2. Środowiska wirtualne

Środowisko wirtualne (ang. **Virtual Environments**, VE) to oprogramowanie symulujące działanie sprzętu komputerowego. W odróżnieniu od tradycyjnych systemów fizycznych umożliwia ono jednoczesne funkcjonowanie wielu niezależnych systemów operacyjnych lub aplikacji na jednej maszynie [1]. Działają one niezależnie od fizycznej struktury sprzętu. Tworzenie środowisk wirtualnych możliwe jest dzięki wykorzystaniu technologii wirtualizacji. Jest to technologia, wykorzystująca środowisko logiczne do przekroczenia fizycznych ograniczeń sprzętowych [2].

Podstawowym elementem środowisk wirtualnych jest maszyna wirtualna (ang. **Virtual Machine**, VM). Jest to aplikacja, wykonujące program tak, jakby była fizycznym urządzeniem, więc można byłoby powiedzieć, że jest to „komputer”, działający wewnątrz fizycznego komputera. Aplikacja VM (nazywana „gościem”) uruchamia swój własny system operacyjny na rzeczywistej maszynie (zwanej „gospodarzem”). Wirtualny system operacyjny może być dowolny, np. Windows lub MacOS, i nie jest ograniczony do jednego systemu operacyjnego na maszynie gospodarza [3]. Każda maszyna wirtualna działa niezależnie i nie ma wpływu na działanie innych VM-ów.

2.1. Charakterystyka wirtualizacji

Wirtualizacja to technologia umożliwiająca tworzenie wielu odizolowanych środowisk komputerowych – zwanych maszynami wirtualnymi (VM) – na jednym fizycznym urządzeniu. Dzięki warstwie pośredniczącej, zwanej hipernadzorcą (**hypervisor**), każda maszyna wirtualna może działać jak odrębny komputer z własnym systemem operacyjnym i aplikacjami, niezależnie od innych instancji. To podejście pozwala na efektywne wykorzystanie zasobów sprzętowych, zwiększenie skalowalności, uproszczenie zarządzania oraz ograniczenie kosztów operacyjnych [4].

Hipernadzorca (hypervisor) to oprogramowanie, które umożliwia tworzenie i zarządzanie maszynami wirtualnymi poprzez oddzielenie systemów operacyjnych gości od fizycznej infrastruktury sprzętowej. W zależności od sposobu działania, wyróżniamy dwa główne typy:

* typ 1: natywny (bare-metal) – działa bezpośrednio na sprzęcie, bez potrzeby instalowania systemu operacyjnego gospodarza. Przykładowo: Microsoft Hyper-V,
* typ 2: hostowany – funkcjonuje jako aplikacja zainstalowana w ramach istniejącego systemu operacyjnego. Przykładowo: VirtualBox, VMware Workstation.

Dodatkowo, hypervisory można klasyfikować ze względu na sposób wirtualizacji:

* Pełna wirtualizacja – system gościa działa bez konieczności modyfikacji, nieświadomy, że funkcjonuje w środowisku wirtualnym, jest w pełni niezależny.
* Wirtualizacja wspierana sprzętowo – wykorzystuje specjalne funkcje procesora, takie jak Intel VT-x czy AMD-V, w celu optymalizacji pracy maszyn wirtualnych.
* Parawirtualizacja – wymaga modyfikacji systemu gościa, który jest świadomy, że działa w środowisku wirtualnym i potrafi efektywnie współpracować z hipernadzorcą [5].

2.2. VMware jako platforma testowa

Vmware Workstation Pro (od firmy Broadcom) jest w pełni zwirtualizowanym środowiskiem sprzętowym dla systemu operacyjnego gościa. Program ten obsługuje wiele systemów operacyjnych gospodarza, w tym Windows, Linux oraz macOS, i został zaprojektowany z myślą o maksymalnym wykorzystaniu fizycznych zasobów komputera, co przekłada się na wysoką wydajność działania maszyn wirtualnych [4].

Oprogramowanie to stara się wykonywać instrukcje bezpośrednio na procesorze gospodarza, o ile jest to możliwe. W przypadkach, gdy bezpośrednie wykonanie kodu nie jest wspierane przez sprzęt, oprogramowanie korzysta z techniki dynamicznego tłumaczenia binarnego, która pozwala na przekształcanie instrukcji w czasie rzeczywistym. Tak przetworzony kod jest przechowywany w pamięci RAM, co umożliwia dalsze działanie maszyny wirtualnej z dużą szybkością – według producenta, z wydajnością sięgającą ponad 80% względem natywnego systemu [4].

VMware emuluje podstawowe elementy sprzętowe, takie jak karta graficzna (czego nie oferuje np., VirtualBox od firmy Oracle), karta sieciowa czy kontrolery dysków, a także umożliwia dostęp do urządzeń USB, portów szeregowych i równoległych za pomocą sterowników pośredniczących. Warto jednak zauważyć, że przenoszenie maszyn wirtualnych między różnymi komputerami, szczególnie z odmiennymi architekturami procesora lub liczbą rdzeni, może wymagać dodatkowej konfiguracji ze względu na różnice w zestawach instrukcji [4].

Dzięki powyższym rozwiązaniom, VMware Workstation Pro wyróżnia się wysoką wydajnością i wszechstronnością, co czyni go solidnym narzędziem do prowadzenia badań nad bezpieczeństwem w środowiskach wirtualnych.

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 1 Widok podstawowy Vmware Workstation w wersji 17.6.3

Podstawowymi funkcjami dostarczanymi przez środowisko Vmware Workstation są m.in.:

- tworzenie maszyn wirtualnych z wybraną konfiguracją systemu operacyjnego, pamięci RAM, przestrzeni dyskowej,

- snapshoty, czyli zapisywanie stanu maszyny wirtualnej w wybranym przez użytkownika momencie, co daje możliwość przywrócenia zapisanego obrazu,

- klonowanie maszyn wirtualnych,

- uruchamianie maszyn wirtualnych jednocześnie przy jednoczesnym ich izolowaniu.

- zaawansowana konfiguracja sieciowa, pozwalająca na definiowanie topologii sieci wirtualnych (m.in. NAT, bridge, host-only)

- połączenia ze zdalnymi serwerami, np. z Vmware ESXi.

2.3 Konfiguracja środowiska testowego

Do przeprowadzenia testów wykorzystany został komputer stacjonarny. Wyposażony został w następujące podzespoły:

- procesor AMD Ryzen 5 3600,

- 64GB pamięci RAM typu DDR4, 3600MHz,

- dysk M.2 o pojemności 2TB,

- karta graficzna NVIDIA GeForce RTX 3060 Ti.

Zainstalowany system operacyjny gospodarza to Windows 11 Pro w wersji 24H2.

Wykorzystana wersja Vmware to Workstation Pro 17.6.3 (rys.1).

2.3.1 Tworzenie maszyny wirtualnej systemu Windows

Do postawienia nowej maszyny wirtualnej z systemem Windows 10 wykorzystano zaawansowane ustawienia, które pozwalają na określenie opcji takich jak chociażby:

- typ kontrolera SCSI,

- typ dysku wirtualnego,

- kompatybilność ze starszymi produktami VMware (fot. 2).

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 2. Wybór typu konfiguracji przy tworzeniu nowej maszyny wirtualnej w VMware Workstation Pro 17.

Została wybrana kompatybilność sprzętowa dla wersji Workstation 17.5 lub nowszych (fot.3).

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 3. Wybór kompatybilności sprzętowej dla nowej maszyny wirtualnej.

Instalacja odbyła się przy użyciu pliku iso z systemem Windows 10 22H2 w architekturze 64-bitowej (fot. 4).

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 4. Wybór pliku iso z systemem operacyjnym

Dla porządku, maszynie wirtualnej została nadana nazwa „Windows 10 x64 Security Tests” oraz przypisana została odpowiednia lokalizacja na dysku.

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 5. Nazwa maszyny wirtualnej oraz nadanie ścieżki.

Istotną kwestią jest wybranie odpowiedniego typu instalowanego oprogramowania.

A screenshot of a computer

AI-generated content may be incorrect.

Fot.6. Wybór typu oprogramowania (UEFI lub BIOS)

Dla stawianego na maszynie wirtualnej systemu Windows 10 został wybrany typ oprogramowania UEFI (Unified Extensible Firmware Interface) , ze względu na lepsze dostosowanie do obecnych standardów bezpieczeństwa (niektóre narzędzia i techniki zabezpieczeń mogą działać tylko z UEFI, np. Microsoft Defender Credential Guard). BIOS jest technologią przestarzałą, a system Windows 10 jest domyślnie instalowany w technologii UEFI. Za użyciem UEFI przemawia także to, że:

- posiada modularną architekturę (co może być przydatne w kontekście testów bezpieczeństwa przy użyciu VMware, np. do wgrania sterowników DXE[[1]](#footnote-1))

- ma szybszy czas rozruchu niż BIOS,

- posiada abstrakcję sprzętową,

- oferuje zaawansowane opcje, obejmujące np. obsługę większych dysków twardych,

- dostarcza mechanizmy bezpieczeństwa. [7].

Opcja Secure Boot nie została zaznaczona, ponieważ w pierwszej fazie testów system będzie sprawdzany przy możliwie najmniejszych zabezpieczeniach (opcja ta chroni np. przed bootkitami czy rootkitami, czyli złośliwym kodem wykonywanym już przy starcie systemu).

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 7. Konfiguracja liczby procesorów i rdzeni.

Procesor zainstalowany w jednostce centralnej używanej do testów to AMD Ryzen 5 3600, co za tym idzie, posiada on 6 rdzeni fizycznych i obsługuje 12 wątków. Optymalnym wyborem będzie przypisanie jednego procesora logicznego oraz czterech rdzeni, co daje w sumie cztery wirtualne rdzenie. Przy takiej konfiguracji zapewniona zostanie wystarczająca wydajność, a także zostaną zachowane potrzebne zasoby dla hosta i innych zadań, wykonywanych równolegle przez komputer. Przydzielenie zbyt dużej liczby rdzeni maszynie wirtualnej mogłoby spowodować utratę stabilności systemu.

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 8. Konfiguracja przydzielonej ilości pamięci operacyjnej.

Wykorzystywany do testów komputer dysponuje 64GB pamięci RAM, zatem dla maszyny wirtualnej optymalną wartością będzie przypisanie 8GB pamięci (dokładnie 8192MB). Pozwoli to na płynne działanie pod potencjalnym większym obciążenie, gdzie rekomendowane 2GB mogłyby okazać się niewystarczające.

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 8. Konfiguracja sieci dla maszyny wirtualnej.

Dla przeprowadzanych w pracy testów bezpieczeństwa najlepszym typem połączenia z siecią jest połączenie mostkowane, ponieważ pozwoli to na przeprowadzenie ataków takich jak np. Man-in-the-Middle, sniffing czy ARP spoofing. Maszyna wirtualna w tym wypadku jest dostępna z poziomu sieci lokalnej. Umożliwia to sprawdzenie jak maszyna wirtualna wchodzi w interakcje z innymi urządzeniami. Wybór NAT mógłby być problematyczny, ze względu brak bezpośredniego dostępu z sieci, przez co maszyna wirtualna nie byłaby dostępna az poziomu innych urządzeń w sieci.

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 9. Konfiguracja dla urządzeń wejścia/wyjścia.

Wybrano rekomendowany typ LSI Logic SAS, który jest domyślnym kontrolerem I/O zapewnia wysoką kompatybilność z systemami operacyjnymi. Wadą tego rozwiązania względem Paravirtualized SCSI jest mniejsza wydajność, jednak przy wyborze wymienionej opcji należałoby ręcznie instalować wymagane sterowniki.

A screenshot of a computer

AI-generated content may be incorrect.

Fot. 10. Konfiguracja typu dysku.

Dla zapewnienia odpowiedniej wydajności został wybrany rekomendowany typ dysku NVMe, który wykorzystuje wydajne (w porównaniu do SCSI czy SATA) interfejsy SSD.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Bibliografia

[1] Virtualization Throughout the Software Lifecycle, Sarah N. Crutchfield

[2] Virtualization and Security Aspects: An Overview, Rui Filipe Pereira, Rui Miguel Silva & João Pedro Orvalho

[3] Virtualization and Forensics A Digital Forensic Investigator’s Guide to Virtual Environments

[4] VMware vSphere Essentials: A Practical Approach to vSphere Deployment and Management Luciano Patrão

[5] Optimal guest file system for type-2 hypervisorbased virtualization in Virtual box

[6] <https://uefi.org/specs/PI/1.8/V2_Overview.html>

[7] UEFI Memory Forensics: A Framework for UEFI Threat Analysis Kalanit Suzan Segal∗

1. Są to sterowniki, odpowiadające za inicjalizację chipsetu, procesora czy innych komponentów, a także za dostarczenie warstwy abstrakcji dla np. usług systemowych [6] [↑](#footnote-ref-1)