

```

1 contract Bank
2 {
3     struct Payee {
4         address addr;
5         uint256 value;
6     }
7
8     Payee[] payees; // Storage
9     uint256 nextPayeeIndex;
10
11     function payOut() {
12         uint256 i = nextPayeeIndex;
13         while (i < payees.length && msg.gas > 200000)
14         {
15             payees[i].addr.send(payees[i].value);
16             i++;
17         }
18     }
19 }

```

Listing 1: Problematic Contract

```

1 contract Bank
2 {
3     function payOut() {
4         if (!checkGas(nextBlockNo,0))
5             Do something that developer wants;
6
7         uint256 i = nextPayeeIndex;
8         if (!checkGas(nextBlockNo, payees.length))
9             Do something;
10        while (i<payees.length && msg.gas > 200000)
11        {
12            payess[i].addr.send(payees[i].value);
13            i++;
14        }
15    }
16
17    function checkGas(nextBlockNo, itercount)
18    {
19        int["block_count"] gasPriceTable = [19,20000,20,10];
20        if (CFG.isLoop(nextBlockNo))
21        {
22            if (gasPriceTable[nextBlockNo] * itercount) >=
23                gasLeft())
24                return 0;
25        }
26        else
27            if (gasPriceTable[nextBlockNo] >= gasLeft())
28                return 0;
29        return 1;
30    }
31 }

```

Listing 2: Instrumented Contract