnerability.

5.2.4 ODP Outdated SSL/TLS Protocols: Severity 4

Exactly the same as with the cipher suite selection, upon OpenSSL interrogation, there was no report of weak SSL/TLS protocols in use. It was discovered that Certbot does not allow modification towards deprecated and weak protocols in any way, and even for educational or testing purposes that this safety feature cannot be overridden - which is very reassuring, considering this would also be a critical-severity vulnerability.

5.2.5 DMM Domain Mismatch: Severity 3

Although the browser will render a safety issue as with self-signed certificates, querying sudo certbot certificates will display the certificate as valid as they haven't been overwritten when the path within the VirtualHost was overwritten:

```
- - - - - - - - - - - - - - - - - - - - -
Certificate Name: mirror06.ransomrecon.
domain.co.uk
Serial Number:
33055c210727eab94d1162d4c6f217f4ec5
Key Type: RSA
Domains: mirror06.ransomrecon.domain.co.uk
Expiry Date: 2024-04-17 17:03:32+00:00
(VALID: 89 days)
Certificate Path: /etc/letsencrypt/
live/mirror05.ransomrecon.domain.co.uk/
fullchain.pem
Private Key Path: /etc/letsencrypt/live/
mirror05.ransomrecon.domain.co.uk/privkey.pem
- - - - - - - - - - - - - - - - - - - - -
```

5.2.6 RVC Revoked Certificates: Severity 4

The output does not contain any information indicating that cert.pem for mirror07.ransomrecon.domain.co.uk has been revoked. The SSL handshake is successful, and the certificate appears to be valid. On browser visit to the URL, it also appears secure and valid. The output does not contain any information indicating that fullchain.pem for mirror08.ransomrecon.domain.co.uk has been revoked. The SSL handshake is successful, and the certificate appears to be valid. On browser visit to the URL, it also appears secure and valid.

A revoked certificate often indicates a security compromise. If a certificate has been revoked due to key compromise, it could potentially be used by an attacker to impersonate a legitimate service, leading to ransomware deployment. It is a great concern that an in-built error such as certificate revocation with a severity rating of 4, or critical, can so easily be left unnoticed by browser security features, even OpenSSL interrogation. Whilst Let's Encrypt's Certbot is a trusted, and largely robust implementation of free and secure certificates, this is the one example of where it shows considerable, critical vulnerability. Towards addressing this concern, Online Certificate Status Protocol (OCSP), and a Certificate Revocation List (CRL), should be installed globally within the subdomain to impact on nested subdomains, and within the root domain to impact on nested subdomains, subdomains, and the domain itself.

5.3 Results Summary

Of the six original errors subject to testing, four were permitted by modification of Certbot-generated configuration files or otherwise interference via OpenSSL. Reassuringly, tampering with item WCS and ODP, cipher suites and protocols respectively), were blocked by Certbot infrastructure. DMM was also seemingly available to manipulate, but then flagged as the same safety warning as self-signed certificates, and so the user was protected from that. However, there were three permitted issues, overlooked by Certbot, and of various severity and detection via OpenSSL:

1. **ICC, Incomplete Certificate Chain.** Medium severity, and unflagged from the browser. Upon closer inspection from OpenSSl, the output showed two errors proving the valid status was false.
2. **SAN, Overly broad of inappropriate Subject Alternative Names (SANs).** Medium severity. Showed valid on the browser, and OpenSSL detected no issues displaying two very different branches of subdomains. Not even nested subdomains; one was nested (the mirror), and one was not - so they were not even on the same hierarchical level. **RVC, Revoked Certificates.** Critical Severity. This was very concerning, as the revoked certificate did not flag up either on the browser nor the OpenSSL query, even though we have written evidence that the certificate was revoked - proving that X.509 certificate are a liability for attacks pertaining to ransomware, or even support the distribution of ransomware itself.

6 CONCLUSION AND FURTHER WORK

This study has highlighted numerous unobvious and intricate vulnerabilities of CA certificate issuance, underpinning the susceptibility of current certificate infrastructure to ransomware attack. Whilst CA tools such as Certbot offer a largely robust and reliable framework for certificate management, the findings of the study reveal vulnerabilities still exist, ranging from negligible to critical risk. Particularly concerning incomplete chains and revoked certificates, security breaches are at their highest potential and particularly as automation in attacks grows through bleeding-edge technologies. With support for ransomware deployment through Artificial Intelligence (AI), Machine Learning (ML), and the decentralised Web, Web3, it is paramount that future efforts towards safeguarding are brought forward.

6.1 Future Work

Towards such future work, or preferably, imminent work, let us consider the following innovative opportunities:

- **Improvements on Current Certificate Issuance.** Introducing inherent Online Certificate Status Protocol (OCSP), and Certificate Revocation Lists (CRL), which should correlate and update the certificates under revocation to be subsequently reflected in the browser as a flagged warning as other breaches currently are.
- **Improvements on Current Query Tools.** Automated scripting techniques through Bash (Singh and Vishwakarma, 2023), or similar in a plug-and-play execution file for ease of use, where severity ratings are automatically issued to the user from queries launched on URL browsing. Such query tools could be added as browser extensions for automated detection.
- **User Education Programmes.** User education in organisational training - although it is widely understood this is not a reliable method of eradicating ransomware per se, and automated prevention is preferable.
- **Utilisation of AI and ML.** Models to crawl the web towards detecting potential ransomware entry points by employing query tools and other automated scripts through user behaviour - but as these models demand enormous datasets, this could be a slow exercise and may also be criticised in attempt to ‘boil the ocean’.
- **Advancements Towards Web3.** As the web in general moves increasingly towards decentralised and distributed practices, smart contracts and immutable rulesets are coming into their own. This is potentially the most effective starting point, as such practices are currently an open area and collaboration with large and mature, trusted CA such as Let’s Encrypt need not be approached for collaboration with suggestions of improvement.

REFERENCES

Chukka, A. and Devi, V. (2021). Detection of malicious binaries by deep learning methods. In *Proceedings of the 6th International Conference on Internet of Things, Big Data and Security*. SCITEPRESS - Science and Technology Publications.

Housley, R. and Polk, T. (2001). *Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

Lu, Z. and Thing, V. L. L. (2022). PhilaeX: Explaining the failure and success of AI models in malware detection.

Obetta, S. O. and Moldovan, A.-N. (2023). Detection of DDoS attacks on urban IoT devices using neural networks. In *Proceedings of the 8th International Conference on Internet of Things, Big Data and Security*. SCITEPRESS - Science and Technology Publications.

(OWASP), O. W. A. S. P. OWASP risk rating methodology. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology. Accessed: 2021-10-16.

Rudd, S. and Cunningham, H. (2022). Threat modelling with the GDPR towards a security and privacy metrics framework for IoT smart-farm application.

Singh, K. and Vishwakarma (2023). Automation in manual penetration testing using bash shell script. *International Research Journal of Modernization in Engineering Technology and Science*.

Tamás, C., Papp, D., and Buttyán, L. (2021). SIMBIoTA: Similarity-based malware detection on IoT devices. In *Proceedings of the 6th International Conference on Internet of Things, Big Data and Security*. SCITEPRESS - Science and Technology Publications.

Tuleuov, B. I. and Ospanova, A. B. (2024). OpenSSL. In Tuleuov, B. I. and Ospanova, A. B., editors, *Beginning C++ Compilers: An Introductory Guide to Microsoft C/C++ and MinGW Compilers*, pages 157–163. Apress, Berkeley, CA.

Vlad, A., Luca, A., Hodea, O., and Tataru, R. Proceedings of the romanian academy. https://acad.ro/sectii2002/proceedings/doc2013-3s/ProceedingsSpecialssue2013.pdf#page=20. Accessed: 2024-1-25.

Zulfiqar, M., Janjua, M. U., Hassan, M., Ahmad, T., Saleem, T., and Stokes, J. W. (2022). Tracking adoption of revocation and cryptographic features in x.509 certificates. *Int. J. Inf. Secur.*, 21(3):653–668.