STAC Engagement 2 Vulnerabilities[1]

1 CyberPoint

1.1 GabFeed

GabFeed is web forum software. It allows community members to post messages, search posted messages, and provides additional information on selected ('special') topics.

1.1.1 GabFeed 1

1.1.1.1 Vulnerabilities

<u>SC Time</u>

The SC Time vulnerability allows a third party to discover the private key used by the server to authenticate itself to a user. Users are provided with a script which accepts a numerical input (u) and calculates an input to be sent to the server ($U = g^u \bmod m$) and the expected server response ($S = g^{us} \bmod m$). Users authenticate the server by navigating to https://localhost:8080/authenticate and submitting the value “U”. The server calculates the value of “$U^s \bmod m$” and returns it to the user. The user confirms that the server’s output and the expected server response are the same. The vulnerability lies in the server’s calculation of “$U^s \bmod m$”. The server iterates through the bits of the server secret key and performs an expensive operation when the bit is 1 vs when the bit is 0. This introduces an SC Time which a malicious user can use the leak the server’s secret key.

<u>AC Time</u>

The AC Time vulnerability is in the shortest path line breaking algorithm. The algorithm has O (n*width) behavior, where n is the number of words and width is the width of the client browser in pixels – used to inform the server of when to line-break the text. Normally, there is a JavaScript function that finds the width (in pixels) of the column where the text will be placed. The function issues a ‘PUT’ command to https://localhost:8080/width<width>, which sets the width. Using this method the width should never be large enough to cause the resource usage limit to be exceeded. An attacker can call the ‘PUT’ with a large width setting and post a corresponding long message. This will cause the resource usage limit to be exceeded.

1.1.1.2 Positive Questions

<u>Question 007 AC Time</u>
Challenge Program:

[1] The Apogee Research effort on STAC is funded by the United States Airforce Research Lab (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under Contract Number FA8750-15-C-0089. THE VIEWS AND CONCLUSIONS CONTAINED IN THIS DOCUMENT ARE THOSE OF THE AUTHORS AND SHOULD NOT BE INTERPRETED AS REPRESENTING THE OFFICIAL POLICIES, EITHER EXPRESSED OR IMPLIED, OF THE DEFENSE ADVANCED RESEARCH PROJECTS AGENCY OR THE US GOVERNMENT.

gabfeed_1.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

GabFeed is a web based application that supports multiple users, however, this question assumes there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 400000 bytes

Resource Usage Limit:

Runtime: 300 seconds measured as the time interval between sending the last byte of the attack and the server sending the last byte of the response.

Question 010 SC Time

Challenge Program:

gabfeed_1.jar

Challenge Question:

Is there a side channel in time in the challenge program from which a third party can discover the server's private key used in the server's authentication? Note: This is referring to the private key for authentication at https://localhost:8080 not the private key for establishing an SSL connection.

Additional background for question:

For the sake of this problem, a user session is defined as the user interactions between (and including) the establishment of a session key to the renegotiation of the next session key. The session key is related to an SSL session and is not related to the server's private key.

Available Operations:

Passive Operations: observe the encrypted packets within a single user session
or authentication of the server (cost 1 per session)
Active Operations: Issue an HTTP request and receive a response. (cost 1 per operation)
Oracle Query: query the oracle with a candidate private key to get a binary
response (yes/no) as to whether the candidate key is server's
private key (cost 1 per query)

Operational Budget:

Max number of operations : 60000
Probability of success : 95%

1.1.1.3 Null Questions

Question 029 Null SC Time

Challenge Program:

gabfeed_1.jar

Challenge Question:

Is there a side channel in time in the challenge program from which a third party can
learn information that allows them to impersonate any given user?

Additional background for question:

For the sake of this problem, a user session is defined as the user
interactions between (and including) the establishment of a session key to
the renegotiation of the next session key. The session key is related to an
SSL session and is not related to the server's private key.

Available Operations:

Passive Operations: observe the encrypted packets within a single user session
or authentication of the server (cost 1 per session)
Active Operations: Issue an HTTP request and receive a response. (cost 1 per operation)
Oracle Query: query the oracle to get a binary response (yes/no) as to whether
or not you have collected a valid secret from any given user such
that you can impersonate that user
(cost 1 per query)

Operational Budget:

Max number of operations : 500,000

Probability of success : 95%

Question 038 Null AC Space

Challenge Program:

gabfeed_1.jar

Challenge Question:

Is there an algorithmic complexity in space that would cause the challenge program's memory usage to exceed the resource usage limit in one user session given the input budget?

Additional background for question:

GabFeed is a web based application that supports multiple users, however, this question assumes there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 400000 bytes

Resource Usage Limit:

Available Memory: 512 MB (total memory usage of the JVM shown under the RSS column in pmap -x <PID>)

1.1.2 GabFeed 2

1.1.2.1 Vulnerabilities

AC Time

GabFeed 2 uses a tree-based hash table which is vulnerable to denial of service. The tree used is unbalanced therefore collisions can result in a slow performance. If an attacker posts a large message with n words containing hash collisions, the performance will become O (n^2) and the resource usage limit will be exceeded.

1.1.2.2 Positive Questions

Question 006 AC Time

Challenge Program:

gabfeed_2.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

GabFeed is a web based application that supports multiple users, however, this question assumes there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 400000 bytes

Resource Usage Limit:

Runtime: 150 seconds measured as the time interval between sending the last byte of the attack and the server sending the last byte of the response.

1.1.2.3 Null Questions

Question 032 Null SC Time

Challenge Program:

gabfeed_2.jar

Challenge Question:

Is there a side channel in time in the challenge program from which a third party can discover the server's private key used in the server's authentication? Note: This is referring to the private key for authentication at https://localhost:8080 not the private key for establishing an SSL connection.

Additional background for question:

For the sake of this problem, a user session is defined as the user interactions between (and including) the establishment of a session key to the renegotiation of the next session key. The session key is related to an SSL session and is not related to the server's private key.

Available Operations:

Passive Operations: observe the encrypted packets within a single user session
or authentication of the server (cost 1 per session)
Active Operations: Issue an HTTP request and receive a response. (cost 1 per operation)
Oracle Query: query the oracle with a candidate private key to get a binary
response (yes/no) as to whether the candidate key is the server's
private key (cost 1 per query)

Operational Budget:

Max number of operations : 60000
Probability of success : 95%

1.1.3 GabFeed 3

1.1.3.1 Vulnerabilities

SC Space/Time

In GabFeed 3, searches for special search terms contain a SC in both Space and Time. Searches for special search terms take longer than any other GabFeed operation. A malicious user can observe the server's response time during a user's session and determine when the user's request is a search for a special search term. Special search terms packets are uniquely sized, therefore a malicious user can map each special search term to a unique packet size. By combining both the SC Time and SC Space a malicious user can determine when a user has searched for a special term, and which special term the user searched for.

AC Time

The sorting algorithm used in GabFeed 3 is vulnerable to AC Time attacks. The algorithm typically has O (n log n) behavior but can have O (n^2) behavior depending on the recurrence relation provided to the instance of sort. The vulnerability does not depend on the contents of the collection being sorted, only on the size of the collection.

1.1.3.2 Positive Questions

Question 023 SC Time

Challenge Program:

gabfeed_3.jar

Challenge Question:

Is there a side channel in time in the challenge program with which a third party
can identify how many 'special' search terms a user has searched for within the

operational budget?

Additional background for question:

For the purpose of the question, assume that a user session will include at most 30 HTTP request/recieve response pairs.

Available Operations:

Passive Operations: observe the encrypted request and corresponding response packets and timings thereof for a single user session (cost: 1 per observed user session)
Active Operations: issue an HTTP request and receive a response (no cost per request)
Oracle queries: query the oracle with the candidate number of 'special' searches and receive a binary response (yes/no) as to whether the candidate number of 'special' search terms is correct. (cost: 1 per query)

Operational Budget:

Max number of operations : 2 operations
Probability of success : 90%

<u>Question 036 AC Time</u>

Challenge Program:

gabfeed_3.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

GabFeed is a web based application that supports multiple users, however, this question assumes
there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 400000 bytes

Resource Usage Limit:

Runtime: 300 seconds measured as the time interval between sending the last byte of the attack and
the server sending the last byte of the response.

Question 039 SC Space/Time

Challenge Program:

gabfeed_3.jar

Challenge Question:

Is there a side channel in time and space in the challenge program with which a
third party can identify if a user has searched for a single 'special' search term and
which specific 'special' term was searched for first within the operational budget?

Additional background for question:

For the purpose of the engagement assume that if a user will execute a special search
it will be one of their first three actions after logging in. Each search of a special term
is independent of all prior searches.

Available Operations:

Passive Operations: observe the encrypted request and corresponding response packets
and timings thereof for a single user http action
(cost: 1 per observed action)
Active Operations: issue an HTTP request and receive a response (no cost per request)
Oracle queries: query the oracle with a candidate search term to get a binary response
(yes/no) as to whether the candidate search term is the user's 'special'
search term. (cost: 1 per query)

Operational Budget:

Max number of operations : 5 operations
Probability of success : 90%

1.1.4 GabFeed 4

1.1.4.1 Vulnerabilities

AC Time

GabFeed 4 uses a hash table based on a red black tree. On ‘put’ calls, the tree is not balanced. An attacker can exceed the resource usage limit by posting a large message containing words with hash collisions.

1.1.4.2 Positive Questions

Question 003 AC Time

Challenge Program:

gabfeed_4.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

GabFeed is a web based application that supports multiple users, however, this question assumes
there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 400000 bytes

Resource Usage Limit:

Runtime: 150 seconds measured as the time interval between sending the last byte of the attack and
the server sending the last byte of the response.

1.1.5 GabFeed 5

1.1.5.1 Vulnerabilities

GabFeed 5 contains no intended vulnerabilities that are asked about in Engagement 2.

1.1.5.2 Null Questions

Question 001 Null SC Space

Challenge Program:

gabfeed_5.jar

Challenge Question:

Is there a side channel in space in the challenge program from which a third party can discover the GabFeed username of a user involved in a private chat?

Additional background for question:

None.

Available Operations:

Passive Operations: observe the encrypted packets and timings thereof within a single user session (cost: 1 per session)
Active Operations: issue an HTTP request and receive a response (no cost per query)
Oracle queries: query the oracle with a candidate user to get a binary response (yes/no) as to whether the candidate user is valid. (cost: 1 per query)

Operational Budget:

Max number of operations : 2 Operations
Probability of success : 90%

Question 004 Null SC Space

Challenge Program:

gabfeed_5.jar

Challenge Question:

Is there a side channel in space in the challenge program from which a third party can discover the server's private key used in the server's authentication? Note: This is referring to the private key for authentication at https://localhost:8080 not the private key for establishing an SSL connection.

Additional background for question:

For the sake of this problem, a user session is defined as the user interactions between (and including) the establishment of a session key to the renegotiation of the next session key. The session key is related to an SSL session and not related to the server's private key.

Available Operations:

Passive Operations: observe the encrypted packets within a single user session or authentication of the server (cost 1 per session)
Active Operations: Issue an HTTP request and receive a response. (cost 1 per operation)

Oracle Query: query the oracle with a candidate private key to get a binary response (yes/no) as to whether the candidate key is the server's private key (cost 1 per query)

Operational Budget:

Max number of operations : 60000
Probability of success : 95%

<u>Question 009 Null SC Time</u>

Challenge Program:

gabfeed_5.jar

Challenge Question:

Is there a side channel in time in the challenge program with which a third party can identify any search term searched by a user within the operational budget?

Additional background for question:

For the purpose of the engagement assume that if a user will execute a search it will be one of their first three actions after logging in.

Available Operations:

Passive Operations: observe the encrypted request and corresponding response packets and timings thereof for a single user http action (cost: 1 per observed action)
Active Operations: Issue an HTTP request and receive a response. (no cost per request)
Oracle queries: query the oracle with a candidate search term to get a binary response (yes/no) as to whether the candidate search term is the user's 'special' search term. (cost: 1 per query)

Operational Budget:

Max number of operations : 5 operations
Probability of success : 90%

<u>Question 014 Null AC Time</u>

Challenge Program:

gabfeed_5.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

GabFeed is a web based application that supports multiple users, however, this question assumes
there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 400000 bytes

Resource Usage Limit

Runtime: 600 seconds measured as the time interval between sending the last byte of the attack and
the server sending the last byte of the response

1.2 SnapBuddy

SnapBuddy is a web application for image sharing. Users can login to SnapBuddy to upload photos to share with their friends. Photos can be marked public in which case any user of the system can view them (all profile photos are public). Photos that aren't marked public should only be viewable by the owner and their friends. In addition to uploading photos, users can apply filters to their photos to change how they look. Finally, users are able to have the server tell them their current location (similar to how observed Wi-Fi access points can be used to determine a user's location). Once their location is set, they can see the people around them in that location (location changes can only occur a certain number of times per day), and their neighbors can see them.

1.2.1 SnapBuddy 1

1.2.1.1 Vulnerabilities

SC Space

In SnapBuddy 1, users have the ability to set their location by providing a set of BSSIDs to the SnapBuddy server. SnapBuddy 1 contains a fixed number of locations each of which has a unique number of BSSIDs associated with it. The BSSIDs are padded to 16 bytes, therefore the size of each post request can be mapped directly to a location. If there are any collisions, the response size containing the location name can be used to resolve the collision. The challenge program description states that users proceed to https://localhost:8080/login and log in with valid credentials. Once a user is signed in the user will set their location and confirm their location. From this description it is evident that all users will set their location as their second operation after proceeding to the SnapBuddy login page. Using this information an attacker can determine which packet represents a user setting his/her location, and the location of the user based on the size of the packet.

AC Space

In SnapBuddy users can add filters to their photos. Each filter operation uses memory; in order to prevent users from exceeding the resource usage limit, there are a limited number of filters which users can use and users cannot apply duplicate filters. The duplicate filter check for the 'ScaleUp' filter is case sensitive, an attacker can specify the same filter multiple time by using a different combination of cases for each letter in the command. Using this method the attacker can exceed the resource usage limit.

1.2.1.2 Positive Questions

Question 25 AC Space

Challenge Program:

snapbuddy_1.jar

Challenge Question:

Is there an algorithmic complexity in space that would cause the challenge program's memory usage to exceed the resource usage limit in one user session given the input budget?

Additional background for question:

SnapBuddy is a web based application that supports multiple users, however, this question assumes there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 10000 bytes

Resource Usage Limit:

Available Memory: 512 MB (total memory usage of the JVM shown under the RSS column in pmap -x <PID>)

Question 31 SC Space

Challenge Program:

snapbuddy_1.jar

Challenge Question:

Is there a side channel in space in the challenge program from which a third party can identify an observed user's location?

Additional background for question:

None

Available Operations:

Passive Operations: observe the encrypted packets and timings thereof within a single user session (cost: 1 per session)
Active Operations: issue an HTTP request and receive a response (no cost per query)
Oracle queries: query the oracle with a candidate user location to get a binary response (yes/no) as to whether the candidate user location is the user's location. (cost: 1 per query)

Operational Budget:

Max number of operations : 5 operations
Probability of success : 99%

1.2.1.3 Null Questions

Question 12 Null SC Time

Challenge Program:

snapbuddy_1.jar

Challenge Question:

Is there a side channel in time in the challenge program from which a third party can identify any one (or more) of the user's friends?

Additional background for question:

None

Available Operations:

Passive Operations: observe the encrypted request and corresponding response packets and timings thereof for a single user http action (cost: 1 per observed action)
Active Operations: Issue an HTTP request and receive a response. (no cost per request)
Oracle queries: query the oracle with a candidate user/friend pair to get a binary response (yes/no) as to whether the candidate user/friend pair is valid. (cost: 1 per query)

Operational Budget:

Max number of operations : 5 operations
Probability of success : 99%

1.2.2 SnapBuddy 2

1.2.2.1 Vulnerabilities

SC Time

SnapBuddy 2 contains the same SC Time vulnerability as GabFeed 1.

SC Space

In SnapBuddy 2, every user's profile photo is public. When a user navigates to the 'Send Invitation' link, the user can see the profile photos of each SnapBuddy user. The size of each profile photo in SnapBuddy can be mapped to a user. If there are collisions, the identity of a given user cannot be resolved by the size of the profile photo alone. The operational budget for the SC Space allows for enough oracle queries to resolve any collisions.

1.2.2.2 Positive Questions

Question 015 SC Time

Challenge Program:

snapbuddy_2.jar

Challenge Question

Is there a side channel in time in the challenge program from which a third party can discover the server's private key used in the server's authentication? Note: This is referring to the private key for authentication at https://localhost:8080 not the private key for establishing an SSL connection.

Additional background for question:

For the sake of this problem, a user session is defined as the user interactions between (and including) the establishment of a session key to the renegotiation of the next session key. The session key is related to an SSL session and is not related to the server's private key.

Available Operations:

Passive Operations: observe the encrypted packets within a single user session or authentication of the server (cost 1 per session)
Active Operations: Issue an HTTP request and receive a response. (cost 1 per operation)
Oracle Query: query the oracle with a candidate private key to get a binary response (yes/no) as to whether the candidate key is server's private key (cost 1 per query)

Operational Budget:

Max number of operations : 60000
Probability of success : 95%

Question 028 SC Space

Challenge Program:

snapbuddy_2.jar

Challenge Question:

Is there a side channel in space in the challenge program from which a third party can identify any one (or more) of the user's friends?

Additional background for question:

None

Available Operations:

Passive Operations: observe the encrypted packets and timings thereof within a single user session (cost: 1 per session)
Active Operations: issue an HTTP request and receive a response (no cost per query)
Oracle queries: query the oracle with a candidate user/friend pair to get a binary response (yes/no) as to whether the candidate user/friend pair is valid. (cost: 1 per query)

Operational Budget:

Max number of operations : 5 operations
Probability of success : 99%

1.2.2.3 Null Questions

Question 002 Null SC Space

Challenge Program:

snapbuddy_2.jar

Challenge Question

Is there a side channel in space in the challenge program from which a third party can learn information that allows them to impersonate any given user?

Additional background for question:

For the sake of this problem, a user session is defined as the user interactions between (and including) the establishment of a session key to the regnegotiation of the next session key. The session key is related to an SSL session and is not related to the server's private key.

Available Operations:

Passive Operations: observe the encrypted packets and timings thereof within a single user session (cost: 1 per session)
Active Operations: issue an HTTP request and receive a response (no cost per query)
Oracle queries: query the oracle to get a binary response (yes/no) as to whether or not you have collected a valid secret from any given user such that you can impersonate that user (cost 1 per query)

Operational Budget:

Max number of operations : 10000
Probability of success : 95%

Question 035 Null AC Space

Challenge Program:

snapbuddy_2.jar

Challenge Question:

Is there an algorithmic complexity in space that would cause the challenge program's memory usage to exceed the resource usage limit in one user session given the input budget?

Additional background for question:

SnapBuddy is a web based application that supports multiple users, however, this question assumes there is only one user interacting with the system in any vulnerability demonstration.

For the sake of this problem, a user session is defined as the user interactions between (and including) the establishment of a session key to the renegotiation of the next session key. The session key is related to an SSL session and is not related to the server's private key.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 10000 bytes

Resource Usage Limit:

Available Memory: 512 MB (total memory usage of the JVM shown under the RSS column in pmap -x <PID>)

1.2.3 SnapBuddy 3

1.2.3.1 Vulnerabilities

AC Time

Similar to SnapBuddy 1, users can add filters to their photos and each filter operation take time. In SnapBuddy 3, the web page limits the number of filters which users can apply. The 'POST' handler does not impose this restriction, thus an attacker can make a 'POST' and add many filters to a given image and exceed the resource usage limit.

1.2.3.2 Positive Questions

Question 022 AC Time

Challenge Program:

snapbuddy_3.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

SnapBuddy is a web based application that supports multiple users, however, this question assumes there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the HTTP requests sent from the user to the server: 10000 bytes

Resource Usage Limit:

Runtime: 300 seconds measured as the time interval between sending the last byte of the attack and
the server sending the last byte of the response

1.2.3.3 Null Questions

Question 018 Null SC Time

Challenge Program:

snapbuddy_3.jar

Challenge Question

Is there a side channel in time in the challenge program from which a third party can learn information that allows them to impersonate any given user?

Additional background for question:

For the sake of this problem, a user session is defined as the user interactions between (and including) the establishment of a session key to the regnegotiation of the next session key. The session key is related to an SSL session and is not related to the server's private key.

Available Operations:

Passive Operations: observe the encrypted packets and timings thereof within a single user session (cost: 1 per session)
Active Operations: issue an HTTP request and receive a response (no cost per query)
Oracle queries: query the oracle to get a binary response (yes/no) as to whether or not you have collected a valid secret from any given user such that you can impersonate that user (cost 1 per query)

Operational Budget:

Max number of operations : 10000
Probability of success : 95%

1.3 TextCrunchr

TextCrunchr is a text analysis program. It can perform text analysis (word frequency, word length, etc...) and includes other utilities (automatic enigma encryption of input). TextCrunchr can process plain text files and can handle compressed files as well (which it will uncompress to analyze the contents).

1.3.1 TextCrunchr 1

1.3.1.1 Vulnerabilities

AC Time

TextCrunchr 1 is vulnerable to a zip bomb. The ZipDecompressor will continuous unzip data from an archive without a limit on the depth of the archive. Given an input zip file that reproduces itself after zipping the resource usage limit can be exceeded.

1.3.1.2 Positive Questions

Question 021 AC Time

Challenge Program:

textcrunchr_1.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file (or files): 400000 bytes

Resource Usage Limit

Runtime: 300 seconds

1.3.2 TextCrunchr 2

1.3.2.1 Vulnerabilities

AC Time

TextCrunchr 2 contains a Hash Table which is vulnerable to hash collisions. A compressed file input with enough word collisions can exceed the resource usage limit.

1.3.2.2 Positive Questions

Question 037 AC Time

Challenge Program:

textcrunchr_2.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file (or files): 400000 bytes

Resource Usage Limit:

Runtime: 300 seconds

1.3.3 TextCrunchr 3

1.3.3.1 Vulnerabilities

AC Time

TextCrunchr 3 has a sorting algorithm which typically has O (n log n) behavior. Certain inputs can cause the behavior to become O (n^2). The vulnerability is triggered via quicksort; provided an input which is shuffled in such a way as to cause the list to be sorted prior to the quicksort call.

1.3.3.2 Positive Questions

Question 011 AC Time

Challenge Program:

textcrunchr_3.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file (or files): 400000 bytes

Resource Usage Limit:

Runtime: 300 seconds

1.3.4 TextCrunchr 5

1.3.4.1 Vulnerabilities

TextCrunchr 5 contains no intended vulnerabilities.

1.3.4.2 Null Questions

<u>Question 017 Null AC Space</u>

Challenge Program:

textcrunchr_5.jar

Challenge Question:

Is there an algorithmic complexity in space that would cause the challenge program's memory usage to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file (or files): 400000 bytes

Resource Usage Limit:

Available Memory: 512 MB (total memory usage of the JVM shown under the RSS column in pmap -x <PID>)

Question 024 Null AC Time

Challenge Program:

textcrunchr_5.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file (or files): 400000 bytes

Resource Usage Limit:

Runtime: 300 seconds

1.3.5 TextCrunchr 6

1.3.5.1 Vulnerabilities

AC Time

TextCrunchr 6 has a sorting algorithm which normally has O (n log n) behavior. Provided an input list which has length divisible by 3, the behavior of the sort becomes O (n^2).

1.3.5.2 Positive Questions

Question 008 AC Time

Challenge Program:

textcrunchr_6.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file (or files): 400000 bytes

Resource Usage Limit:

Runtime: 300 seconds

1.3.6 TextCrunchr 7

1.3.6.1 Vulnerabilities

AC Time

TextCrunchr 7 uses a hash table which is based on a red black tree. The tree is not balance on puts, therefore the challenge program is vulnerable to DOS. Provided an input size file with enough collisions the resource usage limit can be exceeded.

1.3.6.2 Positive Questions

Question 033 AC Time

Challenge Program:

textcrunchr_7.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file (or files): 400000 bytes

Resource Usage Limit:

Runtime: 300 seconds

2 Raytheon/BBN Technologies

2.1 Blogger

The Blogger challenge program is a java web server which serves Plain Old Java Objects (POJOs), and POJOs that collectively implement a simple public blogging application.

2.1.1 Vulnerabilities

AC Time

The method which processes the arguments in a provided URL is vulnerable to AC Time attacks. The method implements a regular expression parser which provided an input of repeated characters will exceed the resource usage limit.

2.1.2 Positive Questions

Question 040 AC Time

Challenge Program:

nanohttpd-javawebserver-2.2.0-SNAPSHOT-jar-with-dependencies.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the servers' response time (on the reference platform) to an HTTP request to exceed the resource usage limit given the input budget?

Additional background for question:

The server can support mutliple users but for this question assumes there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the TCP requests sent from the user to the server: 5000 bytes

Resource Usage Limit:

Runtime: 300 seconds measured as the time interval between sending the last byte of the attack
and the server sending the last byte of the response.

2.1.3 Null Questions

Question 016 Null AC Space

Challenge Program:

nanohttpd-javawebserver-2.2.0-SNAPSHOT-jar-with-dependencies.jar

Challenge Question:

Is there an algorithmic complexity in space that would cause the challenge program's memory usage to exceed the resource usage limit given the budget?

Additional background for question:

The server can support mutliple users but for this question assumes there is only one user interacting with the system in any vulnerability demonstration.

Input Budget:

Maximum sum of the PDU sizes of the TCP requests sent from the user to the server: 5000 bytes

Resource Usage Limit:

Available Memory: 512 MB (total memory usage of the JVM shown under the RSS column in pmap -x <PID>)

2.2 Graph Analyzer

This challenge program read in an input graph as a dot file and translates it to either a PNG file or a PostScript file.

2.2.1 Vulnerabilities

<u>AC Time</u>

The Graph Analyzer challenge program is vulnerable to a “billion lols” style attack. The application accepts an input of a dot file which references graphs by graph names and allows nodes and edges to be created for each graph. Nodes can be provided properties such as type and color. The vulnerability within the challenge program is that a graph can be represented by a node if node’s type contains the key word “container:” followed by the referenced graph’s graph name. A malicious user can provide and input file which exploits this feature to create a very large graph. This can cause the resource usage limit to be exceeded.

<u>AC Space</u>

The AC Time vulnerability described above can be used to launch an AC Space attack if the user specifies a PNG output file.

2.2.2 Positive Questions

<u>Question 020 AC Time</u>

Challenge Program:

GraphDisplay.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file (or files): 5000 bytes

Resource Usage Limit:

Runtime: 1 hour measured from the submission of the last byte of input to the completion of the command execution.

Question 034 AC Space

Challenge Program:

GraphDisplay.jar

Challenge Question:

Is there an algorithmic complexity in space that would cause the challenge program to output a file with a logical size that exceeds the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file: 5000 bytes

Resource Usage Limit:

Available Logical Size: 400 MB (virtual size of outputfile measured with 'stat')

2.2.3 Null Questions

Question 013 Null AC Space

Challenge Program:

GraphDisplay.jar

Challenge Question:

Is there an algorithmic complexity in space that would cause the challenge program's memory usage to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input file: 5000 bytes

Resource Usage Limit:

Available Memory: 512 MB (total memory usage of the JVM shown under the RSS column in pmap -x <PID>)

2.3 Image Processor

The application allows users to train an image processor using a fixed set of training images and submit an image to the application for identification.

2.3.1 Vulnerabilities

AC Time

Prior to classifying a provided image, the challenge program intensifies the image. In the intensify function the challenge program iterates through each pixel and applies a function to the pixel. The function applies runs operations based on the RGB value of a particular pixel. Certain RGB values can cause the resource usage limit to be exceeded.

2.3.2 Positive Questions

Question 005 AC Time

Challenge Program:

ipchallenge-0.1.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the real runtime of the challenge program (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input files including both training and classification images submitted: 70000 bytes

Resource Usage Limit:

Runtime: 1080 seconds measured from the submission of the last user command to the completion
of the application's processing of the command.

2.3.3 Null Questions

Question 030 Null AC Space

Challenge Program:

ipchallenge-0.1.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in space that would cause the memory usage of the challenge program to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Total size of input files including both training and classification images submitted: 70000 bytes

Resource Usage Limit:

Available Memory: 2048 MB (total memory usage of the JVM shown under the RSS column in pmap -x <PID>)

2.4 Law Enforcement Database

The Law Enforcement Employment Database Server is a network service application that provides access to records about law enforcement personnel. Employee information is referenced with a unique employee ID number. The database contains restricted and unrestricted employee information. The ID numbers of law enforcement personnel working on clandestine activities is restricted information.

2.4.1 Vulnerabilities

SC Time

When a user runs the search command, the challenge program creates a list contains all of the keys (public and private) contained within the requested interval. The challenge program iterates through this list, and, for each key determines if the key is public, and, if so, returns a UDP packet containing the key before processing the next key in the list. If the key is private, additional processing overhead is incurred before processing any further public key(s) (and no UDP packet is returned for the private key). The timing differences between subsequent packets containing public keys can be used to determine whether or not a given gap between two public keys contains a secret key. A malicious user can use this information to identify an interval which contains a secret key and use insert operations to run a binary serve over the interval.

2.4.2 Positive Questions

Question 026 SC Time

Challenge Program:

DistributedStore.jar

Challenge Question:

Is there a side channel in time in the challenge program from which a third party can determine the value of any single Restricted ID currently in the provided database?

Additional background for question:

None

Available Operations:

Passive Operations: None
Active Operations: Issue a UDP request to the server and receive a response.
(cost 1 per operation)
Oracle Query: query the oracle with a candidate ID and receive a
binary response (yes/no) as to whether the candidate ID is a

restricted ID. (cost 1 per query)

Operational Budget:

Max number of operations : 500
Probability of success : 95%

2.4.3 Null Questions

Question 019 Null AC Time

Challenge Program:

DistributedStore.jar

Challenge Question:

Is there an algorithmic complexity vulnerability in time that would cause the challenge program's real runtime (on the reference platform) to exceed the resource usage limit given the input budget?

Additional background for question:

None

Input Budget:

Maximum sum of the PDU sizes of the UDP requests sent from the user to the server: 1000 bytes

Resource Usage Limit:

Runtime: 30 seconds measured as the time interval between sending the last byte of the attack and the server sending the last byte of the response.

Question 027 Null SC Space

Challenge Program:

DistributedStore.jar

Challenge Question:

Is there a side channel in space in the challenge program from which a third party can determine the value of any single Restricted ID currently in the provided database?

Additional background for question:

None

Available Operations:

Passive Operations: None
Active Operations: Issue a UDP request to the server and receive a response. (cost 1 per operation)
Oracle Query: query the oracle with a candidate ID and receive a binary response (yes/no) as to whether the candidate ID is a restricted ID. (cost 1 per query)

Operational Budget:

Max number of operations : 50
Probability of success : 95%

2.5 SubSpace

SubSpace is an anonymous message application. The application allows users to register and set their location in longitude and latitude. When a registered user sends a ‘hail’ the challenge program sends a message to the nearest user.

2.5.1 Vulnerabilities

SC Space

The challenge program allows users to set their location without a restriction on the number of location changes a user can do within a certain period. By creating a set of 2 to 5 users, an attacker can move the users around the grid and triangulate the location of other users within the SubSpace application.

2.5.2 Positive Questions

Question 042 SC Space

Challenge Program:

Subspace.jar

Challenge Question:

Is there a side channel in space in the challenge program from which a third party can discover the location of any user using the Subspace application?

Additional background for question:

Assume that users will not change their location during the attack.

Available Operations:

Passive Operations: None
Active Operations: issue an HTTP request and receive a response (1 cost per query)
Oracle queries: query the oracle with a candidate location to get a binary response (yes/no) as to whether there is a user at that location. (cost: 1 per query)

Operational Budget:

Max number of operations : 2500 Operations
Probability of success : 90% (where success is defined as correctly identifying a user's location to 1e-4 degrees latitude and 1e-4 degrees longitude)

2.5.3 Null Questions

Question 041 Null SC Time

Challenge Program:

Subspace.jar

Challenge Question:

Is there a side channel in time in the challenge program from which a third party can learn information that allows them to impersonate any given user?

Additional background for question:

For the sake of this problem, a user session is defined as the user interactions between (and including) the establishment of a session key to the renegotiation of the next session key. The session key is related to an SSL session.

Available Operations:

Passive Operations: observe the encrypted packets within a single user session (cost 1 per session)
Active Operations: Issue an HTTP request to the server and receive a response. (cost 1 per operation)
Oracle Query: query the oracle to get a binary response (yes/no) as to whether or not you have collected a valid secret from any given user such that you can impersonate that user (cost 1 per query)

Operational Budget:

Max number of operations : 100,000
Probability of success : 95%