Testy Penetracyjne

Ekspolatacja i eksploracja

Arkadiusz Chrobot

Katedra Systemów Informatycznych

24 kwietnia 2024

Plan

Wstęp

Eksploatacja i eksploracja (ang. *exploiting and pivoting*) jest kolejną fazą w testowaniu penetracyjnym, po zbieraniu informacji i skanowaniu podatności. Głównym celem tego etapu jest wykorzystanie podatności znalezionych w określonym urządzeniu sieciowym w celu uzyskania do niego dostępu. Do kolejnych czynności zalicza się zwiększenie uprawnień (ang. *escalation of privileges*) — jeśli okaże się to konieczne, uzyskanie dodatkowych danych o atakowanym systemie lub infrastrukturze, zapewnienie trwałego (ang. *persistent*) dostępu do urządzenia, zatarcie/ukrycie śladów włamania i ekspansja (ang. *pivot*) na inne elementy systemu/infrastruktury.

Przygotowanie

Przygotowania do przeprowadzenia testu penetracyjnego obejmują wybranie elementu systemu, którego zabezpieczenia zostaną sforsowane jako pierwsze. Decyzję tę podejmuje się nie tylko na podstawie wyników skanowania podatności, ale także bazując na celu testów, umowie z klientem, zebranych informacjach (np. liście użytkowników, szczegółach aplikacji), a także umiejętnościach poszczególnych testerów. Najczęściej dobrym wyborem okazują się elementy systemu, w których skanery znalazły największą liczbę podatności, ze względu na różnorodny charakter luk. Jednakże niekoniecznie luki z najwyższą oceną CVSS są tymi, które umożliwią nieautoryzowany dostęp do systemu.

Przygotowanie

Jeśli zespół testujący dysponuje informacjami na temat kont założonych w atakowanym systemie, to może wybrać te, które są rzadko używane i których właściciele nie zauważą podejrzanej aktywności. Enumeracja (ang. *enumeration*) legalnych użytkowników systemu może być dokonana za pomocą siłowych (ang. *brute-force*) prób zalogowania do systemu (łatwe do wykrycia), użycia mechanizmu obsługi zapomnianych haseł (pozwala ustalić prawdziwość loginów), sprawdzenia zawartości pliku `/etc/passwd` (konieczny jest dostęp), zapytań Active Directory (jak wcześniej), listy katalogów domowych lub innych elementów systemu plików (także konieczny dostęp), odpytywania usług katalogowych lub poszukiwania informacji na pracowniczych portalach społecznościowych.

Przygotowanie

Oprócz informacji o użytkownikach użyteczne są także dane o ich grupach. Dzięki nim testerzy mogą dowiedzieć się którzy użytkownicy mają uprawnienia administratorskie lub dotyczące interesujących zasobów. W uniksowych systemach operacyjnych te informacje zawarte są w pliku `/etc/group`, a w systemach rodziny Windows można je uzyskać przy pomocy Active Directory, powłoki *PowerShell* lub za pomocą narzędzi zażądania lokalnymi użytkownikami i grupami.
Usługa Active Directory posiada również wysokopoziomowy kontener, nazwany *lasem* (ang. *forest*), który zawiera informacje o domenach, użytkownikach, komputerach i grupach. Jest on zatem źródłem cennych informacji z punktu widzenia zespołu testów penetracyjnych. W przypadku gdy domena ufa innej domenie możliwa jest enumeracja lasu należącego do tej innej domeny.

Przygotowanie

Umiejscowienie i dostępność w systemie wrażliwych danych (ang. *sensitive data*) zależne są od wielu czynników, takich jak polityka bezpieczeństwa podmiotu będącego właścicielem systemu, indywidualne nawyki użytkowników, szyfrowanie, zgodność ze standardami i wymogami prawnymi i inne. Niekiedy konieczne jest wykonanie całościowego lub częściowego przeszukania systemu plików i wyselekcjonowanie tych zbiorów danych, które mogą zawierać interesujące informacje. W takiej czynności mogą pomóc standardowe narzędzia systemu Windows lub Linux (np. `cat`, `grep`, `strings`, `file`, `find`), jak również oprogramowanie, które bada entropię danych, pozwalając odróżnić pliki zaszyfrowane od jawnych.
To, co będzie podlegało enumeracji, zależy od przyjętego scenariusza prowadzenia testów penetracyjnych oraz poziomu początkowego dostępu do atakowanego systemu.

Przygotowanie

Raporty skanowania podatności mogą zawierać nie tylko informacje o istotności (ang. *severity*) luki, ale również ocenę rozpoznania (ang. *quality of detection*). Wybierając wektor ataku należy uwzględnić obie te dane. Jeśli raport nie uwzględnia oceny rozpoznania, to można sprawdzić, czy znajdują się w nim pozycje czysto informacyjne. Ich treść może potwierdzić lub zaprzeczyć trafności detekcji podatności. Jeśli i taka informacja nie znajduje się w raporcie, to testerzy mogą uzyskać ją sprawdzając bezpośrednio odpowiedzi usług działających na potencjalnym celu. Źle skonfigurowane serwery HTTP (Apache, IIS, ale również inne) podają swoją wersję. Podobnie mogą robić systemy baz danych, a nawet całe aplikacje internetowe.

Przygotowanie

Znajomość wersji usług pozwala zespołom wykonującym testy penetracyjne dobrać oprogramowanie wykorzystujące podatności do uzyskania nieautoryzowanego dostępu, czyli narzędzia nazywane *eksploitami*. Poniżej zostały wymienione strony WWW zawierające gotowe eksploity dla konkretnych luk. Należy zachować pobierając tego typu oprogramowanie z innych źródeł. Najlepiej pobrać kod źródłowy eksploita i przed użyciem starannie go przeanalizować, gdyż może on nie tylko atakować wskazany system, ale także komputer atakującego, zatem jego działanie może naruszyć bezpieczeństwo działań (ang. *operational security*) zespołu przeprowadzającego testy penetracyjne. ▸ Exploit-DB ▸ VULDB ▸ Rapid7 ▸ NVD ▸ Packet Storm ▸ 0day ▸ cxsecurity ▸ Vulnerability Lab

Narzędzia

Przeprowadzenie testów penetracyjnych wymaga użycia odpowiedniego oprogramowania narzędziowego. Część z nich testerzy tworzą samodzielnie, przy użyciu języków skryptowych (interpretowanych) lub kompilowanych, jeśli wymaga tego konkretne zlecenie[1]. Częściej jednak posługują się gotowymi narzędziami. Niektóre z nich zostaną opisane w dalszej części wykładu, ze szczególnym uwzględnieniem programu *Metasploit*.

[1] Należałoby dodać „...i pozwala na to prawo". Przykładowo w Wielkiej Brytanii *Computer Misuse Act* zabrania wytwarzania programów, których celem jest przełamywanie zabezpieczeń.

Metasploit

Oryginalnym twórcą programu Metasploit jest H. D. Moore, który w 2009 roku odsprzedał je firmie Rapid7, tworzącej narzędzia dla cyberbezpieczeństwa, między innymi maszyny wirtualnych *Metasploitable 2* i *Metasploitable 3* używane do ćwiczenia testów penetracyjnych. Narzędzie Metasploit dostępne jest w wersji komercyjnej i darmowej, zupełnie wystarczającej także w profesjonalnych zastosowaniach. Podstawowa wersja tego oprogramowania posiada interfejs tekstowy, ale integruje w sobie wiele innych narzędzi używanych na różnych etapach testów penetracyjnych, między innym program Nmap. Metasploit ma budowę modularną. Najczęściej używanymi modułami są eskploity (ang. *exploits*), *ładunki* (ang. *payloads*) i moduły dodatkowe (ang. *auxiliary*). Do ostatnich zaliczane są skanery, sniffery oraz inne narzędzia. Z kolei ładunki to kod, którego zadaniem jest wykonanie nieuprawnionych operacji na urządzeniu, po uzyskaniu lub podczas uzyskiwania dostępu przez eksploit.

Metasploit

Ładunki (ang. *payloads*) są *modułami eksploitów*, które mogą stanowić razem z eksploitem zamkniętą całość (ang. *inline payloads*) lub składać się z dwóch części: *inscenizatora* (ang. *stager*) i właściwego kodu (ang. *stage*). Ładunków z pierwszej kategorii używa się np. wtedy, gdy dostępne jest jedynie łącze o niskiej przepustowości. Mogą wykonywać różne zadania, od dodania nowego użytkownika lub wykonania pojedynczego polecenia systemowego, do udostępnienia powłoki systemowej. Inscenizatory tworzą połączenie między maszyną atakującego i atakowanym urządzeniem. Zazwyczaj są one niewielkich rozmiarów i zaprojektowane tak, aby były niezawodne. Kiedy zostaną osadzone na docelowej maszynie, to pobierają resztę ładunku (właściwy kod). Ten może mieć formę wykonywalnych plików, lub kodu umieszczanego w pamięci operacyjnej, np. wstrzykiwanego do bibliotek DLL. Przykładem należącym do ostatniej kategorii jest *Meterpreter*, oryginalnie napisany dla systemu Windows, który jest powłoką poleceń, łączącą się z komputerem atakującego.

Metasploit

Metasploit posiada polecenie `search`, które pozwala wyszukać ładunek, eksploit lub dodatkowy moduł. W przypadku eksploitów wynik zawiera nie tylko nazwę oprogramowania, które opisuje jego przeznaczenie, np. `auxiliary/scanner/http/ssl_version`, czyli dodatkowy moduł będący skanerem serwera HTTP, który wykrywa używaną wersję SSL/TLS, ale również jego ocenę. Możliwe wartości dla tej oceny, wraz z ich opisem, podano na slajdzie 14.

Metasploit

Ocena	**Znaczenie**
Excellent	Ekspliot nigdy nie spowoduje awarii atakowanej usługi.
Great	Eksploit automatycznie wykrywa docelową usługę, albo sprawdza jej wersję i zwraca właściwy dla niej adres.
Good	Ekspliot ma domyślną usługę docelową i jest ona często spotykana.
Normal	Eksploit jest niezawodny, ale wymaga odpowiedniej wersji usługi docelowej, która nie jest domyślnie wykrywana.
Low	Eksploit jest trudny w użyciu, a szanse jego powodzenia są niższe niż 50%.
Manual	Eksploit jest niestabilny, trudy w użyciu lub wymaga specyficznej, ręcznej konfiguracji.

Metasploit

Przykład użycia

Działanie narzędzia *Metaspliot*, zainstalowanego w dystrybucji *Kali* systemu operacyjnego *Linux* zostanie zaprezentowane na przykładzie maszyny *Metasploitable 2*, która działa pod kontrolą wersji serwerowej dystrybucji *Ubuntu*. Ma ona zainstalowaną „bezpieczną" usługę FTP, które demon (`vsftpd`) zawiera tylną furtkę (ang. *backdoor*), pozwalającą nieuprawionemu użytkownikowi uzyskać dostęp do powłoki poleceń, bez podawania hasła. Pierwszym krokiem jest uruchomienie w dystrybucji Kali Metasploita poleceniem:

```
msfconsole
```

Po uruchomieniu narzędzia należy wyszukać odpowiedni eksploit, dla oprogramowania `vsftpd`:

```
msf6 > search vsftpd
```

Jednym z wyników będzie `/unix/ftp/vsftpd_234_backdoor`, który potrafi wykorzystać tę podatność. Aby go użyć należy zastosować polecenie `use`:

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
```

Metasploit

Przykład użycia

Esploity zwykle wymagają konfiguracji przed użyciem, aby dowiedzieć się jakie są dostępne parametry tego oprogramowania i które wymagają określenia wartości należy posłużyć się poleceniem `show options`:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show
options
```

Czytając wynik ostatniego polecenia dowiemy się, że należy nadać wartość zmiennej `RHOST`, która określa adres IP atakowanego urządzenia. W tym celu można posłużyć się poleceniem `set`, które określi wartość tej zmiennej tylko dla bieżąco używanego eksploita. Jednak ta zmienna jest używana w większości eksploitów, więc warto zdefiniować jej wartość globalnie, przy pomocy podobnego polecenia, czyli `setg`:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > setg
RHOSTS 192.168.220.131
```

Metasploit

Przykład użycia

Po wykonaniu konfiguracji można uruchomić eksploit poleceniem, które po prostu nazywa się `expliot`:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
```

Po jego wydaniu na ekranie pojawi się szereg komunikatów informujących o postępie pracy eskploita. Jeśli jego działanie zakończy się pomyślnie, to użytkownik otrzyma zdalny dostęp do powłoki poleceń atakowanego urządzenia.

Inne narzędzia

Przydatnym zestawem narzędzi w przypadku ataku na systemy rodziny Windows mogą okazać się skrypty dla powłoki *PowerShell*, o nazwie *PowerSploit*. Pomagają one ominąć programy antywirusowe, eksfiltrować dane, dokonać rekonesansu lub wykonać inżynierię wsteczną, albo uruchomić określony kod lub uzyskać trwały dostęp. Należy pamięta, że najpierw muszą być dostarczone na atakowaną maszynę, co może się wiązać z ich wykryciem przez oprogramowanie antywirusowe.
Aby je umieścić na atakowanym komputerze można np. uruchomić w katalogu, w którym się znajdują, wbudowany serwer Pythona:

```
python -m SimpleHTTPServer
```

a potem sesję Meterpretera na zdalnej maszynie.

Działania powłamaniowe

Kolejnym krokiem po uzyskaniu dostępu do określonego urządzenia może być uzyskanie informacji o innych użytkownikach i próba złamania haseł lub uzyskanie innych danych uwierzytelniających. W tym mogą być pomocne takie narzędzia jak *Mimikatz*, *Impacketer*, *TruffleHog*, *John the Ripper* i inne, np. sniffery.

Inną czynnością wykonywaną po przełamaniu zabezpieczeń może być podniesienie uprawnień (ang. *privileges escalation*), które może mieć charakter *pionowy* (ang. *vertical*) lub *poziomy* (ang. *horizontal*). To pierwsze polega na próbie uzyskania dostępu do kont użytkowników bardziej uprzywilejowanych niż bieżący (zwykle administratorów), a drugie na przejęciu konta użytkownika na podobnym poziomie uprzywilejowania, ale posiadającego inne, interesujące z punktu widzenia testu penetracyjnego, uprawnienia. W tej operacji przydatne mogą być eksploity jądra systemu operacyjnego, aplikacji lub bazy danych. Można też wykorzystać błędy w konfiguracji środowiska, jeśli takie występują, np. pliki wykonywalne z ustawionymi uprawnieniami `setuid` i `setgid` w systemach uniksowych.

Trwałość dostępu i unikanie wykrycia

Uzyskany początkowo dostęp do atakowanego urządzenia może mieć nietrwały charakter. Przyczyn tej ulotności może być wiele, ale najczęściej jest nią użycie eksploitu/ładunku, który istnieje tylko w pamięci operacyjnej atakowanej maszyny. Zaletą takiego oprogramowania jest to, że zwykle nie jest ono wykrywane przez oprogramowanie zabezpieczające takie, jak np. programy antywirusowe. Niestety, po restarcie urządzenia eksploit przestaje istnieć. Dlatego testerzy powinni zapewnić inny, trwały dostęp do zaatakowanej maszyny. Można w tym celu np. zastąpić demony usług (serwery) wersjami, które zawierają tylne furtki lub są po prostu końmi trojańskimi, ale takie działania mogą być stosunkowo łatwo wykryte. Dodanie nowego użytkownika również może zostać zauważone. Lepiej jest skorzystać z dostępnych w systemie programów i poleceń. Np. można dodać polecenie nawiązania połączenia `ssh` lub `HTTPS` z komputerem testera o określonej porze dla demona `cron` (systemy uniksowe) lub podobnego zadania w systemie Windows (polecenie `SchTasks`).

Trwałość dostępu i unikanie wykrycia

Ogólnie, zapewnienie trwałości dostępu do zaatakowanego urządzenia polega głównie na uruchomieniu powłoki poleceń dostępnej zdalnie dla atakującego. Wyróżniane są dwa rodzaje takich powłok:

wiązana powłoka (ang. *bind shell*) powłoka jest uruchamiana jako usługa zdalna, na wybranym przez atakującego porcie zaatakowanej maszyny

odwrócona powłoka (ang. *reverse shell*) w tym przypadku powłoka nawiązuje połączenie zwrotne z komputerem testera pod wskazanym adresem i na określonym porcie.

Drugi rodzaj powłoki może być łatwiejszy do wykorzystania, ponieważ wychodzący ruch wzbudza mniej podejrzeń, a większość zapór sieciowych jest skonfigurowana, aby blokować głównie ruch przychodzący.

Ekspansja

Po uzyskaniu dostępu do określonego urządzenia w sieci może się okazać, że testerzy odnajdą nowe potencjalne cele ataku, które wcześniej nie były widoczne, np. ze względu na segmentację sieci lub działanie takich mechanizmów zabezpieczających jak zapory sieciowe. Te cele mogą pojawić się nawet jeśli urządzenie, którego zabezpieczenia udało się przełamać znajduje się w DMZ. Zespół prowadzący testy powinien zatem powtórzyć czynność zbierania informacji i być może skanowania podatności, a następnie spróbować uzyskań dostęp do nowo odkrytych urządzeń. Takie działanie nazywane jest ekspansją (ang. *pivoting*). Czasem może mieć ono charakter lokalny, tzn. przejęcie konta użytkownika o wyższych uprawnieniach na tym samym urządzeniu też określa się tym mianem.

Zacieranie śladów

Jednym z zadań testerów penetracyjnych jest takie przejęcie kontroli nad podatnym urządzeniem, aby zostało to jak najdłużej, a najlepiej w ogóle, niezauważone. Może to wymagać rozwiązania dwóch problemów:

1. pobranie eksfiltorwanych danych z zaatakowanej maszyny,
2. usunięcie śladów włamania.

Pierwszy problem można rozwiązać nawiązując szyfrowane połączenie wychodzące, np. HTTPS i transmitując dane względnie małymi porcjami do popularnej usługi, np. *Google Drive* lub *Dropbox*. Ruch `HTTPS` zwraca mniejszą uwagę, niż np. połączenie wykorzystujące protokół `ssh`. Niektóre poradniki odnośnie do testów penetracyjnych sugerują użycie *steganografii*, ale w praktyce takie podejście nie jest często stosowane. Ogólnie, pobranie eksfiltrowanych danych powinno się odbywać *ukrytymi kanałami* (ang. *covered channels*).

Zacieranie śladów

Drugi z problemów przedstawionych na slajdzie 23 wymaga bardziej kompleksowego podejścia. Jak zostało to stwierdzone wcześniej, preferowane jest użycie eksploitów/ładunków, które pozostają jedynie w pamięci operacyjnej, a nie masowej. Trwały dostęp testerzy powinni sobie zapewnić stosując dostępne w systemie programy i narzędzia. Ważnym elementem ukrywania działań jest usunięcie wpisów w dziennikach systemowych. W przypadku systemów rodziny Windows może to wymagać modyfikacji *Historii aktywności*, jak i *Dziennika Zdarzeń*. W systemach uniksowym ta informacja jest bardziej rozproszona. Może być przechowywana w katalogu domowym użytkownika, w ukrytym pliku. Zależnie od powłoki ten plik może mieć postać tekstową, binarną lub pośrednią. Zwykle nazywa się `.history` lub `.histfile`, a jego zawartość jest dostępna za pomocą polecenia `history`. Globalnie, informacja o działaniach użytkowników w systemie odkładana jest w plikach znajdujących się domyślnie w katalogu `/var/log`.

Zacieranie śladów

Mogą to być pliki tekstowe (np. `messages` lub `syslog`), albo binarne (np. *wtmp* dla polecenia `last`). Osobną usługę rejestracji (ang. *journal*) dostarcza program `systemd`. Pliki tworzone przez nią są składowane w katalogu wskazanym w pliku konfiguracyjnym `/etc/systemd/journald.conf`.
Należy zauważyć, że oba rodzaje systemów operacyjnych mogą być tak skonfigurowane, aby przekazywać dane o zachodzących zdarzeniach do zdalnej usługi, np. *Security Information and Event Management* — SIEM. W takim przypadku ukrycie śladów eksploatacji jest utrudnione i może wymagać przejęcia kontroli lub wcześniejszego zablokowania tej usługi.
Warto również podkreślić, że ukrycie śladów powinno, w miarę możliwości, polegać na *modyfikacji* plików dzienników, a nie na ich usunięciu. To ostatnie podejście należy stosować tylko w ostateczności, gdyż łatwo je zauważyć i budzi (uzasadnione) podejrzenia.

Pytania

?

Koniec

Dziękuję Państwu za uwagę!