DEFENSIVE SECURITY

IT SECURITY EDUCATION & SERVICES

Ile warstw, tyle szans.

Defensive-Security.com

Leszek Miś
leszek.mis@defensive-security.com

Leszek Miś

- Founder @ Defensive Security
- Chief Security Architect @ Collective Sense
- Offensive Security Certified Professional
- RHCA/RHCSS/RHCX/Sec+
- Członek ISSA/OWASP Poland
- Skupiam się głównie na:
  - Linux & Network Security
  - Web Application Security
  - Penetration testing
  - Hardened IT Infrastructure (SSO/IdM/IDS)
  - Linux forensics

Agenda

- Stan aktualny
- Zagrożenia
- Potrzeby czyli stan porządany
- Podsumowanie

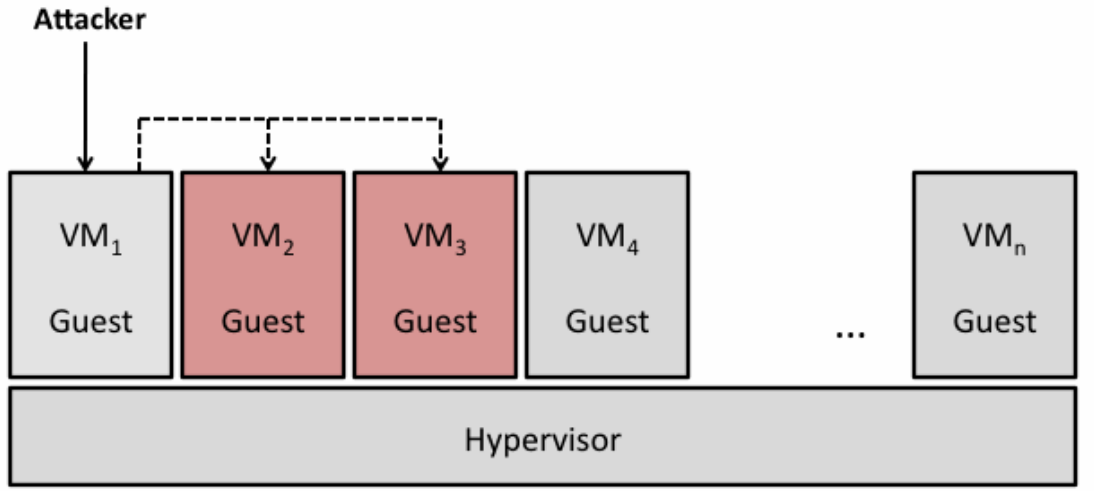
Stan aktualny

- Wygrzana wirtualizacja:
  - Type 1:
    - VMware ESX, Citrix XEN, KVM
  - Type 2:
    - Virtualbox, QEMU, VMware workstation
- Gorąca konteneryzacja:
  - CoreOS Rocket, LXC, Docker
- Środowiska:
  - PaaS
  - SaaS
  - mikroserwisy

Co łączy powyższe technologie?

Zagrożenia

- Escaping from:
  - Guest to guest
  - Guest to host
- Ataki cross-kontenerowe
- DOS:
  - VM crash
  - Host crash
- Ataki na mgmt:
  - API
  - Web apps / restricted shells



Zagrożenia

- Nieograniczone operacje uprzywilejowane:
  - uid=0, CAP_SYS_ADMIN
- Słaba konfiguracja domyślna:
  - Host
  - Network → 0.0.0.0
- Wyciek informacji → procfs, dmesg, sysfs, debugfs, /dev, inne
- Brak ochrony krytycznych elementów systemu → np. LKM / MAC
- Brak ograniczenia dostępu do zasobów:
  - (:(){ :|:& };:) / OOM
  - Random devices
- Brak aktualizacji → nowy kod → nowe błędy
- Kiepskiej jakości kod źródłowy → web apps
- Duże obrazy bazowe

Zagrożenia

- Mnóstwo najróżniejszych CVE:
  - CVE-2014-7975:
  - CVE-2015-2925: double chroot attack
  - CVE-2015-4176
  - CVE-2015-1328: privilege escalation vulnerability when using overlayfs mounts inside of user namespaces
  - CVE-2015-1334: fake procfs for bypassing MAC
  - CVE-2014-6407: symlink attack
  - CVE-2015-1331: directory traversal
  - CVE-2015-3630: /proc/asound manipulation
  - CVE-2015-5165: QEMU: heap overflow in RTL8139 driver
  - CVE-2015-3209: floppy controller (Venom)

Zagrożenia

- Peering into the Depths of TLS Traffic in Real Time.pdf
- Xen Hypervisor VM Escape.pdf
- Virtualization System Vulnerability Discovery Framework.pdf
- Escape From The Docker-KVM-QEMU Machine.pdf

Zagrożenia

- Błędy w tzw. rozwiązaniach „sprzętowych”:
  - Palo Alto Next Generation FW:
    - CLI over SSH → command line injection in SCP connection
    - ApiWgetFilter bypass → PreAuth RCE
    - DOS → /global-protect/login.esp
  - Fireye:
    - Malware Input Processor → email + malicious attachment = Code Execution using JODE
    - Priv Esc → snort autoupdate module

http://conference.hitb.org/hitbsecconf2016ams/materials/D2T1%20-%20Felix%20Wilhelm%20-%20Attacking%20Next%20Generation%20Firewalls.pdf

http://googleprojectzero.blogspot.com/2015/12/fireeye-exploitation-projec

Potrzeby

Potrzeby

- Defense in depth → least privs → utrudnienie ataków
- Izolacja na wielu warstwach:
  - Kernel
  - Host OS
  - Wirtualizacja
  - Konteneryzacja
  - Sieć + usługi sieciowe
  - Aplikacje webowe
  - Użytkownicy, w tym root

Potrzeby - kernel

- Kernel → do_brk(2) , do_remap(2) , vmsplice(2) / snmp, econet, CAN, SCTP, UDP
  - Wyłączenie obsługi modułów:
    - kernel.modules_disabled=1
    - CONFIG_MODULES=n
  - Utwardzona kompilacja – grsecurity+PAX(non-X,non-W):
    - KASLR
    - Address Space Protection
    - Trusted Path Execution
    - Chroot restriction
    - FS restriction
    - RBAC
    - Network protection
  - CONFIG_CC_STACKPROTECTOR_STRONG

Potrzeby – host OS

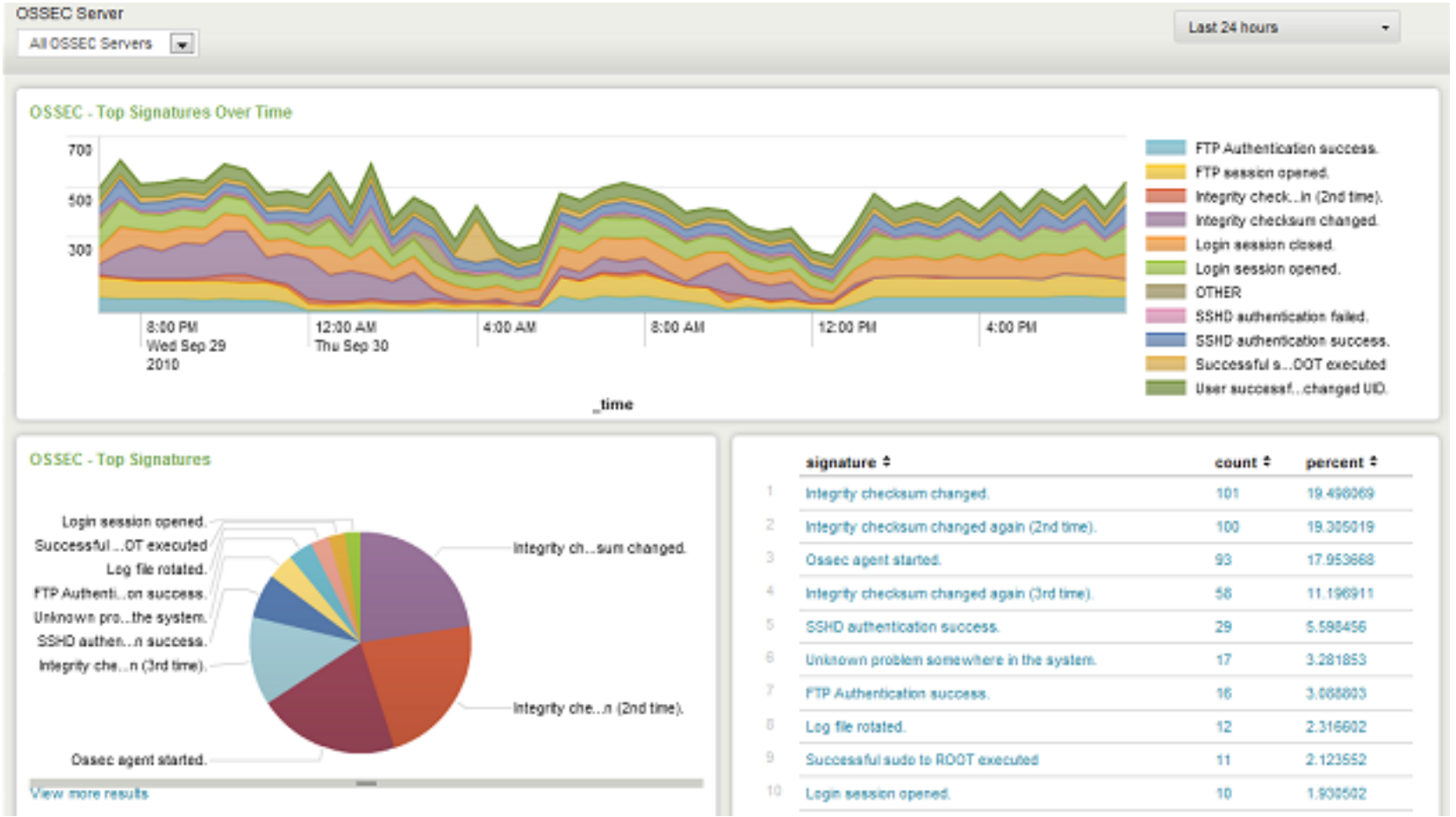
- Separacja uprawnień
- Minimalizm implementacyjny
- Niskopoziomowe audytowanie zdarzeń:
  - auditd, sysdig, stap
- DAC vs MAC → ograniczenie roota
- Multi Category Security
- Okresowa analiza pamięci RAM
- Privchecki:
  - suid, sgid, world writeable dirs, etc.

Potrzeby – host OS

- OSSEC - Host Intrusion Detection System:
  - Wsparcie dla Windows, Linux, Mac OSX, Solaris, HP-UX
  - Funkcjonalność:
    - Analiza logów
    - Integralność systemu plików + IMA/EVM
    - Monitoring zdarzeń systemowych
    - Wykrywanie rootkitów
    - Alertowanie oraz aktywna ochrona

Potrzeby – host OS

- OSSEC - Host Intrusion Detection System:



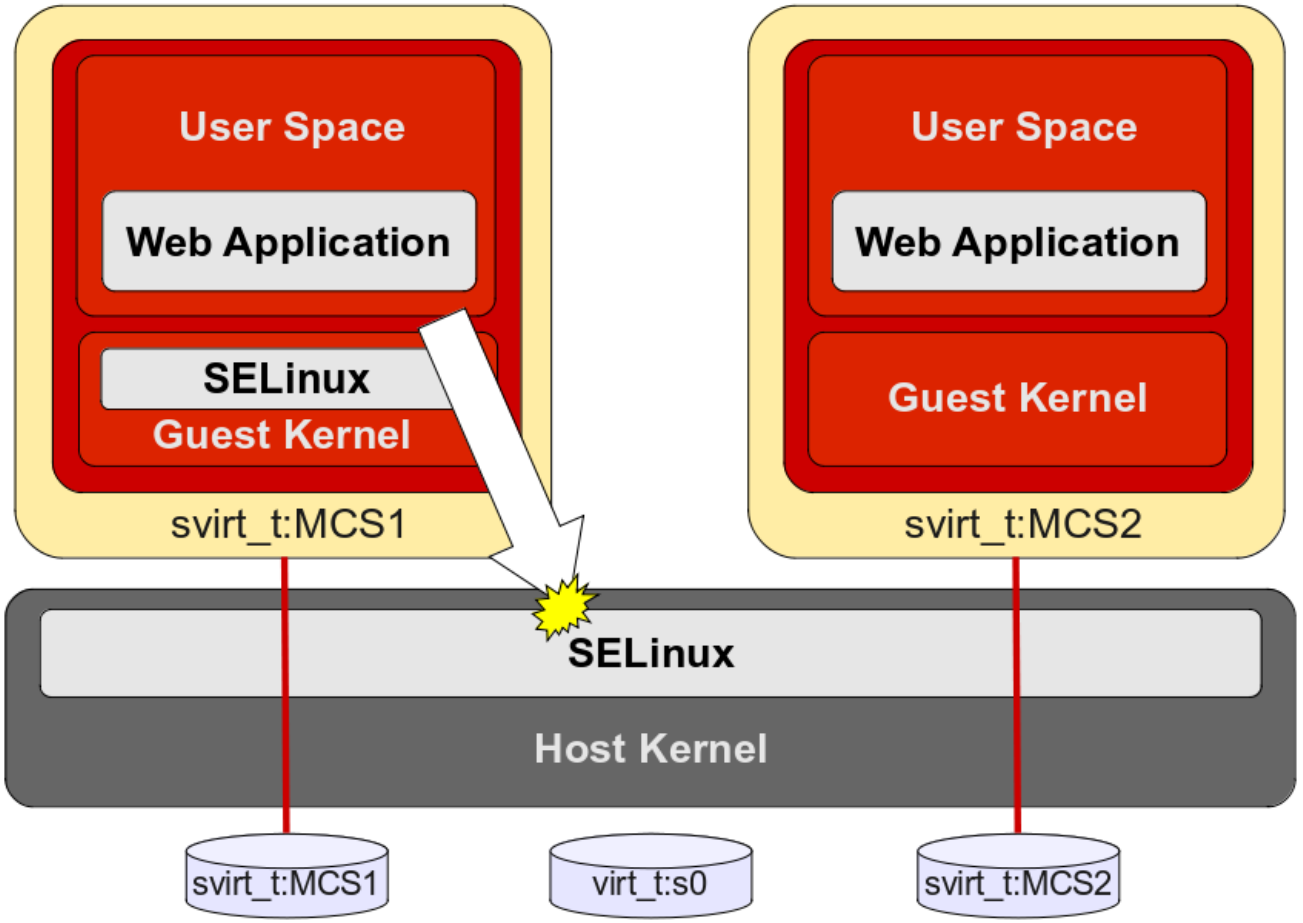
Potrzeby – host OS

- Analiza RAM:
  - GRR
  - Volatillity:
    - Ukryte procesy / moduły / pliki / połączenia
    - Połączenia sieciowe:
      - linux_netstat
      - linux_list_raw
      - linux_arp
      - connections
      - yarascan
      - hosts

http://downloads.volatilityfoundation.org/releases/2.4/CheatSheet_v2.4.pdf

Potrzeby – wirtualizacja

- sVirt → Secure Virtualization:



Potrzeby – konteneryzacja

- NO root!
- Capabalities:
  - Funkcjonalność ograniczająca dostęp do syscalli per proces
  - Redukująca np. pełne SUID/SGID:
    - Nasłuchiwanie na porcie <1024 (CAP_NET_BIND_SERVICE)
    - Ładowanie modułów (CAP_SYS_MODULE)
    - Tworzenie urządzeń (CAP_MKNOD)
    - mount/umount
    - Reboot (CAP_SYS_BOOT)
    - Operacje sieciowe (CAP_NET_RAW)
  - include/linux/capability.h

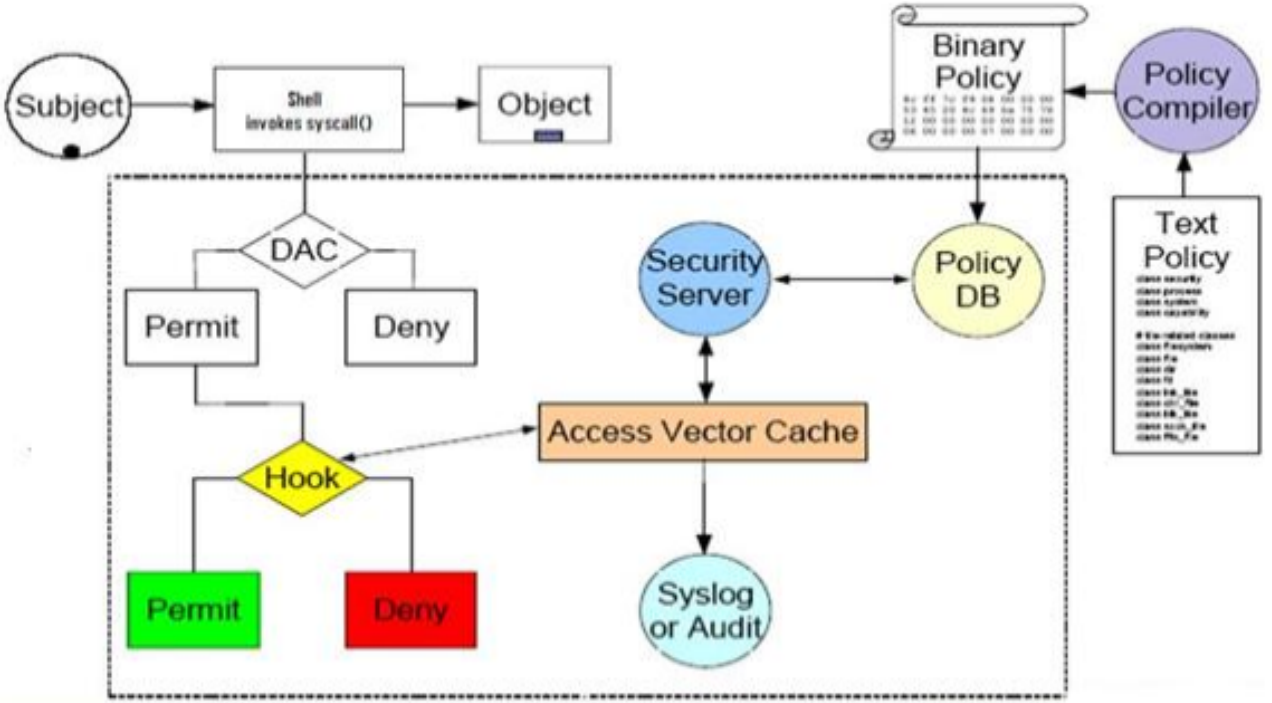
```
# docker run --cap-drop ALL --cap-add SYS_TIME ntpd /bin/sh
# pscap | grep 2382 5417 2382 root sh sys_time
```

Potrzeby – konteneryzacja

- Kernel namespaces:
  - Partycjonowanie globalnej tablicy identyfikatorów i struktur
  - Dedykowane „widoki”:
    - PID (CLONE_NEWPID)
    - MNT (CLONE_NEWNS)
    - IPC (CLONE_NEWIPC)
    - UTS (CLONE_NEWUTS)
    - NET (CLONE_NEWNET)
    - USER (--userns-remap)
  - Brak NS dla: dev, time, syslog

Potrzeby – konteneryzacja

- SELinux / Apparmor:
  - Ograniczenie wywołań systemowych
  - „sandboxy” per proces / kontener



Potrzeby – konteneryzacja

- Seccomp-bpf:
  - Filtrowanie syscalli:
    - Kernel 4.X ma ich ~400
    - V1.10:
      - 310 syscalli na whiteliście
      - 100 syscalli blokowanych by default
  - Wykorzystywane w:
    - Chrome, vsftpd, OpenSSH(UsePrivilegeSeparation)

prctl(PR_SET_SECCOMP , SECCOMP_MODE_FILTER , prog);

docker run –d –security-opt seccomp:allow:clock_adjtime ntpd

Potrzeby – konteneryzacja

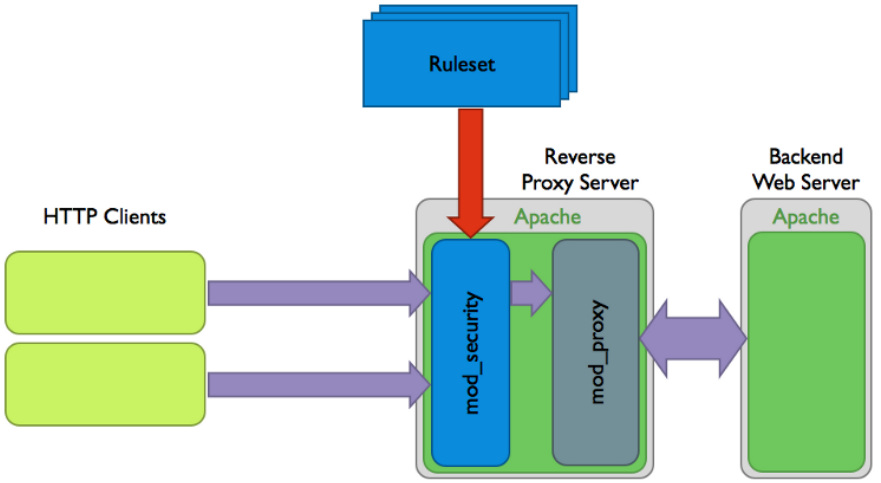
- Control groups – hierarchiczny kontroler dostępu do zasobów:
  - lub inaczej ulimits/rlimits on steroids
  - cgroup VFS
    - CPU
    - BLKIO
    - Memory
    - Network (tagowanie pakietów)
    - Freezer (SIGSTOP / SIGCONT)
    - PID (max. ilość procesów)
  - Wsparcie systemd

Potrzeby – konteneryzacja

- Aktywne monitorowanie/skanowanie kontenerów:
  - Docker Bench for Security
  - Clair
  - Atomic Scan
- Współdzielona sieć / port binding / FW
- Rest API / docker / gid=docker
- Private registry
- TLS
- Aktualizacje

Potrzeby – aplikacje webowe

- Web Application Firewall:
  - Walidacja metod HTTP
  - Wirtualne patchowanie
  - Ochrona przed:
    - SQLi, XSS, LFI/RFI, CSRF, CE, brute-force, directory traversal, itp. itd..
  - Web honeypots



Potrzeby – ruch sieciowy

- Wykrywanie wczesnej fazy enumeracji infrastruktury / systemu
- Aktywne / pasywne rozpoznawanie podatnych systemów / usług / aplikacji
- Wykorzystywanie pułapek
- Tworzenie baseline'ów:
  - Profilowanie ruchu sieciowego
  - Profilowanie zachowania systemów i urządzeń

Potrzeby – ruch sieciowy

- Security Onion → dedykowane Linux distro
  - Snort/Suricara (alerty)
  - Bro (sesje, transakcje, logowanie: http, dns, ftp, smtp, etc)
  - Netsniff-ng (full packet capture)
  - Sguil, Squert, ELSA, Snorby, Networkminer, Xplico (analiza)

  - Tryb pracy: sensor, serwer, standalone
  - Bardzo łatwy w instalacji (klik, klik)

VPS / cloud

- Brak dostępności urządzeń typu TAP / SPAN port
- Security Onion Cloud Client:
  - Daemonlogger lub netsniff-ng → eth0 → tap0
  - OpenVPN + bridge

Testowanie

- Malware pcaps:
  - Google query: „Mila malware pcap”
- Replay'owanie ruchu:
  - tcpreplay, tcpreplay-edit, tcpprep, tcprewrite
  - nfreplay
- Kali Linux
- pytbull, testmyids.com, testShellcode.py
- Python scapy

Podsumowanie

- Wielowarstwowa izolacja kluczem:>
- Minimum dla bezpiecznego „HW” produktu komercyjnego
- Kompletne i utwardzone rozwiązania na bazie Open Source
- Nauczenie się zachowania sieci i systemów podstawą profilowania
- Otwarte standardy gwarancją rozwoju

DEFENSIVE SECURITY

IT SECURITY EDUCATION & SERVICES

Oferta Defensive Security

- Edukacja w postaci ochrona vs atak:
  - *„Open Source Defensive Security” -*
    http://defensive-security.com/
  - Analiza poziomu bezpieczeństwa aplikacji i usług - *testy penetracyjne i audyty bezpieczeństwa*
  - Konsultacje i wdrożenia korporacyjnych rozwiązań Open Source z zakresu bezpieczeństwa i infrastruktury IT

DEFENSIVE SECURITY
IT SECURITY EDUCATION & SERVICES

Dziękuję za uwagę, zapraszam do kontaktu.

http://defensive-security.com

Leszek Miś
leszek.mis@defensive-security.com