

X Window System - wygoda ponad bezpieczeństwo

Autor: [Aleksander Czarnowski](#)

Bardzo często wygodne i pożyteczne dla użytkownika rozwiązania informatyczne okazują się bardzo złe z punktu widzenia bezpieczeństwa. Gdyby bezpieczeństwo sprowadzić do bardzo prymitywnego modelu, można by założyć iż jest to zbiór restrykcji i ograniczeń. Dla użytkownika taka sytuacja jest nie do zaakceptowania. Być może właśnie to tłumaczy olbrzymią popularność systemów, które nie były projektowane z myślą o bezpieczeństwie.

W świecie Uniksa dobrym przykładem, potwierdzającym powyższą tezę, może być X Windows. Użytkownicy raczej by z nich nie zrezygnowali - praca w okienkach w porównaniu z shellem to zupełnie inna jakość. Z drugiej strony X Windows może być poważnym problemem z punktu widzenia bezpieczeństwa.

Architektura i zagrożenia

X Windows jest zbudowany przy użyciu architektury klient-serwer. W przypadku X Windows klient jest tym, co zwykle określamy mianem serwera. Komunikacja między serwerem a klientem odbywa się za pomocą X protocol i dzięki temu może być lokalna lub zdalna. Terminal nie oferuje żadnych możliwości oprócz przetwarzania wiadomości przychodzących za pomocą X protocol.

Rzadko zdarza się, aby interfejs graficzny był kojarzony z niebezpieczeństwem. Tym niemniej możliwości, jakie oferują X Windows pozwalają na wykorzystanie ich w nie zawsze legalnych celach. Dzięki możliwości podłączenia się poprzez sieć do każdego otwartego X Display można przechwytywać zawartość okienek i całego ekranu, odczytywać klawisze wprowadzane przez użytkownika z klawiatury oraz zdalnie uruchamiać aplikacje.

W takim przypadku zdobycie hasła roota lub instalacja konia trojańskiego staje się bardzo prosta. Wykrycie maszyny, na której jest uruchomiony X Windows nie przedstawia żadnego problemu - można to zrobić na przykład za pomocą skanera portów. Standardowo X Windows korzysta z portu 6000 (oraz wyższych). Nie ma jednak żadnego powodu, dla którego inne aplikacje czy demony nie mogłyby korzystać z tego portu. Odpowiedź skanera, że port 6000 jest otwarty, nie oznacza jeszcze, że wykorzystują go X Windows. Problemy z bezpieczeństwem w X Windows są również identyfikowane przez skanery zabezpieczeń. Jedno z pierwszych tego typu narzędzi - SATAN, sprawdza na przykład, czy na wskazanej maszynie można otworzyć display X-ów.

Zabezpieczenie

Istnieją dwie podstawowe metody zabezpieczenia X Windows. Pierwsza z nich opiera się na autentykacji hosta, czyli na sprawdzeniu adresu IP komputera, który próbuje się połączyć. Wadą tej metody jest widoczna od razu - z uprawnionego komputera może połączyć się wielu użytkowników. Nie daje więc to nam żadnej kontroli nad ich poczynaniami. Istnieje też druga metoda, pozwalająca na weryfikację każdego klienta, a nie maszyny, z której został uruchomiony. Aby poprawnie zabezpieczyć X Windows, należy skorzystać z obu metod.

Xhost

Program Xhost pozwala na zarządzanie mechanizmem autentykacji hostów. Wywołanie xhost bez żadnych parametrów spowoduje podanie listy obecnie dopuszczanych adresów. Aby dodać kolejny adres, wystarczy wywołać polecenie

```
xhost z parametrem + [nazwa_hosta].
```

Zezwoli to każdemu użytkownikowi z tego adresu na połączenie z naszym X serwerem. Aby otworzyć nasz X serwer dla całego świata, wystarczy wpisać:

```
xhost +.
```

Oczywiście nigdy nie należy udostępniać X Windows wszystkim użytkownikom na całym świecie. Zamknięcie dostępu dla wybranego komputera jest równie proste. Polecenie:

```
xhost -[nazwa_hosta]
```

wystarczy, aby odciąć dostęp. Dla nas jednak dużo cenniejszą możliwością jest zamknięcie dostępu przez wywołanie "xhost -". W ten sposób zabezpieczyliśmy nasz serwer przed atakami polegającymi na podpięciu się pod X display.

Jeśli w chwili wydawania polecenia xhost - będą otwarte połączenia między komputerami, to nie zostaną przerwane. Należy o tym pamiętać i restartować serwer po dokonaniu zmian.

Użytkownicy lokalni

Odcięcie X Windows od świata zewnętrznego za pomocą polecenia xhost - nie likwiduje w pełni problemu zdalnego dostępu. Jeśli atakujący może się zalogować zdalnie na serwer, to ominie zabezpieczenie. Można tego dokonać na przykład za pomocą telnetu czy ssh. Z drugiej strony, jeśli użytkownik posiada konto na atakowanej maszynie, ma tysiące innych możliwości, a X Windows nie będzie pierwszym celem.

Kontrola dostępu

Podstawowa wada zabezpieczenia polegającego na autentykacji hostów została wyeliminowana za pomocą metody MIT-MAGIC-COOKIE. Pomysł jest bardzo prosty. Oba komputery, które się ze sobą komunikują, muszą znać pewien wspólny sekret - w tym wypadku jest to pewien ciąg bajtów. Każdy klient, który chce się komunikować z serwerem, musi znać ten ciąg. Niestety i ta metoda ma swoje słabe strony. Po pierwsze sekret - ciąg bajtów - musi w jakiś sposób dotrzeć do wszystkich legalnych użytkowników. Jak wiadomo transport kluczy, haseł lub innych poufnych informacji zawsze naraża je na kradzież.

W przypadku sieci jest to o tyle proste, że można podsłuchiwać ruch sieciowy. Nawet jeśli dystrybucja przebiegnie poprawnie, to podczas komunikacji między stronami interesujący nas ciąg bajtów jest przesyłany. To spowodowało, że od X.11 Release 5 istnieje możliwość szyfrowania klucza - nigdy nie jest przesyłany przez sieć w formie otwartej.

Odmowa usługi

Polecenie 'xhost -' nie daje żadnej ochrony przed atakami typu Denial of Service (odmowa usługi), skierowanymi na X Windows. Starsze wersje X Windows są podatne na następujący atak:

```
$ telnet 127.0.0.1 6000
```

Domyślnie na porcie TCP 6000 nasłuchuje pierwszy X serwer. W przypadku gdy po wykonaniu tej komendy ekran X Windows zawiesi się, najwyższy czas zainstalować nową wersję X Windows. W zależności od implementacji, zawieszenie może trwać kilka sekund lub tak długo, jak będzie istniało połączenie.

.Xauthority

W pliku .Xauthority, znajdującym się w domowym katalogu użytkownika, przechowywany jest klucz. Tylko właściciel powinien mieć prawo zapisu i odczytu (-rw-----). Teoretycznie nie powinno się nigdy eksportować swojego domowego katalogu za pomocą NFS-a. Niestety, w pewnych środowiskach (np. sieci uniwersyteckie) takie rozwiązanie może okazać się niemożliwe.

Xauth

Program Xauth służy do zarządzania autoryzacjami użytkowników. Wiele źródeł podaje metodę dystrybucji kluczy za pomocą polecenia:

```
$ xauth extract - $DISPLAY | rsh jakis_host.com xauth merge -
```

Pierwsza część polecenia powoduje wyświetlenie klucza dla obecnego Display. Informacja ta zostaje dalej przekazana do zdalnego shella (rsh), który uruchamia na maszynie jakis_host.com program xauth. Xauth na zdalnej maszynie doda klucz do pliku .Xauthority.

Przedstawiona metoda ma poważną wadę. Komunikacja za pomocą rsh nie jest w żaden sposób chroniona. Potencjalny włamywacz będzie mógł bez problemu ją podsłuchiwać. Dlatego zalecane jest korzystanie z odpowiednika rsh - ssh (SecureShell). W ten sposób za pomocą szyfrowania komunikacja będzie chroniona.

Nie jest to pełny opis zabezpieczenia systemu X Windows. Moim celem było pokazanie podstawowych mechanizmów bezpieczeństwa, jakimi X Windows się posługuje. Mam nadzieję, że pokazałem również, jak rozwiązania wygodne mogą stwarzać problemy. W dzisiejszym świecie nawet interfejs graficzny może stwarzać bardzo duże niebezpieczeństwo. Coraz więcej rozwiązań będzie pracowało w sieciach - ile z nich może stanowić

poważny problem z punktu widzenia bezpieczeństwa?

Warto pamiętać, że siła X Windows jest widoczna tak naprawdę podczas pracy w sieci. To sprawia, że należy filtrować X protocol na firewallach. Pomiędzy różnymi wersjami X Windows występują pewne różnice, istnieją też pewne mutacje mechanizmów autoryzacji użytkowników. Dlatego podczas zabezpieczania swojej wersji X Windows najlepiej skorzystać z dokumentacji dołączonej do pakietu.

Warto przeczytać

* Crash Course in X Windows Security - [www.rootshell.com /docs /X.security](http://www.rootshell.com/docs/X.security)

* Securing X Windows - [ciac.llnl.gov /ciac /documents /CIAC-2316 Securing X Windows.pdf](http://ciac.llnl.gov/ciac/documents/CIAC-2316_Securing_X_Windows.pdf)

Dziękuję kolegom z firmy za pomoc podczas pisania artykułu.