

Dziury w Checkpoint SPLAT

<http://ipsec.pl/dziury-w-checkpoint-splat.html>

Oparta o Linuksa bezpieczna platforma Checkpointa (SPLAT) jest podatna na kilkanaście dziur typu przepełnienia bufora. Co ciekawe, są one możliwe do uruchomienia pomimo że system został poddany starannemu hardeningowi. Szczegółowe informacje na ten temat opublikowała hiszpańska grupa PenTest.

Opisana w artykule <http://www.checkpoint.com/products/secureplatform/index.html> platforma oparta o Linuksa oparta o jądro RedHat jest podatna na hardeningowy dystrybucja Linuksa oparta o jądro RedHat. Zabezpieczenia wbudowane w SPLAT to szereg zabezpieczeń znanych z projektu <http://pax.grsecurity.net/> (blokady wykonywalności stosu i sterty, losowe adresy bazowe obu struktur, ASLR, ASCII Armor) a także własne rozwiązania Checkpointa - CPSHELL (ograniczony shell linuksowy wyposażony m.in. w filtrowanie niebezpiecznych znaków). Na platformie tej może działać firewall Checkpointa FW-1 w wersjach m.in. R55 oraz NGX R60 (platforme wybiera klient).

Hiszpanie opisali szereg ataków, które są możliwe do przeprowadzenia na platformie nawet pomimo opisanych wyżej zabezpieczeń. Z szybkiego przeglądu obszernego dokumentu wynika, że w większości są to dziury typu buffer overflow. Autorzy przyznają, że wszystkie które przetestowali można wykorzystać tylko lokalnie (czyli mając shell w systemie), ale nie wykluczają że część z nich da się też wykorzystać zdalnie. W praktyce będzie to jednak utrudnione, bo normalnie firewallo Checkpointa nie odpowiada na pingi (czasem mają jednak otwarte wysokie porty dla zarządzania).

Autorzy podkreślają fakt, że SPLAT posiada certyfikat bezpieczeństwa na poziomie EAL4+ według Common Criteria. Należy jednak pamiętać o tym, że żaden formalny certyfikat bezpieczeństwa nie gwarantuje faktycznego bezpieczeństwa systemu, w sensie odporności na wszystkie znane aktualnie i przyszłe exploity.

Pełny raport można znaleźć tutaj:

- http://www.pentest.es/checkpoint_ack.pdf > HugoVázquezCaramés, "CheckPointSecurePlatformHack" >