Testy Penetracyjne

Atakowanie specjalizowanych systemów

Arkadiusz Chrobot

Katedra Systemów Informatycznych

22 maja 2024

Plan

1. Wstęp
2. Maszyny wirtualne i kontenery
3. Systemy chmurowe
4. Urządzenia mobilne
5. Systemy automatyki przemysłowej

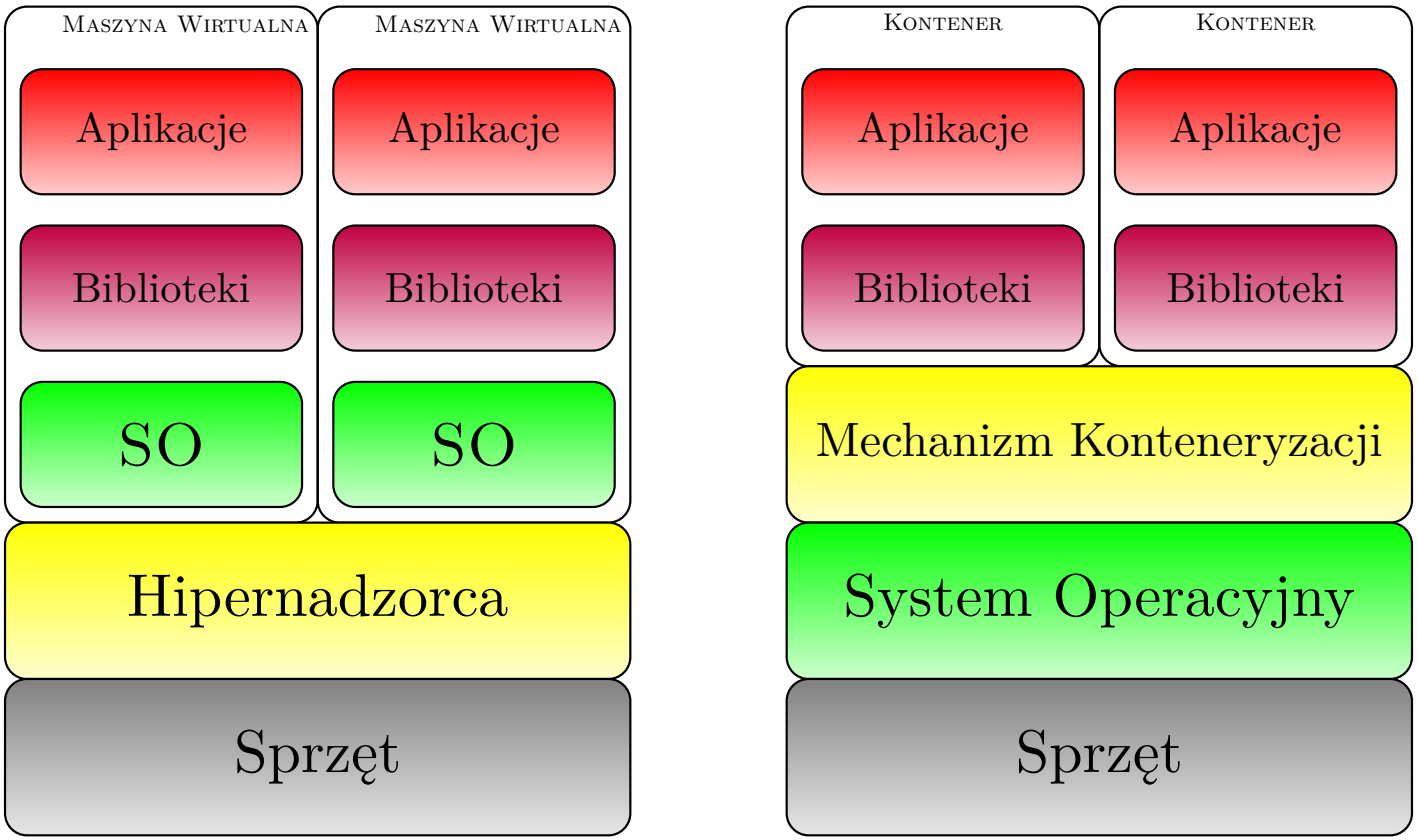
Wstęp

Testowanie penetracyjne systemów informatycznych obejmuje nie tylko infrastrukturę sieciową, systemy operacyjne i aplikacje (oraz czasami zabezpieczenia fizyczne), ale również może dotyczyć bardziej specjalizowanych technologi. Współcześnie w powszechnym użyciu są maszyny wirtualne i kontenery, które również mogą być celem takich testów, podobnie jak urządzenia mobilne oraz coraz częściej podłączane do Internetu systemy automatyki przemysłowej.

Maszyny wirtualne i kontenery

Rysunek 1 ilustruje różnicę między koncepcjami maszyny wirtualnej i kontenera. Maszyna wirtualna jest kopią maszyny fizycznej, z własnym systemem operacyjnym, pod kontrolą którego mogą pracować aplikacje użytkowe. Praca maszyn jest kontrolowana przez hipernadzorcę (ang. hypervisor). W przypadku kontenerów wirtualizacja zachodzi na poziomie systemu operacyjnego. Kontener jest zatem wyizolowanym środowiskiem — przestrzenią użytkownika — w którym mogą pracować aplikacje. Należy również dodać, że lewa strona rysunku przedstawia tylko jedną z możliwości realizacji maszyn wirtualnych, czyli z użyciem hipernadzorcy typu pierwszego. Istnieje również wirtualizacja z użyciem hipernadzorcy typu drugiego, gdzie maszyna fizyczna ma własny system operacyjny, a hipernadzorca pracuje w trybie użytkownika i parawirutalizacja, wymagająca specjalnej wersji systemu operacyjnego dla maszyny wirtualnej.

Maszyny wirtualne i kontenery



Rysunek: Porównanie maszyn wirtualnych i kontenerów na podstawie

▸ „Containers Vs. Virtual Machines”

Maszyny wirtualne i kontenery

Maszyny wirtualne i kontenery mogą być celem ataków w takim samym stopniu jak maszyny fizyczne i przy użyciu tych samych technik. Jednakże cenniejszą zdobyczą dla intruza jest system nimi zarządzający, który może kontrolować wiele takich zasobów. Przejęcie takiego systemu może wymagać bezpośredniego przełamania jego zabezpieczeń lub pokonania mechanizmów odizolowujących maszynę wirtualną bądź kontener od maszyny fizycznej. Ta ostatnia czynność nazywa się, w żargonie, *opuszczeniem piaskownicy* (ang. *sandbox escape*). Pierwszym krokiem do zastosowania tej techniki jest rozpoznanie, czy przejęta maszyna jest wirtualną, czy fizyczną. Jeśli pracuje ona pod kontrolą systemu Windows, to przydatne może być polecenie:

```
wmic baseboard get manufacturer,product
```

Pozwala ono sprawdzić konfigurację sprzętową i tym samym rozpoznać, czy któreś z urządzeń peryferyjnych nie jest wirtualne. W przypadku systemów Linux z zainstalowanym oprogramowaniem `systemd` można użyć polecenia `systemd-detect-virt`.

Maszyny wirtualne i kontenery

Jeśli w wersji Linuksa zainstalowanej na przejętym komputerze nie ma `systemd`, to można wydać polecenie:

```
ls -l /dev/disk/by-id
```

Jego wynik ujawni producenta pamięci masowej, która może być wirtualnym urządzeniem. Inne polecenia sprawdzające konfigurację komputera, takie jak `lspci`, `lsusb`, `lscpu`, `lsmod` również mogą być pomocne. Jeśli przejęte konto należy do użytkownika z uprawnieniami administratorskimi, to można użyć także narzędzia `hdparm`:

```
sudo hdparm -i /dev/sda
```

Opcja `-i` tego narzędzia pozwala pozyskać informację o producencie i parametrach urządzenia pamięci masowej. Nazwa pliku tego urządzenia może być inna, niż ta podana w przykładzie. Wymienione polecenia nie tylko pozwalają ustalić, czy przejęta maszyna jest maszyną wirtualną, ale także odkryć nazwę ewentualnego hipernadzorcy (np. VirtualBox lub VMWare).

Maszyny wirtualne i kontenery

To z kolei umożliwia intruzowi sporządzić listę potencjalnych luk bezpieczeństwa, jakie może posiadać hipernadzorca, szczególnie tych, które pozwalają na zdalne wykonanie kodu. Nawet jeśli istnieją dla nich poprawki, to mogą nie być zainstalowane, z uwagi na to, że hipernadzorcy są zainstalowani w środowiskach produkcyjnych i przerwanie ich działania może być niezgodne z SLA.
Jeżeli jednak okaże się, że poprawki naprawiające podatności zostały zainstalowane, to atakujący może użyć innych metod, takich jak socjotechnika lub wykorzystanie niepoprawnej konfiguracji, zwłaszcza jeśli umożliwia ona zdalne zarządzanie hipernadzorcą.
Znalezienie działającego eksploita, który umożliwiałby opuszczenie maszyny wirtualnej (ang. *virtual machine escape*) jest trudne. Owszem, znaleziono wiele tego typu podatności w prawie każdym hipernadzorcy, ale zostały one szybko naprawione przez producentów i znalezienie ich w systemach produkcyjnych jest wysoce nieprawdopodobne.

Maszyny wirtualne i kontenery

Warto zwrócić uwagę, że istnieją repozytoria, takie jak AWS *Marketplace*, *VMWare Marketplace*, *Azure Marketplace*, *osboxes*, które dostarczają gotowych do pobrania i użycia maszyn wirtualnych (zarówno odpłatnie, jak i bezpłatnie). Jeśli intruzowi uda się umieścić w takim repozytorium odpowiednio spreparowaną maszynę wirtualną, to może on uzyskać w ten sposób dostęp do większości systemów, gdzie ta maszyna zostanie użyta. Być może będzie to nawet dostęp trwały, tzn. możliwy przez dłuższy czas. Z drugiej strony należy odnotować, że istnieją repozytoria maszyn wirtualnych z celowo pozostawionymi podatnościami. Służą one do ćwiczenia technik testów penetracyjnych. Przykładem takiego repozytorium jest *VulnHub*. OWASP udostępnia z kolei maszyny z podatnymi aplikacjami internetowymi w ramach projektu BWA (*Broken Web Applications*). Są one już trochę nieaktualne, lecz nadal spełniają swój cel.

Maszyny wirtualne i kontenery

Atakowanie kontrolerów kontenerów odbywa się podobnie jak w przypadku atakowania hipernadzorców — od przejęcia kontenera. Do tego celu można użyć tych samych technik, które umożliwiają uzyskanie dostępu do maszyn wirtualnych oraz fizycznych. Po uzyskaniu dostępu do kontenera wystarczy czasem uruchomienie klienta mechanizmu konteneryzacji, aby przejąć nad nim kontrolę. Konteneryzacja jest często stosowana w systemach chmurowych. Niektóre z nich polegają na powszechnie instalowanych środowiskach, takich jak Docker i Kubernetes. Inne, jak np. Amazon, mają własne rozwiązania, z kategorii systemów bezserwerowych (ang. *serverless*). Prowadzenie testów penetracyjnych tej ostatniej kategorii może być zatem utrudnione. Z kolei dla takich rozwiązań jak Docker dostępne są ▸ maszyny wirtualne, wraz z ▸ dokumentacją, służące do ćwiczenia testów penetracyjnych na takich środowiskach. Typowe problemy z konfiguracją mechanizmów konteneryzacji są opisane ▸ w serii trzech artykułów dostępnych w Internecie.

Systemy chmurowe

Istnieje kilka typów usług, udostępnianych przez systemy chmurowe, takich jak *infrastruktura jako usługa* (ang. *Infrastructure as a Service — IaaS*), *oprogramowanie jako usługa* (ang. *Software as a Service — SaaS*) lub *platforma jako usługa* (ang. *Platform as a Service — PaaS*). Są one coraz częściej wykorzystywane w systemach informatycznych. Stanowią one również pewną przeszkodę w prowadzeniu całościowych testów penetracyjnych, bo są to systemy należące do zewnętrznych podmiotów, które najczęściej nie życzą sobie, aby były poddawane takiej weryfikacji. Nawet jeśli właściciel systemu chmurowego zgodzi się na objęcie go testami penetracyjnymi, to sprawdzenie go może wymagać żmudnych przygotowań, aby nie zakłócić innych klientów korzystających z tego systemu.

Systemy chmurowe

Do atakowania systemów chmurowych można zastosować typowe techniki, jak i również specjalizowane. Przykładowo techniki istnieją specjalne techniki ▸ eskalacji uprawnień dla AWS. Inną metodą możliwą do zastosowania wyłącznie w AWS jest atak na usługi ▸ metadanych, umożliwiające otrzymanie czasowych uprawnień do innych usług.
Udany atak może też umożliwić nieprawidłowa konfiguracja mechanizmów IAM (*Identity and Access Management*) lub *magazynów obiektów* (ang. *object storage*), takich jak `S3`, udostępnianych przez firmę Amazon. W wyniku niewłaściwego skonfigurowania mogą się one stać publicznie dostępne, mieć otwarty dostęp do przesyłania (ang. *upload*) lub pobierania (ang. *download*) plików, lub wyświetlania zawartości katalogów.
W przypadku systemu *Azure* można napotkać na nieprawidłowe konfiguracje *federacji*, umożliwiających określonym klientom korzystanie z usług dostępnych poza systemem chmurowym.

Systemy chmurowe

Do innych ataków, właściwych dla systemów chmurowych zalicza się:

Wstrzyknięcie malware (ang. *cloud malware injection attacks*) — ich celem jest przekierowanie użytkowników do usług kontrolowanych przez intruza. Wstrzyknięcie może się odbywać tradycyjnymi metodami (np. XSS), lub atakującymi charakterystyczne elementy systemu, np. usługi lub narzędzia.

Wyczerpanie zasobów (ang. *resource exhaustion*) — są podobne do ataków typu *odmowa usługi*. Systemy chmurowe powinny być na nie odporne i najczęściej są pod tym kątem sprawdzane poza testami penetracyjnymi, ale nie można wykluczyć, że pojawi się na nie podatność w wyniku nieprawidłowej konfiguracji lub niewystarczających zasobów sprzętowych.

Systemy chmurowe

Prosto do źródła (ang. *direct-to-origin* D2O) taki, których celem jest pominięcie rozwiązań typu CDN (*Content Delivery Network*), mechanizmy równoważenia obciążenia i pośredników (ang. *proxy*), aby zaatakować mniej skalowalne lub chronione usługi.

Ataki z użyciem kanałów bocznych (ang. *side-channel attacks*) — przykładem takiego ataku może być sytuacja, w której uprawniony klient zmniejszył swój wolumen pamięci masowej, a odzyskane miejsce została natychmiast przydzielone intruzowi, który może z niego odczytać dane legalnego użytkownika. Systemy chmurowe zapobiegają takim zdarzeniom stosując szyfrowanie wolumenów lub inne mechanizmy, ale mogą także istnieć inne kanały boczne. W przypadku chmur mogą to być dowolne zasoby współdzielone.

Systemy chmurowe

Do testów penetracyjnych systemów chmurowych można wykorzystać takie narzędzia jak: *ScoutSuite* — darmowe narzędzie typu *open source* do przeprowadzania audytów różnych systemów chmurowych, *CloudBrute* — narzędzie do identyfikacji aplikacji i magazynów danych w wielu różnych systemach chmurowych, *Pacu* — narzędzie do eksploracji systemów chmurowych AWS, *Cloud Custionan* — narzędzie do zabezpieczania systemów chmurowych, ale może być wykorzystane także do testów bezpieczeństwa. Przydatne mogą także się okazać środowiska SDK, udostępniane np. przez większość dużych dostawców usług chmurowych.

Urządzenia mobilne

Urządzenia mobilne, takie jak smartfony i tablety, coraz częściej stają się elementami infrastruktury informatycznej przedsiębiorstw. Jednocześnie są one przedmiotami osobistego użytku pracowników, co może rodzić problemy w takcie prowadzenia testów penetracyjnych i musi zostać uwzględnione w ich planowaniu. W odniesieniu do tych urządzeń należy rozważyć trzy typy ataków:

Inżynieria wsteczna (ang. *reverse engineering*) — polega na statycznej lub dynamicznej analizie kodu aplikacji, celem poszukiwania haseł, kluczy API i innych cennych informacji.

Sandbox analysis polega na uruchomieniu kodu lub nawet całego obrazu urządzenia w kontrolowanym środowisku, celem zbadania jego zachowania.

Spamowanie (ang. *spamming*) może być wykorzystane np. do phishingu.

Urządzenia mobilne

Wśród typów podatności właściwych dla urządzeń mobilnych można wyróżnić:

- *Użycie niezabezpieczonego magazynu danych*, w postaci usługi chmurowej lub nawet niezaszyfrowanej karty microSD.
- *Podatny system uwierzytelniania* — podatności mogą obejmować obejście z użyciem e-maila resetującego, wstrzyknięcie kodu celem zmiany zachowania funkcji uwierzytelniających lub nawet sprawdzenie, gdzie użytkownik pozostawił odciski palców wprowadzając PIN. Mechanizmy biometryczne mogą być podatne na użycie zdjęcia twarzy lub palca, albo ich dane wyjściowe mogą być zmanipulowane.
- *Problemy z przypinaniem certyfikatu* (ang. *certificate pinning*) — ten mechanizm może być ominięty przy użyciu socjotechniki lub wstrzyknięcia kluczy, albo całych certyfikatów do urządzenia.

Urządzenia mobilne

- *Problemy z łańcuchem dostaw* — oprogramowanie urządzeń mobilnych jest budowane z wielu komponentów, dostarczanych przez różnych producentów i wzajemnie od siebie zależnych. To oznacza, że mogą zawierać one wiele podatności, także wynikających z ich współpracy.
- *Wykonanie aktywności jako administrator* (ang. *execution of activities using root*) — wykonanie kodu jako użytkownik `root` zazwyczaj daje praktycznie nieograniczoną kontrolę nad systemem. W przypadku urządzeń mobilnych utrudniają to rozwiązania typu *SEAndroid*/*SELinux*, ale nie czynią niemożliwym.
- *Podatności na poziomie logiki biznesowej* — np. brak autoryzacji przelewów w mobilnych aplikacjach bankowych.

Urządzenia mobilne

Do narzędzi używanych do przeprowadzania testów penetracyjnych urządzeń mobilnych można zaliczyć:

Burp Suite może posłużyć do testowania aplikacji mobilnych bazujących na technologiach webowych.

MobSF czyli *Mobile Security Framework* jest zautomatyzowanym narzędziem do testów penetracyjnych, oceny zabezpieczeń i analizy złośliwego oprogramowania, przeznaczonym dla systemów operacyjnych iOS, Android i Windows.

Postman to narzędzie do testowania API, które może być przydatne w testowaniu penetracyjnym aplikacji mobilnych w podobnym zakresie jak Burp Suite.

Ettercap narzędzie do wykonywania ataków typu *man*-in-the-middle.

Frida narzędzie do wstrzykiwania kodu JavaScript lub bibliotek do aplikacji przeznaczonych zarówno dla mobilnych, jaki i „stacjonarnych” systemów operacyjnych.

Urządzenia mobilne

Objection oprogramowanie bazujące na narzędziu Frida i przeznaczone do oceny zabezpieczeń aplikacji mobilnych.

Android Studio SDK dla programistów tworzących oprogramowanie dla systemu Android.

Dozer narzędzie o podobnym zastosowaniu jak Metasploit, ale przeznaczone dla systemu Android.

Systemy automatyki przemysłowej

Do systemów automatyki przemysłowej zalicza się rozwiązania typu SCADA (*Supervisory Control and Data Acquisition*), ICS (*Industrial Control System*), *IIoT* (*Industrial Internet of Things*), jak również zwykłe systemy IoT i systemy wbudowane. Z punktu widzenia najważniejszą ich cechą jest *dostępność* (ang. *availability*), co też trzeba uwzględnić planując ich testy penetracyjne. Słabymi punktami tych rozwiązań mogą być:

Protokół BLE będący energooszczędną wersją Bluetooth. Jest on podatny na ataki typu *man-in-the-middle*, podsłuchanie danych uwierzytelniających, fałszowanie adresu MAC, odmowa usługi i zakłócanie. Dodatkowo w urządzeniach korzystających z tego protokołu często nie jest wystarczająco dobrze zabezpieczony proces parowania.

Niebezpieczne wartości domyślne mogą nimi być dane uwierzytelniające lub konfiguracyjne. Niektóre z nich mogą być *zaszyte na stałe* (ang. *hard-coded*) w urządzeniu.

Systemy automatyki przemysłowej

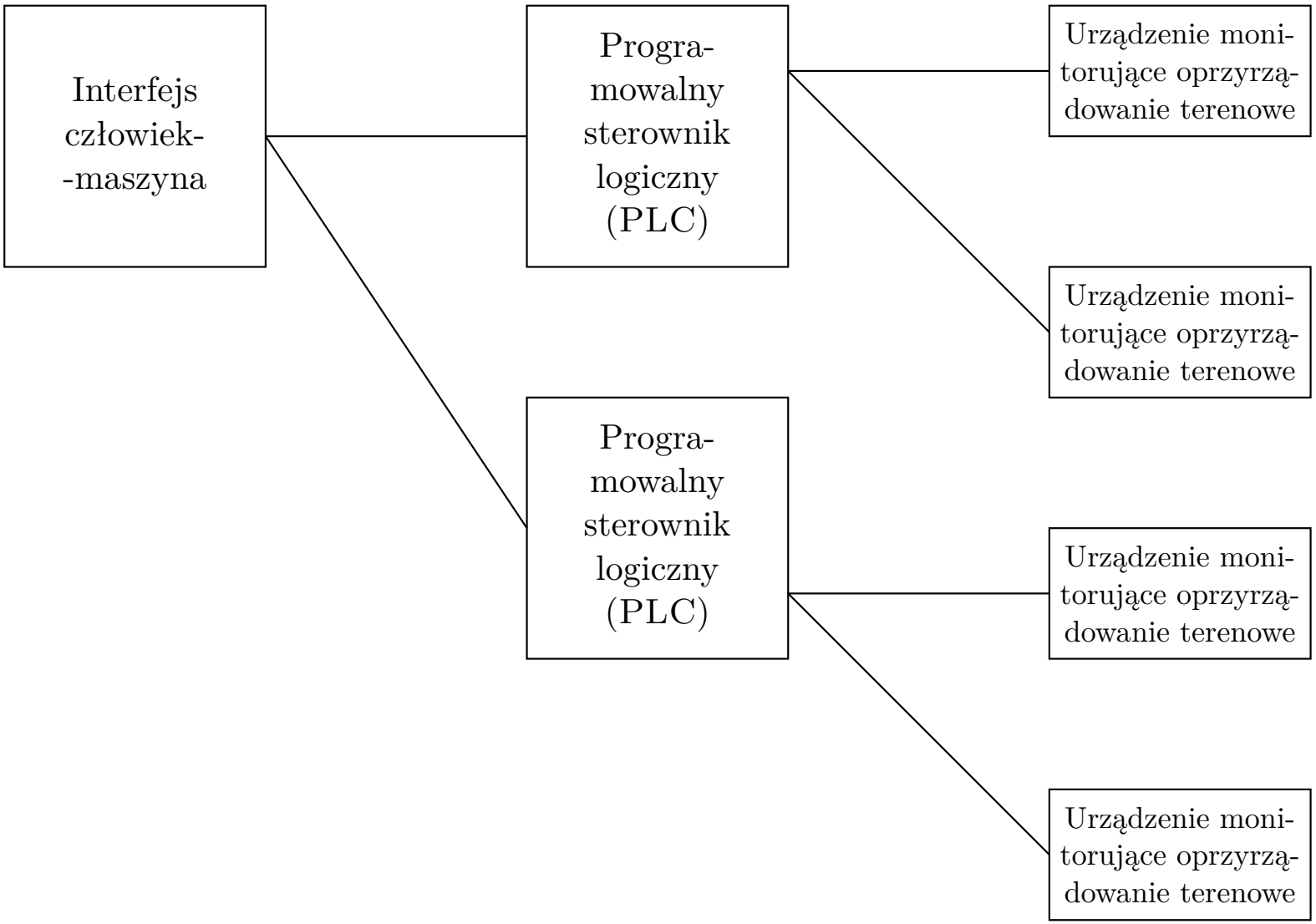
Niezabezpieczone lub nieaktualne komponenty wiele z urządzeń automatyki przemysłowych jest instalowanych w miejscach, gdzie trudno jest do nich dotrzeć, aby zaktualizować ich oprogramowania. Często ich producenci dostarczają poprawki w nierównych odstępach czasu lub w ogóle ich nie dostarczają.

Jawna komunikacja urządzenia automatyki przemysłowej mogą mnie mieć wystarczających zasobów, aby zapewnić szyfrowanie komunikacji, więc wszystkie informacje przesyłane przez nie i do nich są jawne.

Systemy automatyki przemysłowej

Urządzenia automatyki przemysłowej wykorzystują specyficzne dla nich protokoły komunikacyjne. Przykładowo, ▸ SCADA, którego schemat koncepcyjny przedstawia Rysunek 2, używa takich protokołów, jak ModBus, DNP3, BACnet. Ich znajomość może okazać się przydatna do przeprowadzenia testów penetracyjnych. W typowych systemach informatycznych są częściej spotykane rozwiązania typu *Intelligent Platform Management Interface* (IPMI), takie jak DRAC firmy Dell lub iLO firmy HP. To systemy wbudowane umieszczane np. w szafach rakowych, aby zarządzać serwerami klastrowymi w przypadku sytuacji awaryjnych. Są one atrakcyjne z punktu widzenia intruza, bo umożliwiają nawet zdalne przejęcie kontroli, na niskim poziomie, nad takimi komputerami.
Testy penetracyjne systemów wbudowanych mogą oznaczać np. testowanie bezpieczeństwa samochodów. Wykonano je np. dla produktów firmy ▸ Jeep oraz ▸ Tesla. W tym ostatnim przypadku udostępniony został także ▸ raport z tych testów. Ponieważ dotyczyły one autopilota, to warto zwrócić uwagę na ▸ problemy związane z ▸ rozwiązaniami bazującymi na sztucznej inteligencji.

Systemy automatyki przemysłowej



Rysunek: Schemat prostego system typu SCADA

Pytania

?

Koniec

Dziękuję Państwu za uwagę!