# Null Dereference Development Mitigation SOP

Null pointer exceptions usually occur when one or more of the programmer's assumptions is violated. A dereference-after-store error occurs when a program explicitly sets an object to null and dereferences it later. This error is often the result of a programmer initializing a variable to null when it is declared.

Most null pointer issues result in general software reliability problems, but if attackers can intentionally trigger a null pointer dereference, they can use the resulting exception to bypass security logic or to cause the application to reveal debugging information that will be valuable in planning subsequent attacks.

## Defense Against Null Dereference

Implement careful checks before dereferencing objects that might be null. When possible, abstract null checks into wrappers around code that manipulates resources to ensure that they are applied in all cases and to minimize the places where mistakes can occur.

## Example

@Override  
@Cacheable(value = CacheConstants.SHARED\_CACHE, key="T(company.system.kernel.caching.cache.SharedCacheKeyGenerator).generate(#root.methodName, #root.args)")  
 public List<CodeValue> findMilitaryBranches() {  
 List<ListTypeDTO> miltaryBranches;  
  
 try{  
 miltaryBranches = standardDataBGSWebService.findMilitaryBranches();  
 }catch (MessageException e){  
 throw new ServicesGatewayRuntimeException("Message exception while fetching military branches - ", e);  
 }  
  
 List<CodeValue> branches = new ArrayList<>();  
 for(ListTypeDTO dto : miltaryBranches){  
 CodeValue branch = new CodeValue();  
 branch.setCode(dto.getCd());  
 branch.setName(dto.getNm());  
 branches.add(branch);  
 }  
  
 return branches;  
 }

## Explanation

The method findMilitaryBranches() in StandardDataWebServiceGatewayImpl.java can crash the program by dereferencing a null pointer with branch.setCode(dto.getCd());.

## Recommendation

The solution is to wrap the pointer dereference in a try-catch block so that an exception can be properly handled.

@Override  
@Cacheable(value = CacheConstants.SHARED\_CACHE, key="T(company.system.kernel.caching.cache.SharedCacheKeyGenerator).generate(#root.methodName, #root.args)")  
public List<CodeValue> findMilitaryBranches() {  
 try{  
 List<ListTypeDTO> militaryBranches = standardDataBGSWebService.findMilitaryBranches();  
 List<CodeValue> branches = new ArrayList<>();  
 for (ListTypeDTO dto : militaryBranches) {  
 CodeValue branch = new CodeValue();  
 branch.setCode(dto.getCd());  
 branch.setName(dto.getNm());  
 branches.add(branch);  
 }  
 return branches;  
 }catch (MessageException e){  
 throw new ServicesGatewayRuntimeException("Message exception while fetching military branches - ", e);  
 }  
 }

## Example

In the following code, the programmer explicitly sets the variable foo to null. Later, the programmer dereferences foo before checking the object for a null value.

Foo foo = null;

...

foo.setBar(val);

...

}

## References

<http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/null_dereference_dereference_after_store.html>

<https://www.owasp.org/index.php/Null_Dereference>