

Ivyl sample rootkit 분석 (Analysis of Ivyl sample rootkit)

X90c (정경주) @ isec3.co.kr
<tophackers32@gmail.com>

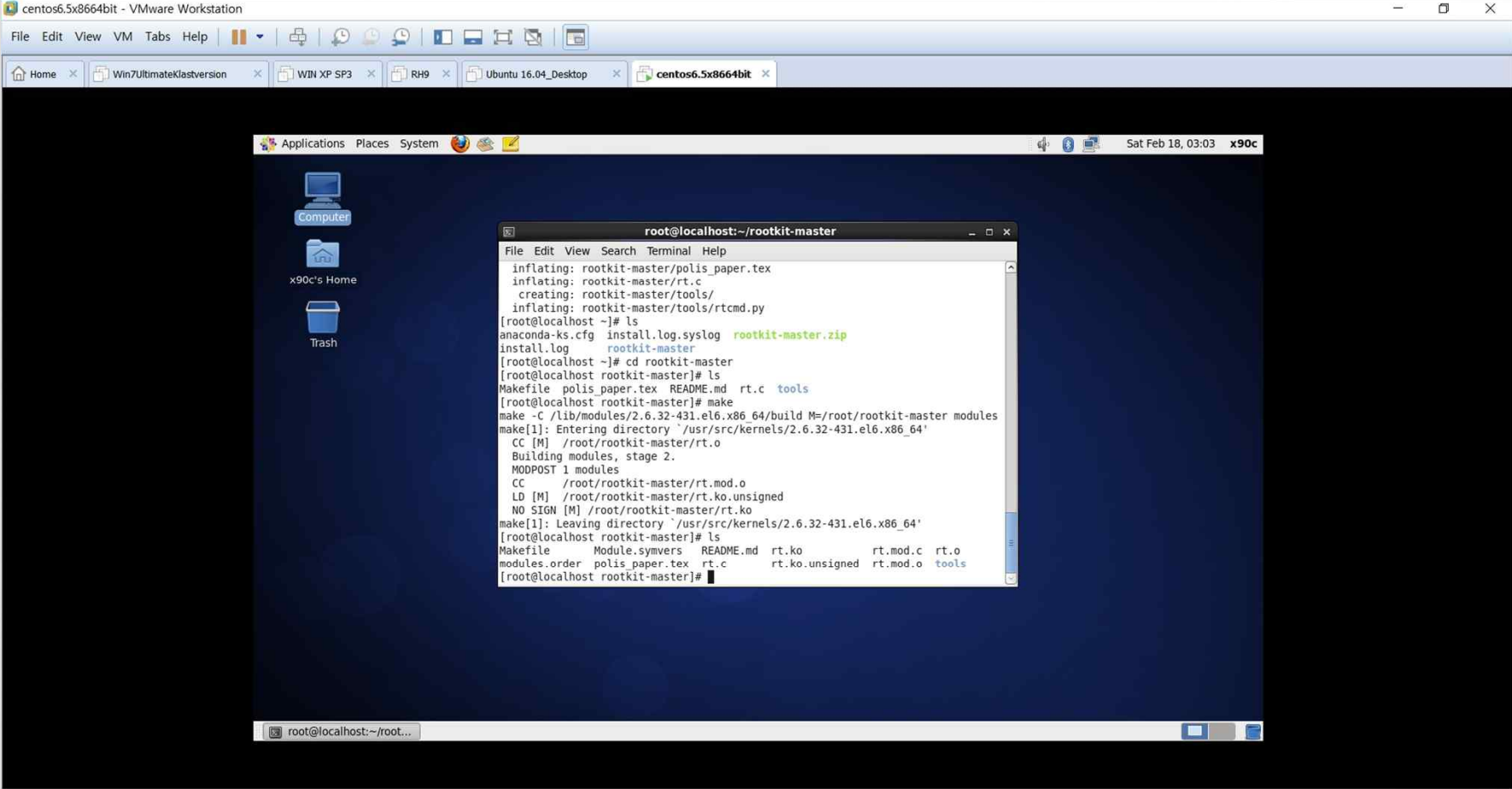
Agenda

- Ivyl sample 리눅스 커널 루트킷?
- 동작 시연 (5)
- 코드 분석
- 코드 분석 (후킹 코드)
- 마침

Iyvl sample 리눅스 커널 루트킷?

- Lkm (loadable kernel module)로 작성된 간단한 리눅스 커널 루트킷 모듈.
- Rt.c (lkm 소스코드)와 tools/rtcmd.py(클라이언트 프로그램)으로 작성되었음.
- Rt.ko를 insmod로 로드하면 lkm을 자동으로 숨기며, lsmod에 나타나지 않음. Rtcmd.py로 커널 루트킷을 제어할 수 있음.
- 특징은 sys_call_table 후킹을 사용하지 않는다는 점이며, procfs나 fs를 초기화 하고 후킹해서 프로세스 하이딩(hide pids)를 구현한다는 점.

동작 시연 (성공적인 빌드)

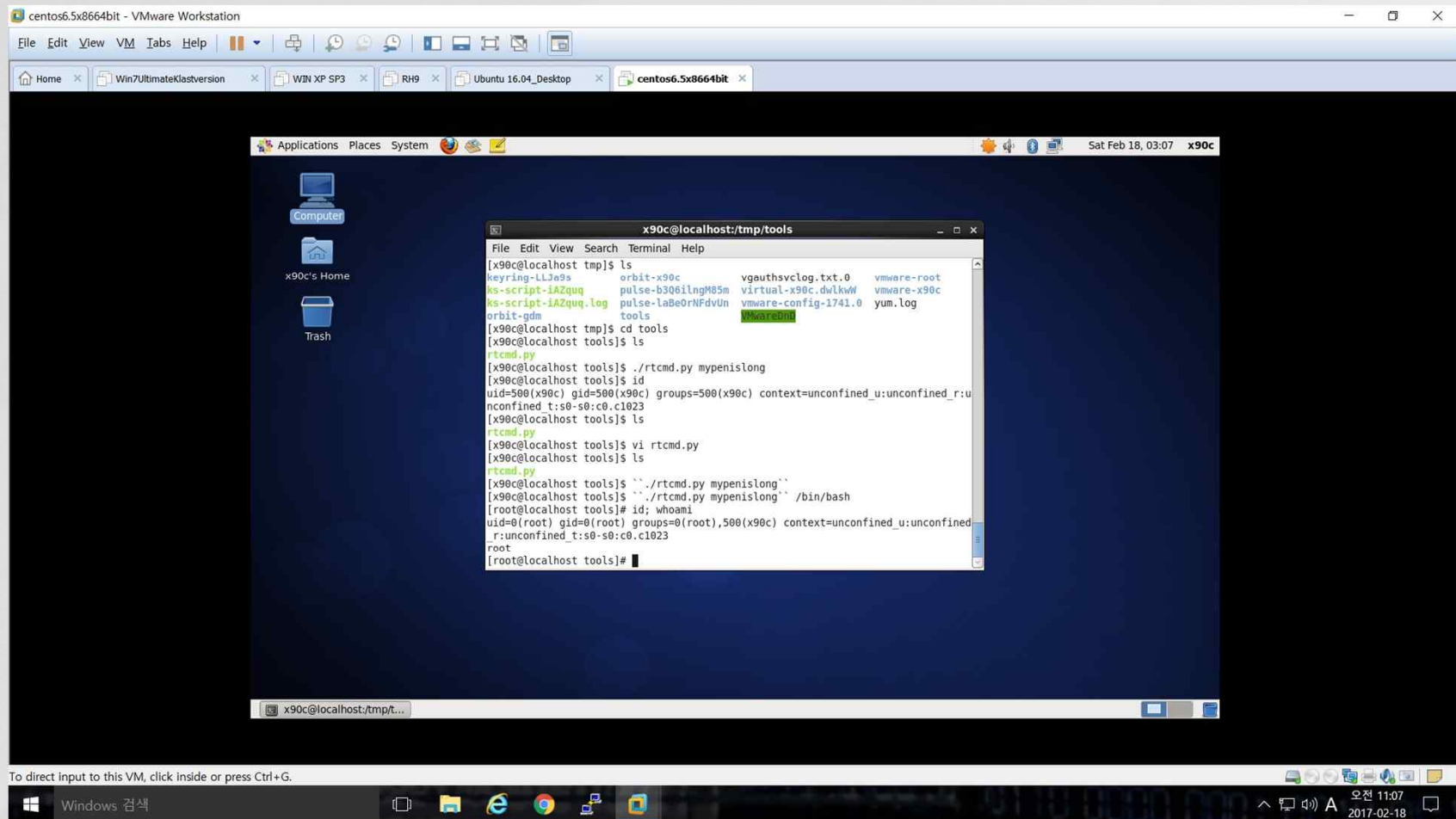


```
centos6.5x86_64bit - VMware Workstation
File Edit View VM Tabs Help
Home Win7UltimateKlsversion WIN XP SP3 RH9 Ubuntu 16.04_Desktop centos6.5x86_64bit
Applications Places System Sat Feb 18, 03:03 x90c
Computer
x90c's Home
Trash
root@localhost:~/rootkit-master
File Edit View Search Terminal Help
inflating: rootkit-master/polis_paper.tex
inflating: rootkit-master/rt.c
creating: rootkit-master/tools/
inflating: rootkit-master/tools/rtcmd.py
[root@localhost ~]# ls
anaconda-ks.cfg install.log.syslog rootkit-master.zip
install.log rootkit-master
[root@localhost ~]# cd rootkit-master
[root@localhost rootkit-master]# ls
Makefile polis_paper.tex README.md rt.c tools
[root@localhost rootkit-master]# make
make -C /lib/modules/2.6.32-431.el6.x86_64/build M=/root/rootkit-master modules
make[1]: Entering directory `/usr/src/kernels/2.6.32-431.el6.x86_64'
CC [M] /root/rootkit-master/rt.o
Building modules, stage 2.
MODPOST 1 modules
CC /root/rootkit-master/rt.mod.o
LD [M] /root/rootkit-master/rt.ko.unsigned
NO SIGN [M] /root/rootkit-master/rt.ko
make[1]: Leaving directory `/usr/src/kernels/2.6.32-431.el6.x86_64'
[root@localhost rootkit-master]# ls
Makefile Module.symvers README.md rt.ko rt.mod.c rt.o
modules.order polis_paper.tex rt.c rt.ko.unsigned rt.mod.o tools
[root@localhost rootkit-master]#
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Windows 검색 2017-02-18 오전 11:03

동작 시연 (루트셸 획득)

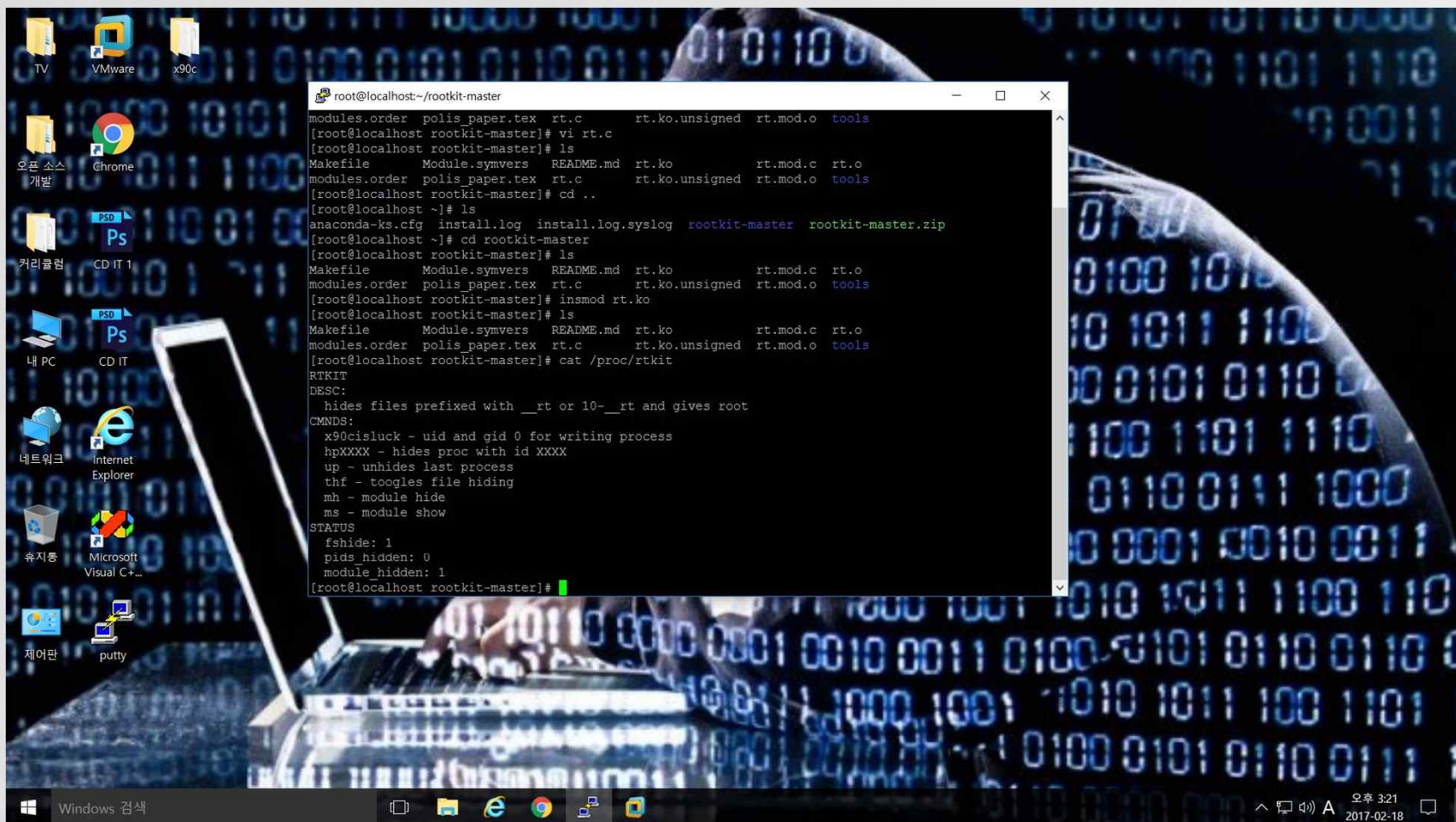


```
centos6.5x8664bit - VMware Workstation
File Edit View VM Tabs Help
Home Win7UltimateKistversion WIN XP SP3 RH9 Ubuntu 16.04_Desktop centos6.5x8664bit
Applications Places System Sat Feb 18, 03:07 x90c
Computer
x90c's Home
Trash
x90c@localhost/tmp/tools
File Edit View Search Terminal Help
[x90c@localhost tmp]$ ls
keyring-LLJe9s orbit-x90c vgaauthsvclog.txt.0 vmware-root
ks-script-iAZquq pulse-b3Q6ilngM85m virtual-x90c.dvLkww vmware-x90c
ks-script-iAZquq.log pulse-laBe0rNFdvUn vmware-config-1741.0 yum.log
orbit-gdm tools VMwareBin
[x90c@localhost tmp]$ cd tools
[x90c@localhost tools]$ ls
rtcmd.py
[x90c@localhost tools]$ ./rtcmd.py mypenislong
[x90c@localhost tools]$ id
uid=500(x90c) gid=500(x90c) groups=500(x90c) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[x90c@localhost tools]$ ls
rtcmd.py
[x90c@localhost tools]$ vi rtcmd.py
[x90c@localhost tools]$ ls
rtcmd.py
[x90c@localhost tools]$ ``./rtcmd.py mypenislong``
[x90c@localhost tools]$ ``./rtcmd.py mypenislong`` /bin/bash
[root@localhost tools]# id; whoami
uid=0(root) gid=0(root) groups=0(root),500(x90c) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
root
[root@localhost tools]#
```

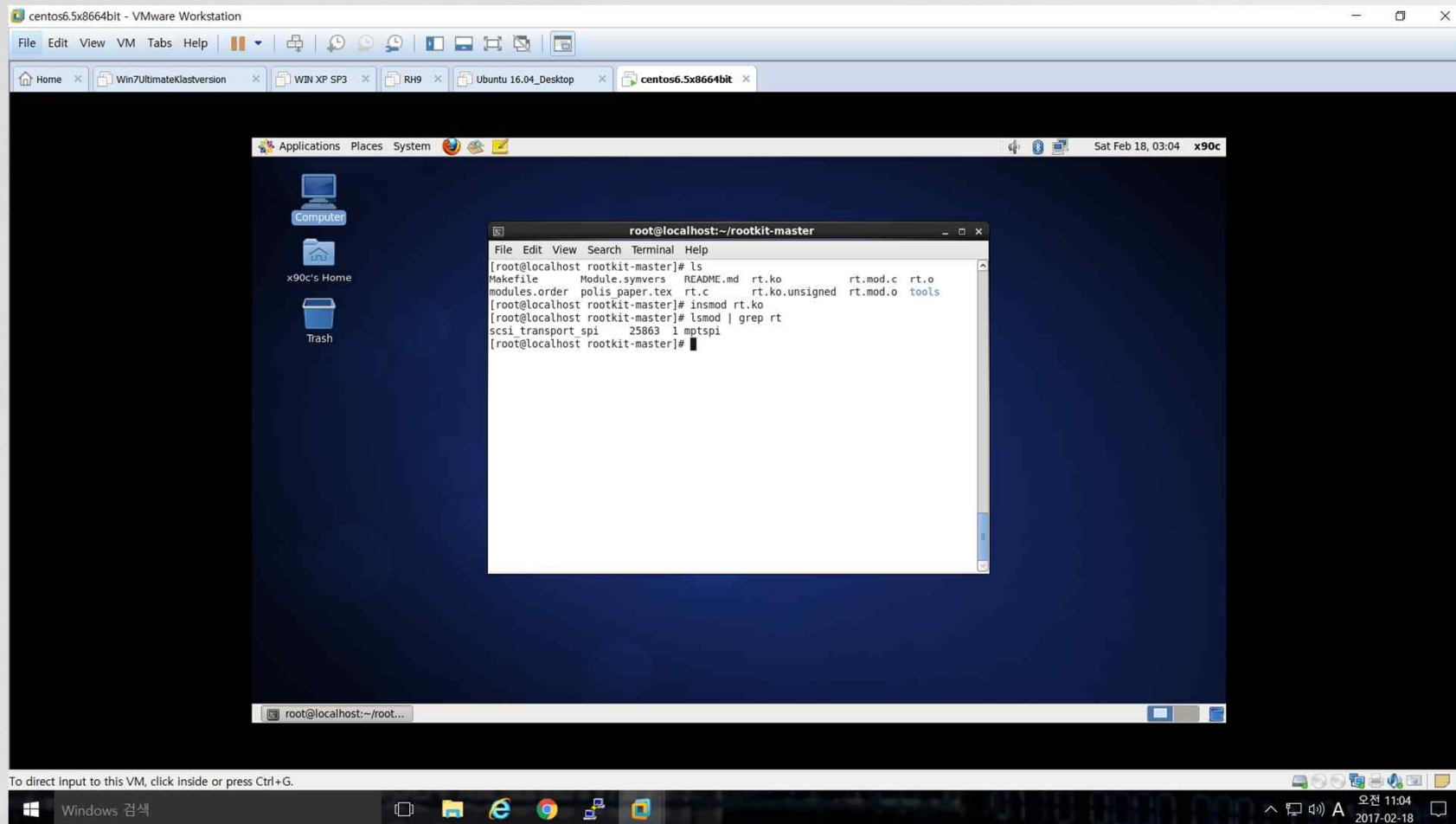
To direct input to this VM, click inside or press Ctrl+G.

Windows 검색 오전 11:07 2017-02-18

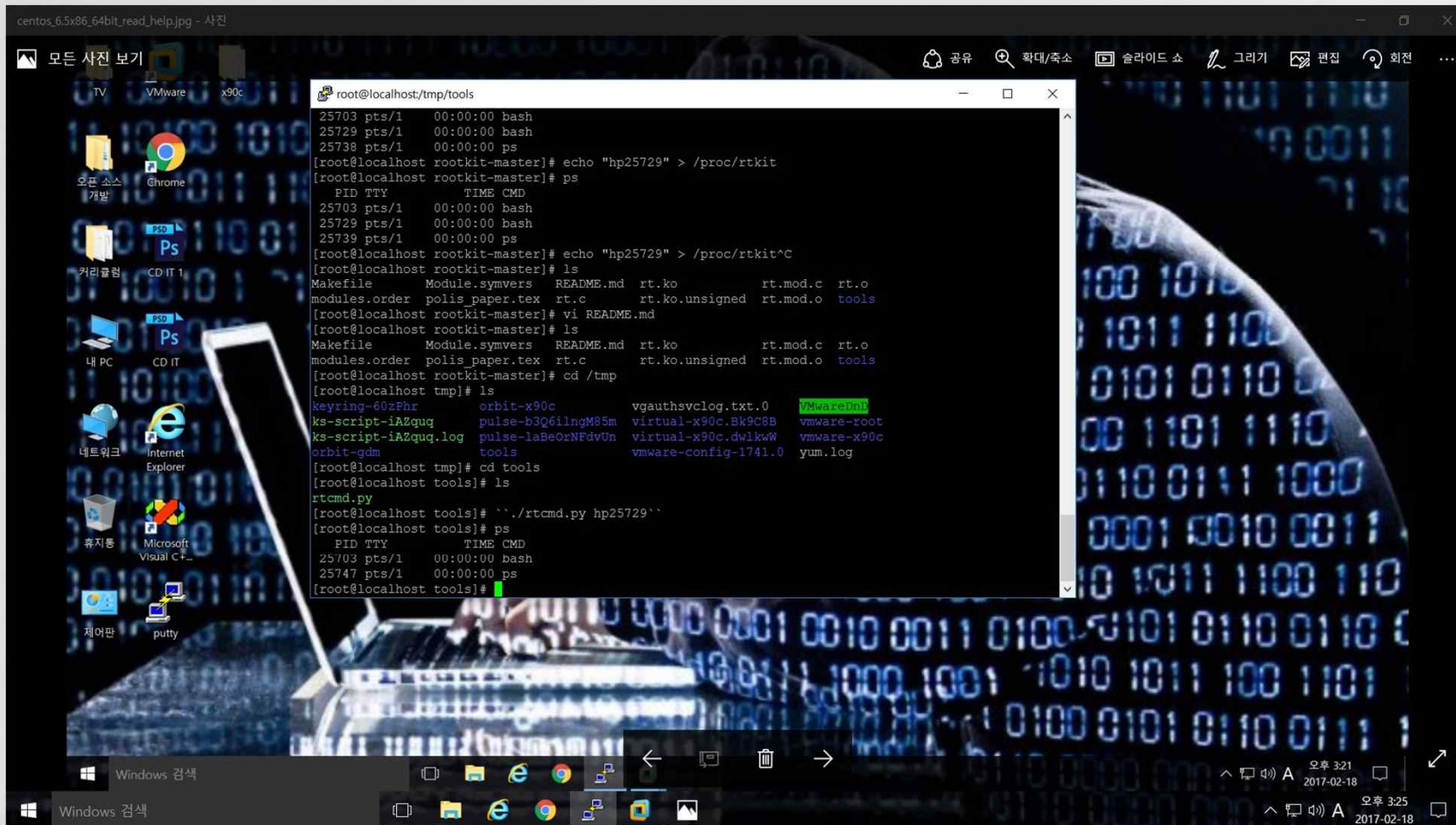
동작 시연 (매뉴얼&동작 상태 보기)



동작 시연 (insmod후 모듈 하이딩됨)



동작 시연 (프로세스 하이딩)



The screenshot shows a Windows desktop environment with a terminal window open. The terminal is running a series of commands to demonstrate process hiding in a Linux environment. The desktop background features a blue and black pattern with binary code (0s and 1s). The terminal window title is 'root@localhost/tmp/tools'. The commands and their outputs are as follows:

```
root@localhost rootkit-master]# echo "hp25729" > /proc/rtkit
root@localhost rootkit-master]# ps
  PID TTY          TIME CMD
 25703 pts/1    00:00:00 bash
 25729 pts/1    00:00:00 bash
 25739 pts/1    00:00:00 ps
root@localhost rootkit-master]# echo "hp25729" > /proc/rtkit^C
root@localhost rootkit-master]# ls
Makefile  Module.symvers  README.md  rt.ko      rt.mod.c  rt.o
modules.order  polis_paper.tex  rt.c      rt.ko.unsigned  rt.mod.o  tools
root@localhost rootkit-master]# vi README.md
root@localhost rootkit-master]# ls
Makefile  Module.symvers  README.md  rt.ko      rt.mod.c  rt.o
modules.order  polis_paper.tex  rt.c      rt.ko.unsigned  rt.mod.o  tools
root@localhost rootkit-master]# cd /tmp
root@localhost tmp]# ls
keyring-60zPhr      orbit-x90c      vgauthsvlog.txt.0  VMware0n
ks-script-iAZquq    pulse-b3Q6ilngM95m  virtual-x90c.Bk9C8B  vmware-root
ks-script-iAZquq.log  pulse-laBeOrNFdvOn  virtual-x90c.dWlkW  vmware-x90c
orbit-gdm           tools           vmware-config-1741.0  yum.log
root@localhost tmp]# cd tools
root@localhost tools]# ls
rtcmd.py
root@localhost tools]# `./rtcmd.py hp25729`
root@localhost tools]# ps
  PID TTY          TIME CMD
 25703 pts/1    00:00:00 bash
 25747 pts/1    00:00:00 ps
root@localhost tools]#
```


동작 시연 (Pros & Cons)

- 앞서 살펴본 스크린샷과 같이 ivyl 리눅스 커널 루트킷은 Centos 6.5 x86 64비트 버전의 배포판에서 빌드에 어려움이 없는 것을
- 알 수 있었으며(장점), 기능들이 정상적으로 동작하는 것을 알 수 있었고, 페도라 코어 6에서 테스트한 결과 빌드 오류가 여러 개 나타나 오류를 해결하기 전엔 사용할 수 없다는 것을 알 수 있어 이 점이 (단점)으로 꼽힌다.

Lkm 동작 과정

- (1) insmod 리눅스 커널 루트킷 모듈
- (2) lkm이 숨겨지고, rtcmd.py를 통해 루트킷 제어.
- (3) 해커 재침입 후, 로딩되어 있는 /proc/rtkit을 rtcmd.py로 제어해서 루트셸 획득 또는 공격에 사용되는 프로세스, 파일 숨김.
- (4) 재부팅 문제 해결을 위해서는 /etc 설정 파일을 통해서 rc.local에 등록해야 하는 단점이 있음. (루트킷이 chkrootkit 등에 탐지될 우려가 있음).
- (5) 로딩된 커널 루트킷 제거는 모듈을 보이게 한 다음에 rmmmod로 제거할 수 있음.

코드 분석

- **모듈 헬퍼 (lkm을 숨기고 보이도록 하는 함수)**
 - module_hide
 - module_show
- **PAGE RW 헬퍼 (후킹을 위해 페이지를 쓰기 전용으로 변경하거나 읽기전용으로 변경하는 함수)**
 - set_addr_rw
 - set_addr_ro
- **콜백 섹션 (프로세스를 pid로 숨기거나 파일을 숨기는 콜백 함수와 read/write를 통한 루트킷 제어 콜백 함수)**
 - proc_filldir_new
 - proc_readdir_new
 - fs_filldir_new
 - fs_readdir_new
 - rtkit_read
 - rtkit_write
- **초기화/클린업 헬퍼 메소드 섹션 (초기화 클린업 함수)**
 - procfs_clean
 - fs_clean
 - procfs_init (프로세스 하이딩을 위한 후킹은 이 함수에서 이루어짐)
 - fs_init (파일 하이딩을 위한 후킹은 이 함수에서 이루어짐)
 - rootkit_init (lkm 초기화 함수)
 - rootkit_exit (lkm 클린업 함수)

코드 분석 (후킹 코드)

- procfs_init() 내 후킹 메커니즘 분석
 - (1) "rtdkit" proc 엔트리 생성
 - (2) proc_root = proc_rtdkit->parent를 설정.
 - (3) proc_root->name이 "/proc"인지 검사.
 - (4) proc_rtdkit->read_proc = rtdkit_read, proc_rtdkit->writeproc = rtdkit_write. (read/write 후킹 함수 설정).
 - (5) **proc_fops = proc_root->proc_fops**
 - (6) **proc_readdir_orig = proc_fops->readdir로 백업.**
 - (7) set_addr_rw(proc_fops) // 페이지 쓰기 권한 획득
 - (8) **proc_fops->readdir = proc_readdir_new (후킹)** 중요 ****
 - (10) set_addr_ro(proc_fops) // 읽기 전용으로 돌려 놓음.
- 위와 같은 방식으로 proc_readdir_new 함수를 proc_fops 즉 /proc의 fops로 사용되게 페이지를 읽기/쓰기 전용으로 변경해 후킹함으로써 루트킷의 일부 프로세스 하이딩 기능 등이 구현된 것을 알 수 있음.

마침

◦ 읽어주셔서 감사합니다.

트위터 팔로워 환영합니다.

<https://twitter.com/x90ctwitt>