INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTERAD 2026-03-19

EXAMENSARBETE Efficient Fuzzing of Web APIs in Embedded Environments
STUDENT Karl Alemo, Fredrik Orheim
HANDLEDARE Alma Orucevic-Alagic (LTH), Victoria Vucic (Axis Communications)
EXAMINATOR Per Runeson (LTH)

Viktiga saker att tänka på innan du börjar ”Fuzza”

POPULÄRVETENSKAPLIG SAMMANFATTNING **Karl Alemo, Fredrik Orheim**

För varje dag blir allt fler av våra prylar kopplade till internet. De behöver ständigt testas för att garantera deras säkerhet och funktion. I detta arbete jämförs olika datorprogram, så kallade "Fuzzers", som automatiskt skapar massvis av tester och kan hitta fel i en nätverksansluten produkt baserat på dess tekniska beskrivning.

Innan mjukvara uppdateras eller lanseras behöver ett företag garantera att allt fungerar som tänkt och är säkert genom testning. För ett litet system med ett fåtal funktioner är detta inget problem att göra för hand, men i ett stort komplext system som påverkar ett hundratal produkter kräver detta enorma resurser. Med hjälp av "fuzzing" kan delar av detta manuella arbete göras automatiskt!

"Fuzzing" är en automatisk testmetod där mjukvaran bombarderas med en massa oväntad och slumpmässig data för att hitta fel och säkerhetsbrister baserat på dess tekniska beskrivning. Metoden har funnits sedan 80-talet, men har de senaste 10 åren börjat bli relevant för att testa en produkts kommunikation med internet. I vårt examensarbete har vi tagit fram vilka aspekter man behöver tänka på när man vill börja "fuzza" på ett större företag och även jämfört olika publikt tillgängliga nätverks-fuzzers för att avgöra vilka som är mest lämpade för användning.

Resultatet visar att de utvecklare och testare som deltagit i vår studie vill att ett "fuzzing"-program ska vara lätt att lära sig och använda, samt att det ska vara enkelt att återskapa hittade fel så att de går att lösa. Dessutom ska programmet såklart vara effektivt på att hitta många fel och gå att ställa in efter vad man vill testa. Andra aspekter som kom fram var att man gärna vill att programmet uppdateras regelbundet och går att använda ur en juridisk aspekt.

I vår jämförelse av öppet tillgängliga fuzzers upptäckte vi att de flesta av programmen antingen inte uppdaterades längre eller ansågs ha många säkerhetsbrister. De få alternativ som ansågs vara säkra nog utvärderades efter effektivitet genom att testa dem mot riktiga produkter i totalt 583 timmar. En analys av hur många olika typer av fel programmen hittade och programmens förmåga att skapa test för så många olika funktioner som möjligt hos produkterna, visade att ’EvoMaster’ och ’Schemathesis’ generellt var de bäst lämpade alternativen för en företagsmiljö.

Fynden som gjorts i studien konkretiserar och gör det mätbart vad ett större företag behöver från ett automatiskt testverktyg. Under arbetets gång utvecklades även en produkt som redan idag kan användas för att automatiskt och enkelt testa en produkts nätverksfunktioner. "Fuzzing" kommer förhoppningsvis att leda till att fler fel kan upptäckas tidigare i utvecklingen av mjukvara och bidra till en fortsatt ökad säkerhet och funktionalitet av alla våra prylar.