Xen – maszyna wirtualna

- Autorzy:
- Piotr Daszkiewicz
- Sławomir Pawlewicz
- Adam Slaski

Co to jest Xen VMM (virtual machine monitor)?

Projekt na licencji open-source, początkowo realizowany w laboratoriach komputerowych uniwersytetu w Cambridge. Przejęty przez firmę XenSource.

Gdzie mam zainstalować?

- Łatki (patches) do wersji 2.4 i 2.6 jądra.
- Łatki do jądra NetBSD/FreeBSD.
- Sun pracuje, aby Xen był również dostępny na ich systemach.

Po co?

Xen umożliwia uruchamianie wielu maszyn wirtualnych, z których każda może uruchomić np. inny system operacyjny.

Dlaczego Xen?

- Pomysł oparty na parawirtualizacji.
- Koszty uruchomienia 3% tego co na sprzęcie fizycznym.
- Przyszłe wbudowane wsparcie procesorów np. Intel Vt i AMD Pacifica.

Różnice między Xen-em, a innymi maszynami wirtualnymi.

- VMware – nie wymaga łatki na system operacyjny gościa (guest OS). Prawdopodobnie wolniejszy. Xen bezpośrednio adresuje do shadow page tables.

Gdzie już jest Xen?

- Fedora Core 4 (w formie RPM-ów)
- Debian
- SuSE Professional 9.3
- (w przyszłości) RHEL5
- Inne dystrybucje również zajmują się przygotowaniem pakietów instalacyjnych.

Dostępne wersje w internecie

- Xen Wersja 2.0
- Xen Wersja 3.0

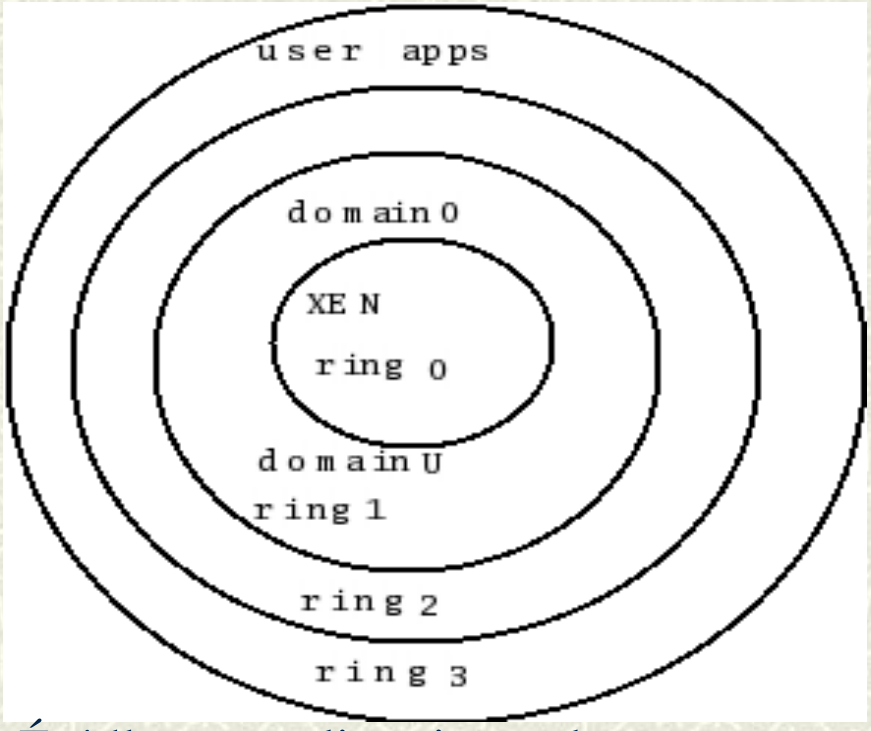
Xen 3.0

Co kryje się w Xen-ie?

- Parawirtualizacja.
- Rozdzielenie wirtualnych urządzeń (split drivers diagram).
- Wsparcie dla Xen-a przez procesory Intel VT-x.
- Live migration.

Parawirtualizacja

- Systemy operacyjne gości (OS guest) uruchamiane z mniej uprzywilejowanego poziomu ochrony. (Uruchamiany nie z poziomu 0).



Źródło: www.linuxjournal.com

Rozdzielenie wirualnych urządzeń

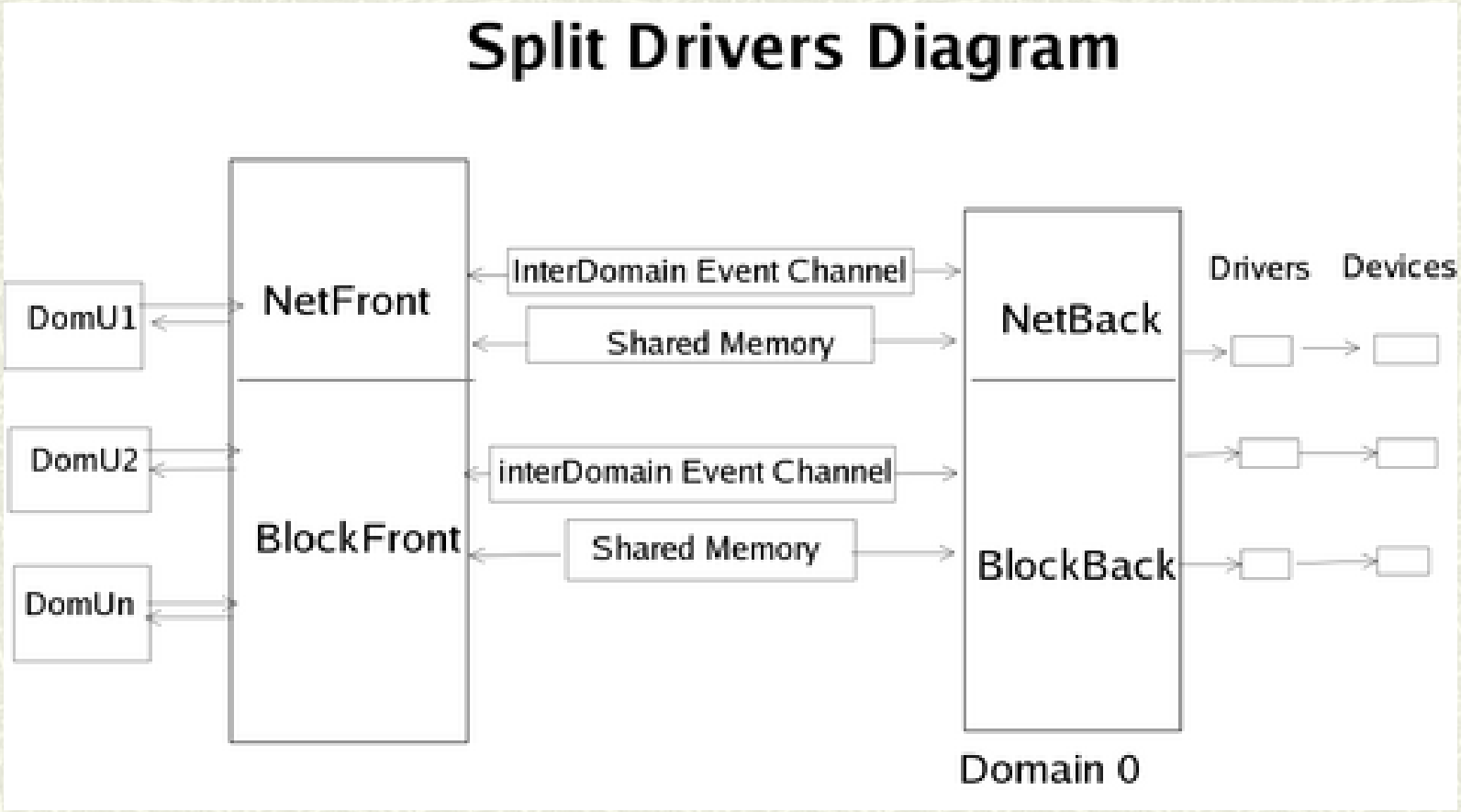
W IA-32 jest zawarte 250 instrukcji. Wywołanie 17 z nich nie jest dozwolone z niezerowego poziomu ochrony.

Np.HLT powoduje general protection fault (GPF).

Komunikacja OS guest z Xen-em.

- OS guest wywołuje tzw. hypercall. Hypercall pozwala na wykonanie uprzywilejowanych instrukcji.
- Hypercalle wysyłane przez Xen (multicall) w strukturze danych entry_t_struct.

Rozdzielenie wirtualnych maszyn.



Źródło: www.linuxjournal.com

Wspólna pamięć

- Istnieje wspólna pamięć pomiędzy wszystkimi domenami, a domeną 0.
- Umożliwia przekazywanie zleceń i danych.

Rozdzielenie wirtualnych urządzeń – schemat komunikacji.

- Domena 0 jako jedyna ma dostęp do sprzętu fizycznego.
- Domena 0 ma również wirtulane urządzenia.
- Nieuprzywilejowane domeny, wysyłają prośbę do interfejsu (fronted).
- Interfejs symuluje dostęp do sprzętu fizycznego. Przekazuje prośbę do interfejsu obsługującego (backend) urządzenia fizyczne.

Intel VT-x – procesor wspierający Xen-a.

- Procesor zawiera 10 nowych instrukcji stworzonych na potrzeby Xen-a.

Np. VMCALL – zakończ pracę wirtualnej maszyny.

VMLAUNCH – uruchom wirtualną maszynę.

Live Migration

- I faza:Pre-copying. Fizyczna pamięć jest kopiowana we wskazane miejsce, podczas gdy migrujące domeny dalej działają.
- Potem przegrywane są tylko te strony, które w międzyczasie były nadpisane. Migrująca domena przestaje działać.
- II faza: Potrzebne strony pamięci są kopiowane, a praca migrujących domen jest wznawiana.

Podsumowanie – też chcę mieć Xen-a.

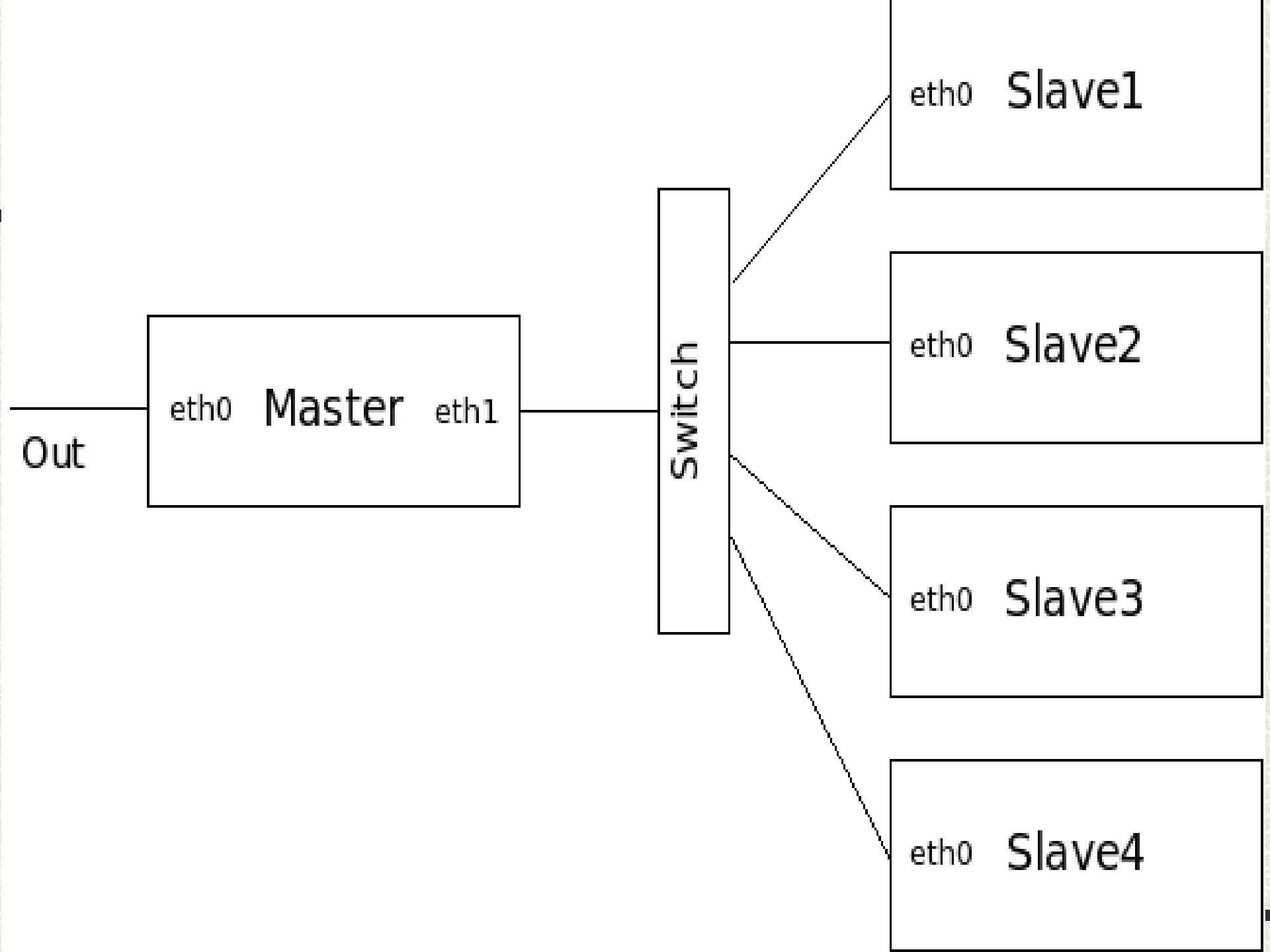
- 64-bitowe procesory Intela będą wspierały wirtualizację. Xen najprawdopobniej będzie najczęściej używanym narzędziem.
- Już teraz jest wspierany przez Intel VT-x, Intel VT-i oraz AMD SVM.
- W przyszłości zintegrowany z oficjalnymi jądrami Linux-a.
- Merytorycznie istotny projekt niosący wiele korzyści oraz z ciekawymi możliwościami.

Część II

Wszyscy lubią symulatory!

- Bolidu F1
- Myśliwca F16

Ale czy ktoś z nas miał już na komputerze symulator klastra obliczeniowego?


eth0 Slave1
eth0 Slave2
eth0 Slave3
eth0 Slave4
Switch
eth0 Master eth1
Out

Co będzie nam potrzebne?

- Świeża instalacja Ubuntu
- Źródła Xena (np. xen-3.0.1-src.tgz)
- GuestOS (np. CentOS z Jailtime.org)
- Sporo RAMu – najlepiej 1GB

Konfigurujemy Xena

- tar -zxf xen-3.0.2-2-src.tgz
- cd xen-3.0.2-2
- make world (sam ściągnie kernel w wersji 2.6.16)
- sudo make install

dodajemy Xena do Gruba

- w pliku /boot/grub/menu.lst dodajemy:

```
title           Xen 3.0 / XenLinux 2.6
kernel            /boot/xen-3.0.gz dom0_mem=262144
module              /boot/vmlinuz-2.6-xen0 root=/dev/sda1 ro console=tty0
max_loop=32
```

Ostatnie poprawki i zaczynamy

- dodajemy xend i xendomains do skryptów startowych
- usuwamy Local Thread Storage (/lib/tls), która nie lubi się z Xenem
- restartujemy komputer
- sudo xm list – Domain0 jest jedyną maszyną działającą w systemie

Tworzymy maszyny wirtualne

- tar -zxf centos.4-3.20060325.img.tgz
  sudo mkdir -p /opt/xen/cray/master (repeat for slave1-4)
  sudo cp centos.swap /opt/xen/cray/master/
  sudo cp centos.4-3.img centos.swap /opt/xen/cray/slave1/ (repeat for slave2-4)
- powiększamy dysk dla Mastera
  dd if=/dev/zero of=/tmp/zero.xen bs=1M count="1024"
  sudo e2fsck -f centos.4-3.img
  cat centos.4-3.img /tmp/zero.xen >> centos.4-3.2GB.img
  resize2fs centos.4-3.2GB.img
  e2fsck -f centos.4-3.2GB.imgsudo cp centos.4-3.2GB.img \
  /opt/xen/cray/master/centos-4-3.img

Edytujemy pliki .cfg

- kernel = "/boot/vmlinuz-2.6-xenU"
  memory = 128
  name = "master"
  vif = ['', '']
  disk = ['file:/opt/xen/cray/master/centos.4-3.img,sda1,w','file:/opt/xen/cra/master/centos.swap,sda2,w']
  root = "/dev/sda1 ro"
- kernel = "/boot/vmlinuz-2.6-xenU"
  memory = 64
  vcpus = 4
  name = "slave1"
  vif = ['']
  disk =['file:/opt/xen/cray/slave1/centos.4-3.img,sda1,w','file:/opt/xen/cray/slave1/centos.swap,sda2,w']
  root = "/dev/sda1 ro"

Konfigurujemy maszyny wirtualne

- sudo mount -o loop /opt/xen/cray/master/centos.4-3.img tmp_img/
- edytujemy ustawienia sieci
- uruchamiamy mastera, konfigurujemy mu nfsa i eksportujemy /home i /cshare do slave'ów
- dla każdego slave'a:
  sudo mount -o loop /opt/xen/vcluster/slave1/centos.4-3.img tmp_img/
- edytujemy ustawienia sieci
- sudo cp portmap-4.0-63.i386.rpm tmp_img/tmp/
- sudo chroot tmp_img/
- rpm -ivh tmp/portmap-4.0-63.i386.rpm
- dodajemy /cshare i /home do fstab jako nfs
- uruchamiamy slave'y

Niech stanie się klaster!

- C3
- Environment Modules
- MPICH
- Torque
- Maui

Garniec miodu

- Nabrawszy doświadczenia w takich zabawach spróbujmy czegoś bardziej przydatnego
- Honeypot to komputer-pułapka udający normalny serwer
- Honeynet to sieć-pułapka
- Dzięki wirtualizacji możemy umieścić je wszystkie na jednej maszynie

Czego potrzeba w dobrym garnku

- przynęty
- haczyka
- gdyby przynętę udało się komuś połknąć, nie zahaczajhąc się, musimy dociec, jak to zrobił, a więc potrzebny jest system monitoringu

DTK

- Deception ToolKit
- kolekcja sktyptów emulujących działanie w systemie różnych usług (np. Sendmaila)
- Sendmail przechowuje fałszywy plik z hasłami, a gdy haker marnotrawi czas rozkodowując go, my namierzamy delikwenta i wysyłamy oddział komandosów

OSH

- Shell, który potrafi ograniczyć wybranym użytkownikom dostęp do niektórych poleceń

snort

- bardzo silny sieciowy system wykrywania ataków
- szeroki zakres mechanizmów detekcji, mogących w czasie rzeczywistym dokonywać analizy ruchu i rejestrowania pakietów w sieciach opartych na protokołach IP/TCP/UDP/ICMP
- Potrafi przeprowadzać analizę strumieni pakietów, wyszukiwać i dopasowywać podejrzane treści, a także wykrywać wiele ataków i anomalii, takich jak przepełnienia bufora, skanowanie portów typu stealth, ataki na usługi WWW, SMB, próby wykrywania systemu operacyjnego i wiele innych

syslog-ng

- Większe bezpieczeństwo zapewnia możliwość użycia protokołu TCP w komunikacji z tzw. loghostem
- Logów zlokalizowanych na innym serwerze włamywacz tak łatwo nie wyczyści

The Coroner's Toolkit (tct)

- Kolekcja narzędzi do analizowania systemu.
- Pozwala między innymi przywrócić usunięte pliki (przez kogoś, kto zacierał po sobie ślady)

TripWire

- pomaga wykryć zmiany w plikach (np. zmiany długości)
- analizując przez dłuższy czas system reportuje nieoczekiwane zmiany

Dziękujemy za uwagę