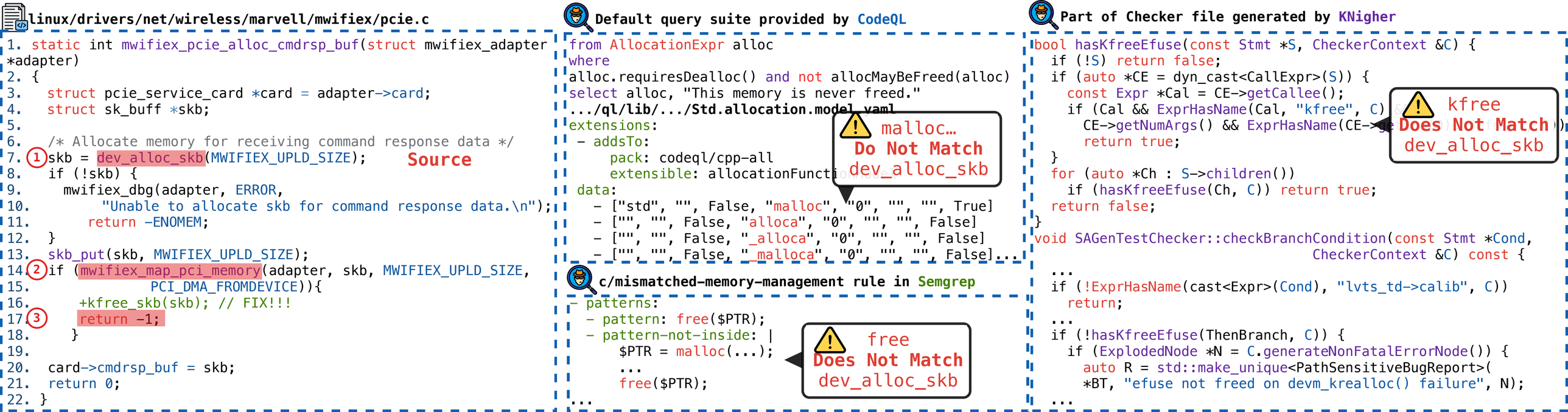
linux/drivers/net/wireless/marvell/mwifiex/pcie.c
1. static int mwifiex_pcie_alloc_cmdrsp_buf(struct mwifiex_adapter *adapter)
2. {
3. struct pcie_service_card *card = adapter->card;
4. struct sk_buff *skb;
5.
6. /* Allocate memory for receiving command response data */
7. 1 skb = dev_alloc_skb(MWIFIEX_UPLD_SIZE);
Source
8. if (!skb) {
9. mwifiex_dbg(adapter, ERROR,
10. "Unable to allocate skb for command response data.\n");
11. return -ENOMEM;
12. }
13. skb_put(skb, MWIFIEX_UPLD_SIZE);
14. 2 if (mwifiex_map_pci_memory(adapter, skb, MWIFIEX_UPLD_SIZE,
15. PCI_DMA_FROMDEVICE)){
16. +kfree_skb(skb); // FIX!!!
17. 3 return -1;
18. }
19.
20. card->cmdrsp_buf = skb;
21. return 0;
22. }
Default query suite provided by CodeQL
from AllocationExpr alloc
where
alloc.requiresDealloc() and not allocMayBeFreed(alloc)
select alloc, "This memory is never freed."
.../ql/lib/.../Std.allocation.model.yaml
extensions:
- addsTo:
pack: codeql/cpp-all
extensible: allocationFunctionModel
data:
- ["std", "", False, "malloc", "0", "", "", True]
- ["", "", False, "alloca", "0", "", "", False]
- ["", "", False, "_alloca", "0", "", "", False]
- ["", "", False, "_malloca", "0", "", "", False]...
malloc…
Do Not Match
dev_alloc_skb
c/mismatched-memory-management rule in Semgrep
- patterns:
- pattern: free($PTR);
- pattern-not-inside: |
$PTR = malloc(...);
...
free($PTR);
...
free
Does Not Match
dev_alloc_skb
Part of Checker file generated by KNigher
bool hasKfreeEfuse(const Stmt *S, CheckerContext &C) {
if (!S) return false;
if (auto *CE = dyn_cast<CallExpr>(S)) {
const Expr *Cal = CE->getCallee();
if (Cal && ExprHasName(Cal, "kfree", C) &&
CE->getNumArgs() && ExprHasName(CE->getArg(0),
return true;
}
for (auto *Ch : S->children())
if (hasKfreeEfuse(Ch, C)) return true;
return false;
}
void SAGenTestChecker::checkBranchCondition(const Stmt *Cond,
CheckerContext &C) const {
...
if (!ExprHasName(cast<Expr>(Cond), "lvts_td->calib", C))
return;
...
if (!hasKfreeEfuse(ThenBranch, C)) {
if (ExplodedNode *N = C.generateNonFatalErrorNode()) {
auto R = std::make_unique<PathSensitiveBugReport>(
*BT, "efuse not freed on devm_krealloc() failure", N);
...
kfree
Does Not Match
dev_alloc_skb