5. Denial of Service

Question

What is denial of service? How does availability mitigate this threat?

Answer

Denial of service is the prevention of authorized access to a system resource or the delaying of system operations and functions. A denial of service attack might be accomplished by bombarding a server with requests to consume all available system resources or by passing it malformed input data that can crash an application process.

Availability is the property of being accessible and usable upon demand by an authorized entity. Availability should be provided at both the network and application level as appropriate. Some potential mitigations include:

- Network-level mitigations:
  - Replication of resources to handle larger numbers of connections
  - Filtering, rate-limiting and a variety of other network techniques that have been developed to minimize effects of large-scale denial of service attacks
- Application-level mitigations:
  - Validation of input data to prevent overwhelming use of resources and prevent malformed data values from crashing the application
  - Handling all possible exceptions in a manner which prevents application down time

Mitigations for Specific Threats

Summary: Denial of service types of threats occur when an attacker can prevent legitimate users from accessing an application or service. In this case, an attacker can cause an application or process to crash, halt, stop, or run slowly.

Recommendations:

- Validate all input data to ensure the values cannot cause the process to crash
- Handle all exceptional cases (both programming language exceptions and exceptions from logical conditions) in a graceful manner which does not result in crashes or degraded service

Audit comments should indicate whether or not appropriate input validation is in place to prevent crashes and how exceptional cases are handled.

Summary: Denial of service types of threats occur when an attacker can prevent legitimate users from accessing an application or service, in this case an external agent interrupts data flowing across a trust boundary in either direction.

Recommendations:

- It should not block waiting for responses that cross the trust boundary
- If it is able to continue processing in a reduced capacity, it should do so
- If it is not able to continue processing, it should provide appropriate responses to users or other processes that are waiting or expecting responses from this process

Audit comments should indicate what, if any, network protections are in place for preventing interruption of the indicated data flow. The comments should also indicate how the application handles an interruption in the data flow. Please be sure that comments apply to the indicated data flow, as other data flows in the threat model may have different mitigations and should be addressed separately.
Summary: Denial of service types of threats occur when an attacker can prevent legitimate users from accessing an application or service. In this case, an external agent prevents access to a data store on the other side of the trust boundary.

Recommendations: Mitigations for this concern are primarily the responsibility of the network, however, the application or process should take measures to handle interruption as well:

- It should not block waiting for responses from the data store
- If it is able to continue processing in a reduced capacity, it should do so
- If it is not able to continue processing, it should provide appropriate responses to users or other processes that are waiting or expecting responses from this process

Audit comments should indicate what, if any, network protections are in place to ensure uninterrupted access to the data store. The comments should also indicate how the application handles an interruption in access to the data store. Please be sure that comments apply to the indicated data store, as other data stores in the threat model may have different mitigations and should be addressed separately.

Summary: Denial of service types of threats occur when an attacker can prevent legitimate users from accessing an application or service. In this case, an attacker can force an application or process to use excessive resources.

Recommendations: Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS handle releasing the resources. Appropriate mitigations depend on the types of resources consumed by the application. Mitigations may include:

- Don't deadlock on resource requests
- Set timeouts for resource requests, where appropriate
- Validate input data that relate to resource consumption
- Limit the size of files processed by the application
- Dispose of resources when they are no longer needed or used by the application

Audit comments should indicate what mitigations are in place to prevent excess resource consumption by the application or process, as well as what mitigations the application expects the operating system or network to provide.

References

1. MSDN: Improving Web Application Security: Chapter 2 Threats and Countermeasures
2. OWASP: Application Threat Modeling
4. Internet Security Glossary, Version 2, IETF RFC 4949