Threat Modeling Report

Created on 11/7/2018 6:30:41 PM

Threat Model Name: Installation and Registration

Owner: Kendrick Urbaniak

Reviewer: Kendrick Urbaniak

Contributors:

Description: This system allows for the installing and registering of an account to use Certbot.

Assumptions: All internal processes and file systems are properly run and they only require correct functionality once.

External Dependencies: There are no external dependencies on this code.

Threat Model Summary:

Not Started	0
Not Applicable	30
Needs Investigation	2
Mitigation Implemented	5
Total	37
Total Migrated	0

Diagram: Diagram 1

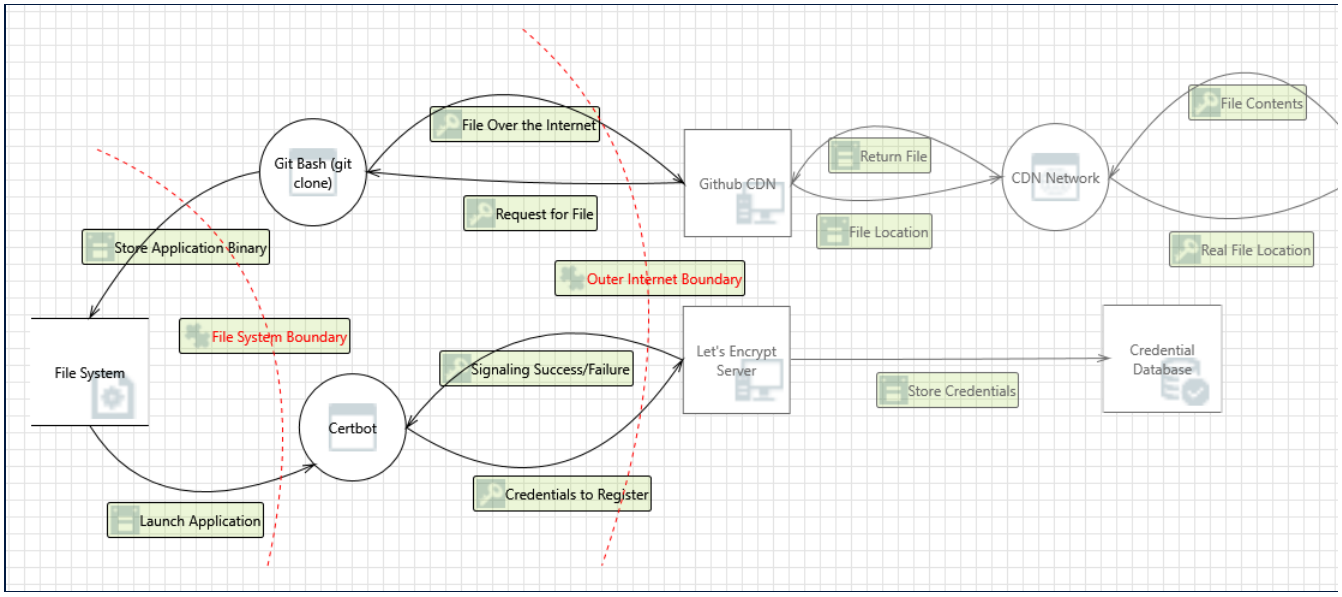
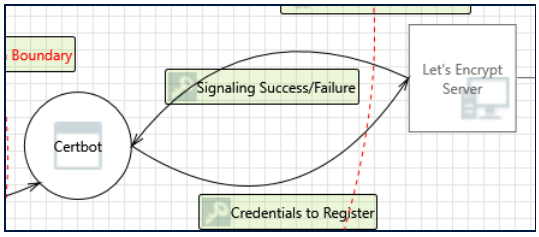


Diagram 1 Diagram Summary:

Not Started	0
Not Applicable	30
Needs Investigation	2
Mitigation Implemented	5
Total	37
Total Migrated	0

Interaction: Credentials to Register



1. Spoofing of the Let's Encrypt Server External Destination Entity [State: Needs Investigation] [Priority: High]

Category: Spoofing
Description: Let's Encrypt Server may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Let's Encrypt Server. Consider using a standard authentication mechanism to identify the external entity.
Justification: <no mitigation provided>

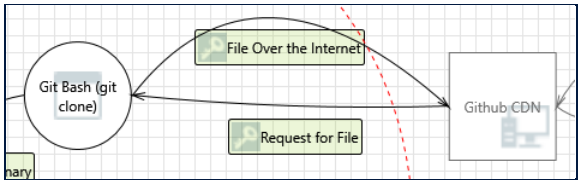
2. External Entity Let's Encrypt Server Potentially Denies Receiving Data [State: Not Applicable] [Priority: High]

Category: Repudiation
Description: Let's Encrypt 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: If the Let's Encrypt Server doesn't get the data then

3. Data Flow HTTPS 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: A Denial of Service results in no certificate generation which does not hinder operation under HTTP information flow.

Interaction: File Over the Internet



4. External Entity Github CDN Potentially Denies Receiving Data [State: Not Applicable] [Priority: High]

Category: Repudiation
Description: Github CDN 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: If Github is down then there are bigger problems.

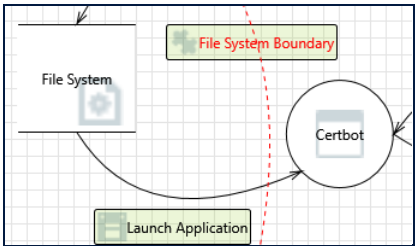
5. Spoofing of the Github CDN External Destination Entity [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: Github CDN may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Github CDN. Consider using a standard authentication mechanism to identify the external entity.
Justification: Github uses certificates which authenticates the address

6. Data Flow HTTPS 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: Not a critical vulnerability and not preventable.

Interaction: Launch Application



7. Weak Access Control for a Resource [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: Improper data protection of File System can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification: The files are owned by the web account.

8. Spoofing of Source Data Store File System [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: File System may be spoofed by an attacker and this may lead to incorrect data delivered to Certbot. Consider using a standard authentication mechanism to identify the source data store.
Justification: Highly unlikely and requires physical access to the machine.

9. Spoofing the Certbot Process [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: Certbot may be spoofed by an attacker and this may lead to information disclosure by File System. Consider using a standard authentication mechanism to identify the destination process.
Justification: No. Just no.

10. Potential Data Repudiation by Certbot [State: Not Applicable] [Priority: High]

Category: Repudiation
Description: Certbot 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: <no mitigation provided>

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

Category: Denial Of Service
Description: Certbot crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: <no mitigation provided>

12. Data Flow Binary 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: The program would crash and need to be restarted which is desirable in this case.

13. Data Store Inaccessible [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: This would require root access.

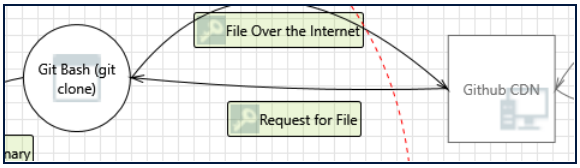
14. Certbot May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: File System may be able to remotely execute code for Certbot.
Justification: <no mitigation provided>

15. Elevation by Changing the Execution Flow in Certbot [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: An attacker may pass data into Certbot in order to change the flow of program execution within Certbot to the attacker's choosing.
Justification: The filesystem can do what ever it wants and editing the file system requires web account access in other words they could just steal the certificate

Interaction: Request for File



16. Elevation by Changing the Execution Flow in OS Process [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: An attacker may pass data into Git Bash (git clone) in order to change the flow of program execution within Git Bash (git clone) to the attacker's choosing.
Justification: Git Bash is controlled and maintained by the Github community and outside the scope of company.

17. OS Process May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Applicable] [Priority: High]

Category: Elevation Of Privilege
Description: Github CDN may be able to remotely execute code for Git Bash (git clone).
Justification: Yes it could but probably not maliciously.

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

Category: Elevation Of Privilege
Description: Git Bash (git clone) may be able to impersonate the context of Github CDN in order to gain additional privilege.
Justification: <no mitigation provided>

19. Data Flow HTTPS 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: A mild inconvinience.

20. Potential Process Crash or Stop for OS Process [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service
Description: Git Bash (git clone) crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: This is an easily restartable process that only needs to work in some capacity once.

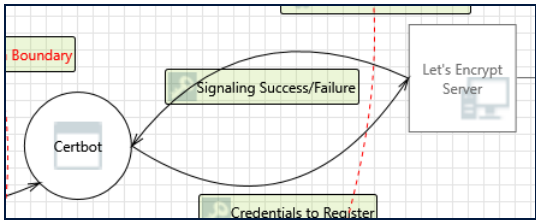
21. Potential Data Repudiation by OS Process [State: Not Applicable] [Priority: High]

Category: Repudiation
Description: Git Bash (git clone) 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: This does not matter in this context.

22. Spoofing the Github CDN External Entity [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: Github CDN may be spoofed by an attacker and this may lead to unauthorized access to Git Bash (git clone). Consider using a standard authentication mechanism to identify the external entity.
Justification: Traffic to this box is on a request only basis and thus Github CDN does not make a request to

Interaction: Signaling Success/Failure



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

Category: Elevation Of Privilege
Description: Certbot may be able to impersonate the context of Let's Encrypt Server in order to gain additional privilege.
Justification: This doesn't make sense in the context of what Certbot does and the Let's Encrypt Server is not client owned.

24. Potential Data Repudiation by Certbot [State: Not Applicable] [Priority: High]

Category: Repudiation
Description: Certbot 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: Not an issue in this regard cause either the certificate is generated or it is not.

25. Potential Process Crash or Stop for Certbot [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: Certbot crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification: This is not a constant service so it does not need constant up time metrics.

26. Data Flow HTTPS 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 would result in a minor problem.

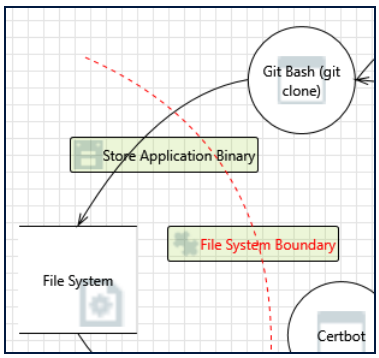
27. Certbot May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege
Description: Let's Encrypt Server may be able to remotely execute code for Certbot.
Justification: Let's Encrypt Server is a trusted domain and is a service website which should not initiate requests

28. Elevation by Changing the Execution Flow in Certbot [State: Needs Investigation] [Priority: High]

Category: Elevation Of Privilege
Description: An attacker may pass data into Certbot in order to change the flow of program execution within Certbot to the attacker's choosing.
Justification: <no mitigation provided>

Interaction: Store Application Binary



29. Data Store Inaccessible [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: An external agent prevents access to a data store on the other side of the trust boundary.
Justification: This requires root access.

30. Authorization Bypass [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: Can you access File System and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.
Justification: Files are owned by an administrator account specifically used by the web server for serving content.

31. Spoofing of Destination Data Store File System [State: Not Applicable] [Priority: High]

Category: Spoofing
Description: File System may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of File System. Consider using a standard authentication mechanism to identify the destination data store.
Justification: <no mitigation provided>

32. Potential Excessive Resource Consumption for Git Bash (git clone) or File System [State: Not Applicable] [Priority: High]

Category: Denial Of Service
Description: Does Git Bash (git clone) or File System 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: No but in the case that it does not timeout it should deadlock as this is a critical part of the application that is required to work.

33. Data Flow Binary Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification: The binary is corrupted and needs to redowloaded.

34. Weak Credential Transit [State: Not Applicable] [Priority: High]

Category: Information Disclosure
Description: Credentials on the wire are often subject to sniffing by an attacker. Are the credentials re-usable/re-playable? Are credentials included in a message? For example, sending a zip file with the password in the email. Use strong cryptography for the transmission of credentials. Use the OS libraries if at all possible, and consider cryptographic algorithm agility, rather than hardcoding a choice.
Justification: This is writing binary data to the file system; no credentials needed.

35. Data Flow Sniffing [State: Not Applicable] [Priority: High]

Category: Information Disclosure
Description: Data flowing across Store Application Binary 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: This would require root access.

36. Data Store Denies File System Potentially Writing Data [State: Not Applicable] [Priority: High]

Category: Repudiation
Description: File System claims that it did not write data received from an entity 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: Well that sucks.

37. The File System Data Store Could Be Corrupted [State: Not Applicable] [Priority: High]

Category: Tampering
Description: Data flowing across Store Application Binary may be tampered with by an attacker. This may lead to corruption of File System. Ensure the integrity of the data flow to the data store.
Justification: <no mitigation provided>