پیشرفت های اخیر در فازینگ

(پاسخ سوالات)

فازینگ در برنامه های پایتون چگونه است؟

برنامه های پایتون به دلیل Type Safe بودن خود زبان پایتون همانند برنامه های نوشته در زبان هایی همچون C/C++ امکان فاز شدن برایشان وجود ندارد. به همین دلیل بیشتر تمرکز بر روی یافتن آسیب پذیری های رفتاری و یا منطقی گسترده شده است. از طرف دیگر خود مفسر پایتون (یا دیگر مفسر ها همچون PHP) را میتوان به شکل کلاسیک فاز کرد چرا که این مفسر ها در زبان های سطح پایین همچون C/C++ نوشته شده اند. برای نمونه رجوع کنید به مقاله LangFuzz :

Fuzzing with Code Fragments
Christian Holler, et al.

چند نوع ورودی برای فاز کردن وجود دارد (string, int, ...)؟

ورودی های موجود برای تولید داده های تصادفی با گوناگونی ورودی های قابل قبول در برنامه زیر تست رابطه مستقیم دارد. به دلیل رایج بودن ورودی هایی چون رشته ها، اعداد، سمبول ها و ... فازرها به شکل پیشفرض توانایی تولید مجموعه های تصادفی آنها را دارند.

روش فازینگ و یا Code Audit سریعتر به جواب میرسید؟

استفاده از فازر به دلیل گسترش پذیری (در مقابل عدم گسترش پذیری نیروی کارشناس برای Code Audit) مطمعنا زودتر به جواب خواهد رسید، اما کیفیت آسیب پذیری های پیدا شده تحت تاثیر قرار میگیرد، زیرا یک کارشناس خبره در امر Code Audit میتوان آسیب پذیری های منطقی و یا عمیقی را پیدا کند که هیچ سیستم کشف اتوماتیکی همچون فازر ها قادر به یافتن آن نیست. برای رسیدن به نتیجه مطلوب باید توازنی میان Code Audit و فازینگ برقرار کرده و هردو را به شکل موازی در سازمان پیش برد.

درمورد جایگزین کردن انسان خود شما چه ایده‌ای دارید؟

اگر درست متوجه منظورتان شده باشم، به نظر من روند خودکار سازی کشف آسیب پذیری (بدون نیاز به انسان) میتواند با فاصله گرفتن از آنالیز سنگین و جهت گرفتن به سوی بهره گیری از آنالیز ایستا و روش های یادگیری ماشین (جهت یادگیری فرمت ورودی نرم افزار زیر تست) می تواند به نتایج و پیشرفت مطلوبی برسد.

آیا فازر Shellphish متن باز است؟

بله این چارچوب کشف آسیب پذیری متن باز بوده و در آدرس زیر قابل دسترس است.

https://github.com/shellphish

اصولا فازر ها برای اجرای کدهای اسمبلی از چه مکانیزمی استفاده میکنند؟ مثلا debugging یا instrument کردن با Pin؟

فازر ها نیازی به اجرای کد اسمبلی ندارند، اطلاعات مورد نیاز آنها برای بررسی جریان اجرا (Control Flow) و یا جریان داده (Data Flow) توسط ابزار هایی همچون Pin، DynamoRio و یا Valgrind قابل بازیابی است، این اطلاعات را میتون با Single Stepping در یک دیباگر و تحلیل هر دستور نیز به دست آورد اما این روند به شدت کند خواهد بود. ابزار هایی هچون Pin توانایی فیلتر کردن رخداد های خاص (به طور مثال اجرای یک دستور اسمبلی خاص) را دارند به همین دلیل برای جمع آوری اطلاعات جریان داده و اجرا بسیار کاراتر عمل میکنند.

برای فاز کردن کرنل از چه شیوه ای استفاده میشود؟ با توجه به اینکه ابزار instrument کردن مثلا Pin برای کرنل وجود ندارد؟

برای فاز کردن در صورت موجود بودن کد منبع میتوان کدهای مورد نیاز برای instrument کردن را در هسته تزریق کرده و سپس آن را کامپایل کنید. از طرف دیگر شبیه ساز های کامل همچون Qemu توانایی instrument کل سیستم عامل (منجمله کرنل) را دارند و میتوان از درگاه های ارائه شده توسط چنین شبیه سازهایی برای فازکردن و instrument کرنل بهره گرفت، برای اطلاعات بیشتر در همین مورد میتوانید فازر هایی چون kAFL و یا TriforceAFL مطالعه کنید.

نقش هوش مصنوعی در فازینگ چیست؟

نقشی که من برای هوش مصنوعی در فازینگ قائل میشوم، یادگیری ساختار های پیچیده ورودی برای تولید test-case های بهتر است. رجوع کنید به مقالات ۳ تا ۵ در صفحه ۱۴ اسلاید.

آیا KFUZZ توانایی فاز کردن پرتکل هایی چون HTTPS را نیز دارد؟

KFUZZ یک فازر ماژولار است از همین رو میتوان تمامی پرتکل ها و یا داده هایی که در کرنل parse میشوند (همچون SMB، RDP و ...) را با ایجاد رابط صحیح در سطح کاربر، فاز کرد.

شرکت کنندگان در مسابقه CGC/DARPA مسئله نیاز به سخت افزار های زیاد را چگونه حل کردند؟

متاسفانه این مشکل به سادگی قابل حل نیست، البته با بهره گیری از الگوریتم های صحیح زمانبندی تا حد بسیار کمی قابل بهبود است. شرکت کننده‌گان در چالش CGC مجهز به سامانه های قوی بودند که در عکس زیر نیز موضوع روشن است.

استفاده از interruptها و Illegal Instruction سرعت حرکت thread را کاهش نمیدهد؟

بله این مسئله میتواند بر روی سرعت اجرای فازر تاثیر بگذارد، اما از آنجایی که بلاک های کوچکتر از ۵بایت تنها درصد اندکی (کمتر از ۱۰ درصد) از برنامه را تشکلی میدهند، این کندی قابل چشم پوشی است.

فازینگ در وب اپلیکیشن ها چه فرقی بانرم‌افزارها دارد؟

در حوضه تخصص بنده نیست، اما میتوانید رجوع کنید به مقالاتی چون :

An automated black box approach for web vulnerability identification and attack scenario generation

Rim Akrout

نرمافزارهایی که در هر بار اجرا کد خود را تغییر میدهند چطور فاز میشوند؟

اگر درست متوجه شده باشم منظورتان کدهایی چون JIT compilerها است. Instrument کردن کد تولید شده توسط این نوع کامپایل هار بسیار سخت است، اما بستر تولید کننده کد JIT، به دلیل ثابت بودن، توسط روش های معمول قابل فاز کردن است. فاز کردن کد های JIT همچون کدهای JavaScript به طور معمول با استفاده از روش های Grammar Based Fuzzing قابل انجام است.

متاسفانه سوال زیر هم برای بنده ناخانا بود :