**Executive Recommendation for Fuzzing Firefox**

Based on the team’s cost/benefit analysis, we do not recommend requiring fuzz testing of Firefox, but this is an activity that could be added as future budget allows.

Four fuzzing products were included in the analysis: two open source solutions (Sulley and Peach Community) and two commercial solutions (Peach Professional and Codenomicon Defensics).

The team talked with sales staff of both Codenomicon and Peach Fuzzer for budgetary, ballpark cost estimates for their products. Both products are licensed on an annual subscription model, and costs will vary based on the number of protocols to be fuzzed.

The team also interviewed external security professionals with experience in fuzzing, and have incorporated those results into our labor estimates.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Sulley** | **Peach Community** | **Peach Professional** | **Codenomicon** |
| **Implementation - Licensing (Year 1)** | $- | $- | $21,000.00 | $15,000.00 |
| **Implementation - Labor ($50/man-hour)** | $10,000.00 | $15,000.00 | $4,000.00 | $4,000.00 |
| **Software Maintenance (Year 2)** | $- | $- | $21,000.00 | $15,000.00 |
| **Ongoing Labor (Year 2) ($50/man-hour)** | $3,750.00 | $7,500.00 | $2,000.00 | $2,000.00 |
| **Software Maintenance (Year 3)** | $- | $- | $21,000.00 | $15,000.00 |
| **Ongoing Labor (Year 3) ($50/man-hour)** | $3,750.00 | $7,500.00 | $2,000.00 | $2,000.00 |
|  |  |  |  |  |
| **Total Estimated 3-Year Costs** | $17,500.00 | $30,000.00 | $71,000.00 | $53,000.00 |

Internal labor costs for Security Engineers were based on GIAC Enterprise’s PMO chargeback rate of $50/hour. “Ongoing labor” would include creating new data models and tests as time goes on. Running the same tests on new versions of Firefox is not included in the “ongoing labor” costs, as it would not contribute significantly in this area.

Should management decide to continue to pursue fuzzing across the enterprise, the team recommends hiring a Security Engineer with at least 5 years of experience in fuzzing applications using Sulley (or Spike – which easily translates to Sulley).