Threat Modeling Report

Created on 11/7/2019 2:46:08 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

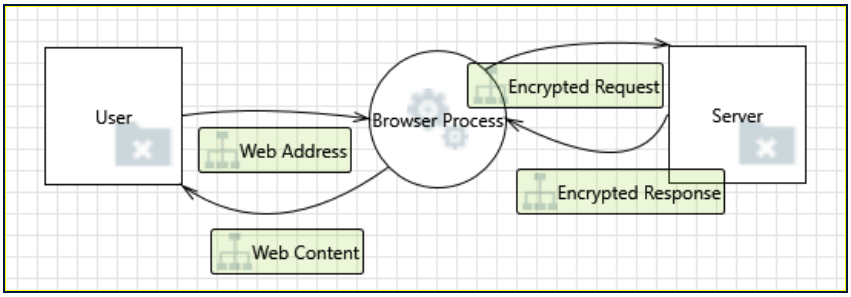
Notes:

Id	Note	Date	Added By
1	CRL List - https://github.com/agl/crlset-tools/blob/master/crlset.go shows google hosts the list	11/7/2019 7:45:18 AM	KIEWITPLAZA\Jacob.Barna
2	updater client source: https://cs.chromium.org/chromium/src/components/update_client/net/network_impl.cc?q=crlset+update&dr=CSs	11/7/2019 7:45:51 AM	KIEWITPLAZA\Jacob.Barna

Threat Model Summary:

Not Started	0
Not Applicable	15
Needs Investigation	5
Mitigation Implemented	18
Total	38
Total Migrated	0

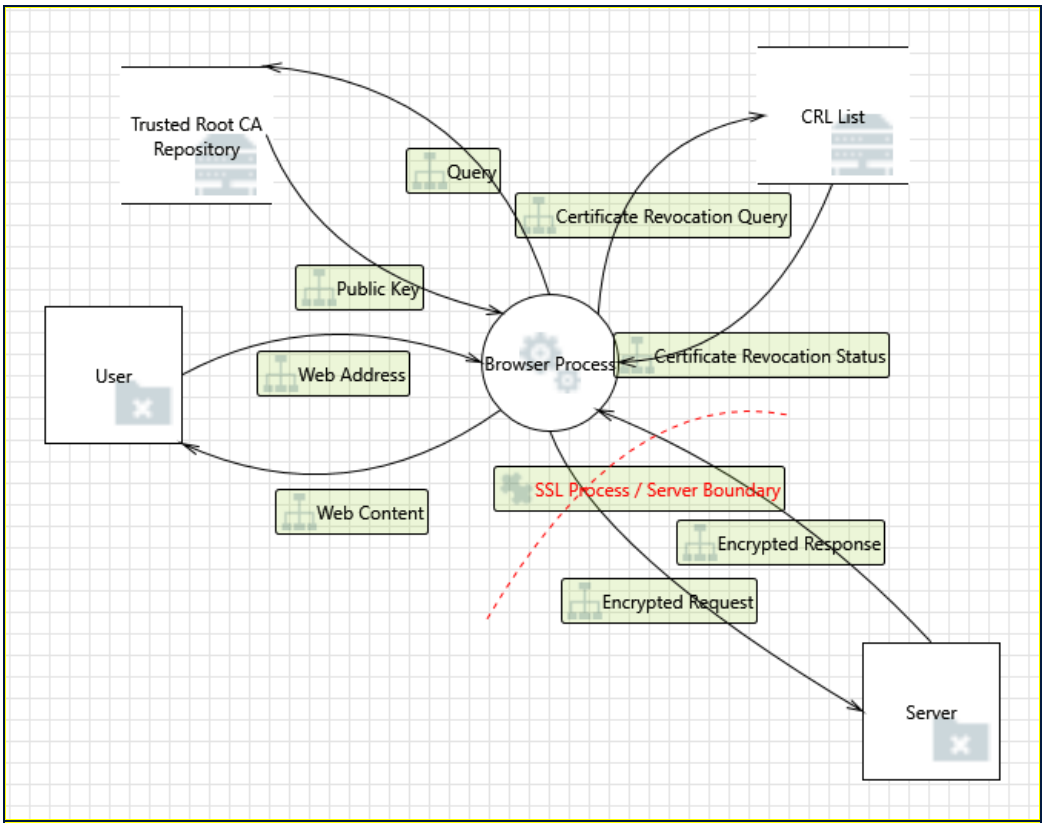
Diagram: Level 0 - SSL



Level 0 - SSL Diagram Summary:

Not Started	0
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	0
Total Migrated	0

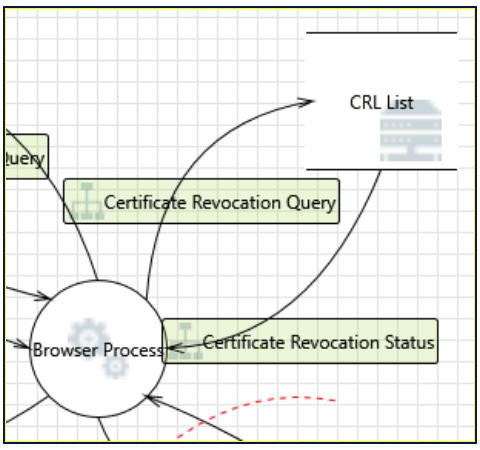
Diagram: Level 1 - SSL



Level 1 - SSL Diagram Summary:

Not Started	0
Not Applicable	12
Needs Investigation	3
Mitigation Implemented	7
Total	22
Total Migrated	0

Interaction: Certificate Revocation Query



1. Potential Excessive Resource Consumption for Browser Process or CRL List [State: Mitigation Implemented] [Priority: High]

Category:	Denial Of Service
Description:	Does Browser Process or CRL List take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.
Justification:	Google places a limit on the size of the file: https://www.grc.com/revocation/crlsets.htm

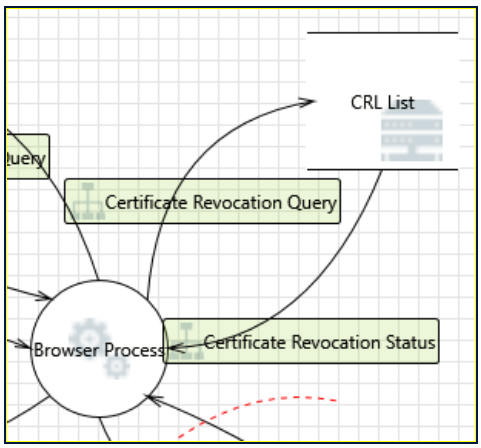
2. Spoofing of Destination Data Store CRL List [State: Not Applicable] [Priority: High]

Category: Spoofing

Description: CRL List may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of CRL List. Consider using a standard authentication mechanism to identify the destination data store.

Justification: Nothing in the request for the CRL list would be considered to be an information disclosure. The CRLSet is public and the request to get the list does not require encryption.

Interaction: Certificate Revocation Status



3. Weak Access Control for a Resource [State: Not Applicable] [Priority: High]

Category: Information Disclosure

Description: Improper data protection of CRL List can allow an attacker to read information not intended for disclosure. Review authorization settings.

Justification: The list is public, so reading the list is not a disclosure event.

4. Spoofing of Source Data Store CRL List [State: Needs Investigation] [Priority: High]

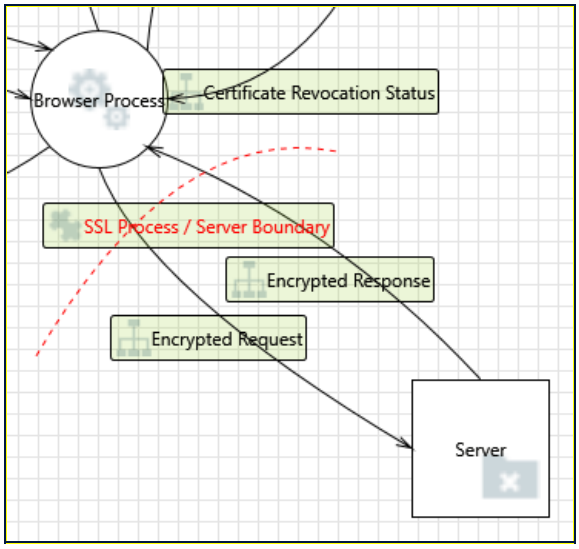
Category: Spoofing

Description: CRL List may be spoofed by an attacker and this may lead to incorrect data delivered to Browser Process. Consider using a standard authentication mechanism to identify the source data store.

Justification: We need to look further into this -- i'm not aware if the list is encrypted... looking at the file on disk, one is able to see the revoked certificates and edit them apparently without causing harm to Chrome. I'm curious if the code that loads this file verifies the authenticity of the file before using. If not, this is a bad design even though it would require someone with write permissions to this users' directory to do anything with it.

NEEDS MORE WORK TO PROVE -- can we come up with a test that shows the CRL list that is loaded in chrome and show that it accepts input?

Interaction: Encrypted Request



5. Data Flow Encrypted Request Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: Denial of service attacks of this type are generally not handled by the browser. These are lower level networking problems.

6. External Entity Server Potentially Denies Receiving Data [State: Not Applicable] [Priority: High]

Category: Repudiation
Description: Server claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification: The Server is out of the control of the browser.

7. Spoofing of the Server External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing
Description: Server may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Server. Consider using a standard authentication mechanism to identify the external entity.
Justification: For Man In The Middle attacks, Chrome has implemented certificate validation:
https://cs.chromium.org/chromium/src/net/cert/cert_verifier.cc?q=certificate+verifier&sq=package:chromium&dr=CSs

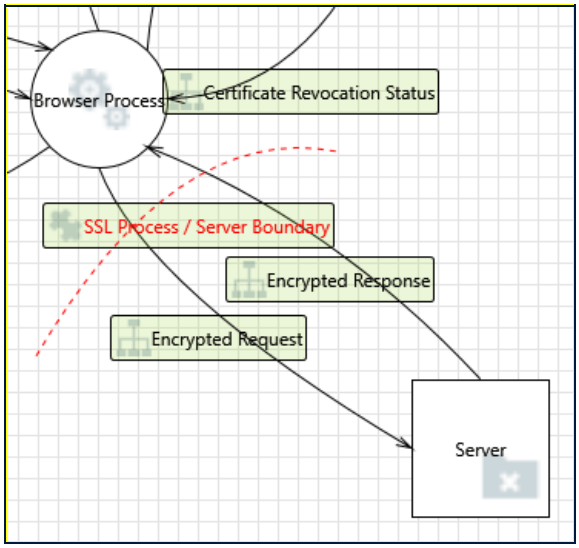
There are very few MITM attacks that would be successful, but they do exist and should be investigated... if WPAD is enabled by default on the Brave browser, this would not be considered a strong default as some sources claim that "the only sensible solution is to disable WPAD and setup proxies manually."
https://t3chnocat.com/tutorial-wpad-mitm/
https://auth0.com/blog/heads-up-https-is-not-enough-when-using-wpad/

If an attacker is just eavesdropping, decryption is improbable. After Heartbleed, Google has forked OpenSSL and renamed it BoringSSL. This means that an attacker may see traffic, but they will not be able to decrypt it. BoringSSL prefers ChaCha20 which is currently believed to be secure for symmetric encryption:
https://cs.chromium.org/search/?q=chacha&sq=package:chromium&type=cs
https://cs.chromium.org/chromium/src/third_party/boringssl/src/ssl/ssl_cipher.cc?q=diffie+boringssl&sq=package:chromium&dr=C

If not available, a built-in list of other secure symmetric encryption schemes are available for use.

For key exchange assymetric encryption (Diffie Hellman key exchange) is used:
https://cs.chromium.org/chromium/src/third_party/boringssl/src/ssl/ssl_cipher.cc?q=diffie+boringssl&sq=package:chromium&dr=C

Interaction: Encrypted Response



8. Browser Process May be Subject to Elevation of Privilege Using Remote Code Execution [State: Needs

Investigation] [Priority: High]

Category: Elevation Of Privilege
Description: Server may be able to remotely execute code for Browser Process.
Justification: This is a real, on-going threat.
https://www.cisecurity.org/advisory/multiple-vulnerabilities-in-google-chrome-could-allow-for-arbitrary-code-execution_2019-118/

Chrome security team constantly releases fixes, but there appears to be a history of such attacks. This makes it dangerous to run the browser in adminstrator mode.

9. Elevation Using Impersonation [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: Browser Process may be able to impersonate the context of Server in order to gain additional privilege.
Justification: The Server is out of the control of the browser.

10. Data Flow Encrypted Response Is Potentially Interrupted [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: Denial of service attacks of this type are generally not handled by the browser. These are lower level networking problems.

11. Potential Process Crash or Stop for Browser Process [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: Browser Process crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: This is not applicable to this analysis.

12. Data Flow Sniffing [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: Data flowing across Encrypted Response may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.
Justification: After Heartbleed, Google has forked OpenSSL and renamed it BoringSSL. This means that an attacker may see traffic, but they will not be able to decrypt it. BoringSSL prefers ChaCha20 which is currently believed to be secure for symmetric encryption:
https://cs.chromium.org/search/?q=chacha&sq=package:chromium&type=cs
https://cs.chromium.org/chromium/src/third_party/boringssl/src/ssl/ssl_cipher.cc?q=diffie+boringssl&sq=package:chromium&dr=C

If not available, a built-in list of other secure symmetric encryption schemes are available for use.

For key exchange assymetric encryption (Diffie Hellman key exchange) is used:
https://cs.chromium.org/chromium/src/third_party/boringssl/src/ssl/ssl_cipher.cc?q=diffie+boringssl&sq=package:chromium&dr=C

13. Potential Data Repudiation by Browser Process [State: Needs Investigation] [Priority: High]

Category: Repudiation
Description: Browser Process claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification: A strong default in this area would be to enable logging. Chrome does not by default enable logging, but perhaps Brave does?

https://www.chromium.org/for-testers/enable-logging

14. Potential Lack of Input Validation for Browser Process [State: Mitigation Implemented] [Priority: High]

Category: Tampering

Description: Data flowing across Encrypted Response may be tampered with by an attacker. This may lead to a denial of service attack against Browser Process or an elevation of privilege attack against Browser Process or an information disclosure by Browser Process. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.

Justification: After Heartbleed, Google has forked OpenSSL and renamed it BoringSSL. This means that an attacker may see traffic, but they will not be able to decrypt it. BoringSSL prefers ChaCha20 which is currently believed to be secure for symmetric encryption:
https://cs.chromium.org/search/?q=chacha&sq=package:chromium&type=cs
https://cs.chromium.org/chromium/src/third_party/boringssl/src/ssl/ssl_cipher.cc?q=diffie+boringssl&sq=package:chromium&dr=C

If not available, a built-in list of other secure symmetric encryption schemes are available for use.

For key exchange assymetric encryption (Diffie Hellman key exchange) is used:
https://cs.chromium.org/chromium/src/third_party/boringssl/src/ssl/ssl_cipher.cc?q=diffie+boringssl&sq=package:chromium&dr=C

15. Spoofing the Server External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Server may be spoofed by an attacker and this may lead to unauthorized access to Browser Process. Consider using a standard authentication mechanism to identify the external entity.

Justification: The server can be authenticated in SSL mode using certificate validation:
https://cs.chromium.org/chromium/src/net/cert/cert_verifier.cc?q=certificate+verifier&sq=package:chromium&dr=CSs

16. Spoofing the Browser Process Process [State: Mitigation Implemented] [Priority: High]

Category: Spoofing

Description: Browser Process may be spoofed by an attacker and this may lead to information disclosure by Server. Consider using a standard authentication mechanism to identify the destination process.

Justification: For Man In The Middle attacks, Chrome has implemented certificate validation:
https://cs.chromium.org/chromium/src/net/cert/cert_verifier.cc?q=certificate+verifier&sq=package:chromium&dr=CSs

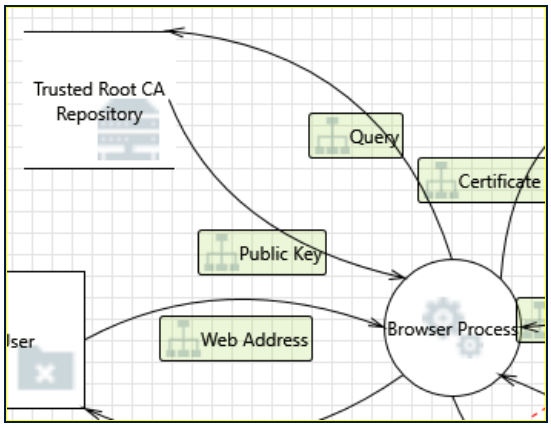
There are very few MITM attacks that would be successful, but they do exist and should be investigated... if WPAD is enabled by default on the Brave browser, this would not be considered a strong default as some sources claim that "the only sensible solution is to disable WPAD and setup proxies manually."
https://t3chnocat.com/tutorial-wpad-mitm/
https://auth0.com/blog/heads-up-https-is-not-enough-when-using-wpad/

If an attacker is just eavesdropping, decryption is improbable. After Heartbleed, Google has forked OpenSSL and renamed it BoringSSL. This means that an attacker may see traffic, but they will not be able to decrypt it. BoringSSL prefers ChaCha20 which is currently believed to be secure for symmetric encryption:
https://cs.chromium.org/search/?q=chacha&sq=package:chromium&type=cs
https://cs.chromium.org/chromium/src/third_party/boringssl/src/ssl/ssl_cipher.cc?q=diffie+boringssl&sq=package:chromium&dr=C

If not available, a built-in list of other secure symmetric encryption schemes are available for use.

For key exchange assymetric encryption (Diffie Hellman key exchange) is used:
https://cs.chromium.org/chromium/src/third_party/boringssl/src/ssl/ssl_cipher.cc?q=diffie+boringssl&sq=package:chromium&dr=C

Interaction: Public Key



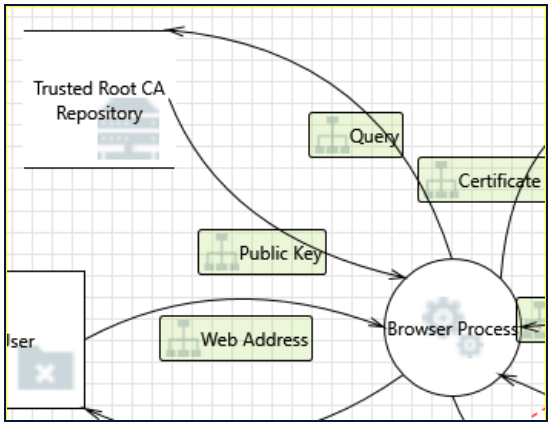
17. Weak Access Control for a Resource [State: Not Applicable] [Priority: High]

Category: Information Disclosure
Description: Improper data protection of Trusted Root CA Repository can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification: This is out of scope for the browser. Typically, the OS only allows trusted roots to be added for the user level if the user is not an administrator. Admins can install trusted roots. The browser cannot and should not control these configurations.

18. Spoofing of Source Data Store Trusted Root CA Repository [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: Trusted Root CA Repository may be spoofed by an attacker and this may lead to incorrect data delivered to Browser Process. Consider using a standard authentication mechanism to identify the source data store.
Justification: This is out of scope for the browser. Typically, the OS only allows trusted roots to be added for the user level if the user is not an administrator. Admins can install trusted roots. The browser cannot and should not control these configurations.

Interaction: Query



19. Potential Excessive Resource Consumption for Browser Process or Trusted Root CA Repository [State: Not Applicable] [Priority: High]

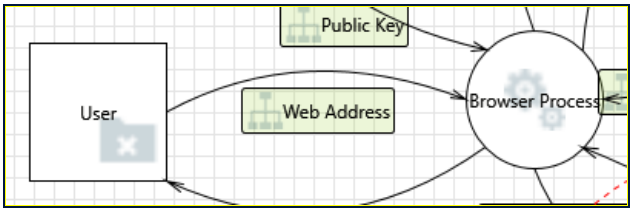
Category: Denial Of Service
Description: Does Browser Process or Trusted Root CA Repository take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.
Justification: Out of scope -- this is an OS-level feature.

20. Spoofing of Destination Data Store Trusted Root CA Repository [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: Trusted Root CA Repository may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Trusted Root CA Repository. Consider using a standard authentication mechanism to identify the destination data store.

Justification: This is out of scope for the browser. Typically, the OS only allows trusted roots to be added for the user level if the user is not an administrator. Admins can install trusted roots. The browser cannot and should not control these configurations.

Interaction: Web Address



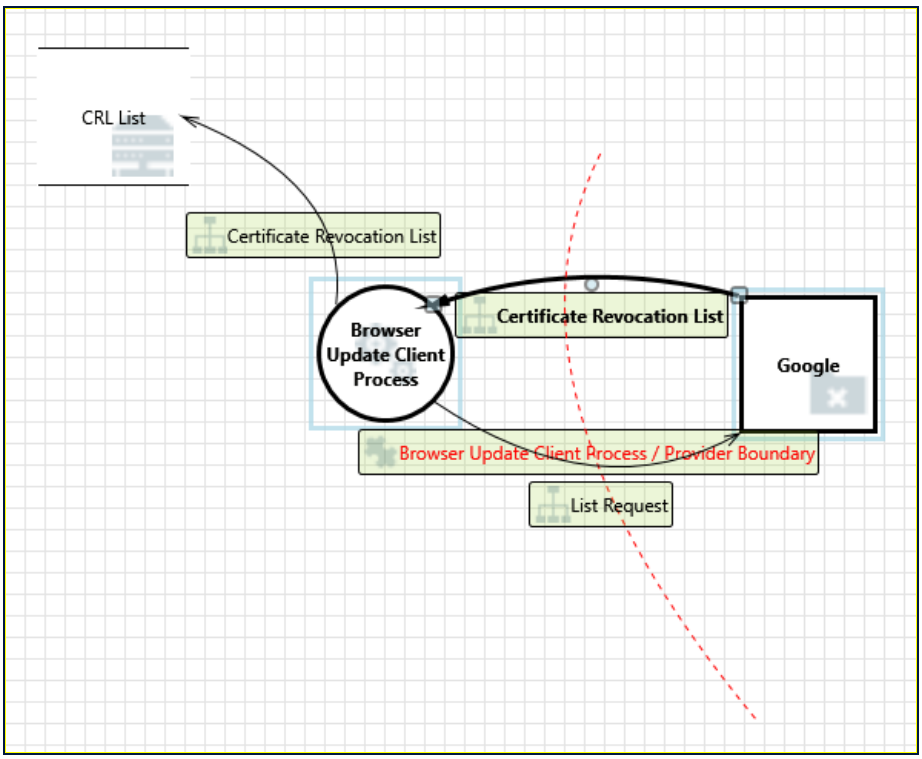
21. Spoofing the User External Entity [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: User may be spoofed by an attacker and this may lead to unauthorized access to Browser Process. Consider using a standard authentication mechanism to identify the external entity.
Justification: Any attacks requiring access to the machine are out of scope for this browser. If a user account has been compromised, there is nothing for the browser to do.

22. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege
Description: Browser Process may be able to impersonate the context of User in order to gain additional privilege.
Justification: The browser runs in least privilege mode -- meaning that unless the user has run the browser in administrator context, the user will be warned if any operations requiring an elevated privilege level are attempted.

Diagram: Level 1 - CRL Update

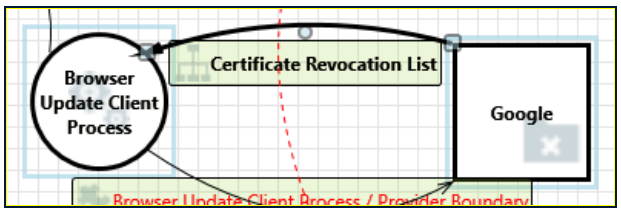


Level 1 - CRL Update Diagram Summary:

Not Started	0
Not Applicable	3
Needs Investigation	2

Mitigation Implemented	11
Total	16
Total Migrated	0

Interaction: Certificate Revocation List



23. Spoofing the CRL Provider External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing
Description: Google may be spoofed by an attacker and this may lead to unauthorized access to Browser Update Client Process. Consider using a standard authentication mechanism to identify the external entity.
Justification: The CRL Set is verified using the public key of the certificate signing the CRL set: https://cs.chromium.org/chromium/src/components/component_updater/component_installer.cc?type=cs&g=0&l=400 https://cs.chromium.org/chromium/src/chrome/browser/component_updater/crl_set_component_installer.cc?type=cs&g=0&l=31

24. Spoofing the CRL Update Process Process [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: Browser Update Client Process may be spoofed by an attacker and this may lead to information disclosure by Google. Consider using a standard authentication mechanism to identify the destination process.
Justification: The CRL Set shared between this is public, so there is no risk of information disclosure.

25. Potential Lack of Input Validation for CRL Update Process [State: Mitigation Implemented] [Priority: High]

Category: Tampering
Description: Data flowing across Certificate Revocation List may be tampered with by an attacker. This may lead to a denial of service attack against Browser Update Client Process or an elevation of privilege attack against Browser Update Client Process or an information disclosure by Browser Update Client Process. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.
Justification: The CRL Set is verified using the public key of the certificate signing the CRL set: https://cs.chromium.org/chromium/src/components/component_updater/component_installer.cc?type=cs&g=0&l=400 https://cs.chromium.org/chromium/src/chrome/browser/component_updater/crl_set_component_installer.cc?type=cs&g=0&l=31

26. Potential Data Repudiation by CRL Update Process [State: Mitigation Implemented] [Priority: High]

Category: Repudiation
Description: Browser Update Client Process claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification: There is no threat arising from the browser process claiming that it did not receive data from the CRL List Server. However, if one wished to examine the content sent from the CRL it is available in the directory: C:\Users\<USER>\AppData\Local\Google\Chrome\User Data\CertificateRevocation

27. Data Flow Sniffing [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: Data flowing across Certificate Revocation List may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.

Justification: There is no risk of information disclosure as the set is public. Furthermore, the CRL Set is verified using the public key of the certificate signing the CRL set:
https://cs.chromium.org/chromium/src/components/component_updater/component_installer.cc?type=cs&g=0&l=400
https://cs.chromium.org/chromium/src/chrome/browser/component_updater/crl_set_component_installer.cc?type=cs&g=0&l=31

28. Potential Process Crash or Stop for CRL Update Process [State: Not Applicable] [Priority: High]

Category: Denial Of Service

Description: Browser Update Client Process crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: This is not applicable to the CRL update process.

29. Data Flow Certificate Revocation List Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: The CRLSet is stored locally:
C:\Users\<USER>\AppData\Local\Google\Chrome\User Data\CertificateRevocation

An attacker could theoretically prevent the downloading of updated CRLSets, however, any old revoked certificates would still be present in the CRLSet. An attacker would need to compromise a recent certificate and conduct the DDOS attack.

Interestingly, the Chromium team has decided to stop using online certificate revocation checks (due to many reasons including performance and soft failures), so there has been some criticism of this approach that should be investigated further. According to Gibson Research Corporation, there could be as few as 3% of revoked certificates in this curated list from Google (https://www.grc.com/revocation/crlsets.htm) with many root CAs not even on the list.

If an attacker were to gain control of a revoked certificate not on the list they could potentially conduct a successfull Man-in-the-Middle attack without security warnings from Chrome.

30. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Browser Update Client Process may be able to impersonate the context of Google in order to gain additional privilege.

Justification: Browser update client process validates the CRL using the public key provided by Google. Furthermore, the browser runs in least privilege mode -- meaning that unless the user has run the browser in administrator context, the user will be warned if any operations requiring an elevated privilege level are attempted.

31. CRL Update Process May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

Description: Google may be able to remotely execute code for Browser Update Client Process.

Justification: The component installer is designed to install only properly packaged objects. Unless an attacker could successfully move malicious code through the Pull Request process for Chromium, this risk is mitigated by the component installer.
https://cs.chromium.org/chromium/src/components/component_updater/component_installer.cc?type=cs&g=0&l=388

32. Elevation by Changing the Execution Flow in CRL Update Process [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege

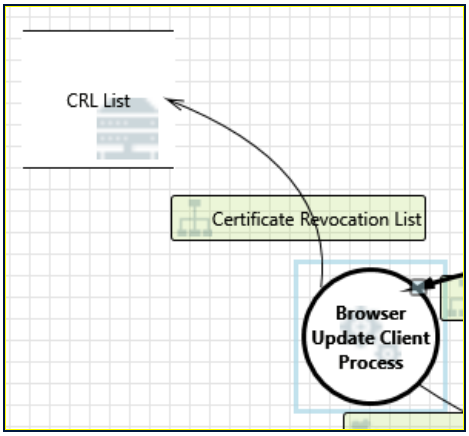
Description: An attacker may pass data into Browser Update Client Process in order to change the flow of program execution within Browser Update Client Process to the attacker's choosing.

Justification: The CRL Set is verified using the public key of the certificate signing the CRL set:
https://cs.chromium.org/chromium/src/components/component_updater/component_installer.cc?type=cs&g=0&l=400
https://cs.chromium.org/chromium/src/chrome/browser/component_updater/crl_set_component_installer.cc?type=cs&g=0&l=31

33. Cross Site Request Forgery [State: Mitigation Implemented] [Priority: High]

Category:	Elevation Of Privilege
Description:	Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting, … The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations.
Justification:	The network traffic here is not between browser and web page, it is downloading a file. This file, the CRL Set, is verified using the public key of the certificate signing the CRL set: https://cs.chromium.org/chromium/src/components/component_updater/component_installer.cc?type=cs&g=0&l=400 https://cs.chromium.org/chromium/src/chrome/browser/component_updater/crl_set_component_installer.cc?type=cs&g=0&l=31

Interaction: Certificate Revocation List



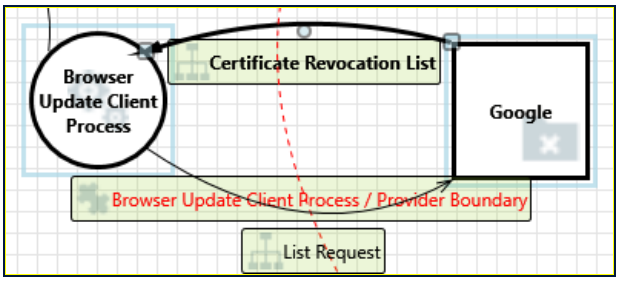
34. Spoofing of Destination Data Store CRL List [State: Not Applicable] [Priority: High]

Category:	Spoofing
Description:	CRL List may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of CRL List. Consider using a standard authentication mechanism to identify the destination data store.
Justification:	Nothing in the request for the CRL list would be considered to be an information disclosure. The CRLSet is public and if an attacker gains the list, they have nothing that is not considered public.

35. Potential Excessive Resource Consumption for CRL Update Process or CRL List [State: Mitigation Implemented] [Priority: High]

Category:	Denial Of Service
Description:	Does Browser Update Client Process or CRL List take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.
Justification:	Google places a limit on the size of the file: https://www.grc.com/revocation/crlsets.htm

Interaction: List Request



36. Spoofing of the CRL Provider External Destination Entity [State: Mitigation Implemented] [Priority: High]

Category:	Spoofing
Description:	Google may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Google. Consider using a standard authentication mechanism to identify the external entity.
Justification:	There is no data that is worth encrypting that is sent to Google, so an attacker could only gain a request for the CRL Set. There is no proprietary information sent in the request. https://cs.chromium.org/chromium/src/chrome/browser/component_updater/crl_set_component_installer.cc?q=crlset+update&dr=CSs

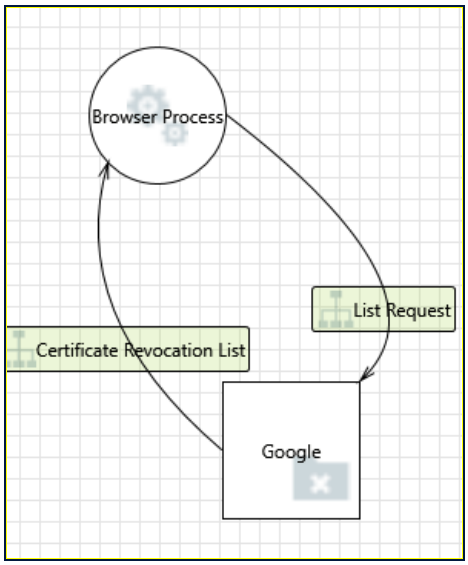
37. External Entity CRL Provider Potentially Denies Receiving Data [State: Mitigation Implemented] [Priority: High]

Category:	Repudiation
Description:	Google claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification:	There is no threat arising from the browser process claiming that it did not receive data from the CRL List Server. However, if one wished to examine the content sent from the CRL it is available in the directory: C:\Users\<USER>\AppData\Local\Google\Chrome\User Data\CertificateRevocation

38. Data Flow List Request Is Potentially Interrupted [State: Needs Investigation] [Priority: High]

Category:	Denial Of Service
Description:	An external agent interrupts data flowing across a trust boundary in either direction.
Justification:	The CRLSet is stored locally: C:\Users\<USER>\AppData\Local\Google\Chrome\User Data\CertificateRevocation An attacker could theoretically prevent the downloading of updated CRLSets, however, any old revoked certificates would still be present in the CRLSet. An attacker would need to compromise a recent certificate and conduct the DDOS attack. Interestingly, the Chromium team has decided to stop using online certificate revocation checks (due to many reasons including performance and soft failures), so there has been some criticism of this approach that should be investigated further. According to Gibson Research Corporation, there could be as few as 3% of revoked certificates in this curated list from Google (https://www.grc.com/revocation/crlsets.htm) with many root CAs not even on the list. If an attacker were to gain control of a revoked certificate not on the list they could potentially conduct a successfull Man-in-the-Middle attack without security warnings from Chrome.

Diagram: Level 0 - CRL Update



Level 0 - CRL Update Diagram Summary:

Not Started	0
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	0
Total Migrated	0