



بخشی از ترجمه مقاله

عنوان فارسی مقاله :

یک آشکارساز (دیکدر) سر ریز بافر پویای کاربردی

عنوان انگلیسی مقاله :

A Practical Dynamic Buffer Overflow Detector



توجه !

این فایل تنها قسمتی از ترجمه میباشد. برای تهیه مقاله ترجمه شده کامل با فرمت ورد (قابل ویرایش) همراه با نسخه انگلیسی مقاله، [اینجا](#) کلیک نمایید.



بخشی از ترجمه مقاله

6- نتیجه گیری

6. Conclusions

We presented CRED, a practical dynamic buffer overflow detector for C programs. Our solution is built upon Jones and Kelly's technique of tracking the referent object of each pointer. Compared to the previous system, our solution does not break existing codes that compute with out-of-bound addresses and is significantly more efficient by limiting the buffer overrun checks to strings.

Our experimental results demonstrate the compatibility of our technique with commonly used programs and its effectiveness in detecting buffer overrun attacks in vulnerable programs. The overheads experienced range from 1% to 130%, depending on the use of strings in the application. These numbers are significantly better than previously published results on dynamic bounds checking. Further improvement in performance is possible using known techniques for optimizing bounds checking[15], and using static verification to reduce the portion of code that requires instrumentation[9].

ما CRED، یک دیکدر سرریز بافر پویای کاربردی را برای برنامه C ارائه دادیم. راه حل ما بر اساس روش ردیابی شیء ارجاعی هر اشاره گر (تکنیک جونز و کلی) ساخته شده است. در مقایسه با سیستم قبلی، راه حل ما کد های موجود را که، آدرس های برون مرزی را محاسبه می کنند را نقض نمی کند و به طور قابل ملاحظه ای با حذف کنترل های بیش از حد بافر برای رشته ها کارآمدتر است. نتایج تجربی ما سازگاری تکنیک ما با برنامه های استفاده شده رایج و اثربخشی آن در تشخیص حملات سرریز بافر در برنامه های آسیب پذیر را نشان می دهد. سربارها بردی از 1٪ تا 130٪، را بسته به استفاده از رشته ها در برنامه های کاربردی تجربه کرده اند. این اعداد به طور قابل توجهی بهتر از نتایج منتشر شده قبلی در bounds checking پویا هستند. بهبود بیشتر عملکرد با استفاده از تکنیک های شناخته شده برای بهینه سازی bounds checking [15] و استفاده از اعتبارسنجی ایستا برای کاهش بخشی از کد که نیاز به ابزار دقیق دارد، ممکن است [9].



توجه!

این فایل تنها قسمتی از ترجمه میباشد. برای تهیه مقاله ترجمه شده کامل با فرمت ورد (قابل ویرایش) همراه با نسخه انگلیسی مقاله، [اینجا](#) کلیک نمایید.

برای جستجوی جدیدترین مقالات ترجمه شده، [اینجا](#) کلیک نمایید.