Industrieel Ingenieur Informatica

Werkplan Masterproef

Naam student: Lennert Franssens

Datum: 22 oktober 2021

Titel: Boosting Multi-Variant Execution through Modern OS Extensions

Bedrijf of onderzoeksgroep
Naam: Computer Systems Lab
Tel: /
Promotors: prof. dr. Bart Coppens, prof. dr. ir. Bjorn De Sutter
mailadressen: Bart.Coppens@UGent.be, Bjorn.DeSutter@UGent.be
Begeleider: dr. ir. Bert Abrath
mailadres: Bert.Abrath@UGent.be

Bestaande situatie en probleemstelling

MVEE's bieden een veilige, maar trage manier om veiligheidsproblemen bij software te ontdekken. Door elke *system call* te monitoren, om het gedrag van verschillende varianten van een programma te testen, wordt een *overhead* geïntroduceerd. ReMon biedt hiervoor een handige oplossing. Er wordt gebruik gemaakt van *monitoring relaxation* zodat ongevaarlijke *system calls* niet gemonitord hoeven te worden. De gebruiker van ReMon kan zelf aangeven welke *system calls* binnen de context van het programma als ongevaarlijk worden gezien. Door meer *system calls* ongemonitord door te laten kan er aan een grotere snelheidswinst gedaan worden, met een beveiligingsafweging tot gevolg.

ReMon is dus een MVEE waarbij, in standaardconfiguratie, dezelfde beveiliging wordt geboden als een *native* MVEE, maar veel sneller is in uitvoering. Om dit mogelijk te maken is er een specifieke kernel patch nodig. Deze kernel patch is niet aanwezig in de reguliere Linux kernel, waardoor bestaande MVEE's geen gebruik maken van deze snellere implementatie.

Doelstelling van het project

Om een kernel patch te vermijden kan er gebruik gemaakt worden van moderne functies in de Linux kernel. De nieuwe Linux-functie Syscall User Dispatch kan in combinatie met eBPF en seccomp gebruikt worden om in de kernel bepaalde *system calls* al dan niet direct toe te laten. Een nieuw ontwerp van de bestaande applicatie, dat gebruik maakt van voorgaande nieuwe technologieen, kan ervoor zorgen dat de kernel patch niet meer nodig is. Het is belangrijk dat met deze nieuwe implementatie dezelfde snelheden behaald kunnen worden als de implementatie met de kernel patch.

Planning en mijlpalen (per week – je mag weken toevoegen):

Week	Type	Beschrijving
27 sep – 3 okt	Planning	Literatuurstudie: • Secure and Efficient Application Monitoring and Replication (ReMon) – paper • Informatie opzoeken over eBPF en seccomp
	Scriptie	
	Deadline	
4 okt – 10 okt	Planning	Literatuurstudie: • Brede context van MVEE's begrijpen • Informatie opzoeken over eBPF en seccomp
	Scriptie	
	Deadline	
11 okt – 17 okt	Planning	Literatuurstudie: • Brede context van MVEE's begrijpen • Versnelde innovatie door middel van eBPF
	Scriptie	
	Deadline	

18 okt – 24 okt	Planning	Literatuurstudie:
		• Kernel patches
		• Hoe kan ik eBPF in deze technologie gebruiken? (seccomp en ptrace)
		Technologieverkenning:
		• ReMon
		  ○ Installeren van applicatie
		  ○ Installeren van kernel patch voor IP-MON
		  ▪ Kernel patch begrijpen

	Scriptie	
	Deadline	• Werkplan indienen
25 okt – 31 okt	Planning	Literatuurstudie: • Kernel patches • Advanced Techniques for Multi-Variant Execution en verwante literatuur Technologieverkenning: • ReMon ○ Code begrijpen
	Scriptie	
	Deadline	
1 nov – 7 nov	Planning	Literatuurstudie: • Advanced Techniques for Multi-Variant Execution en verwante literatuur • Syscall User Dispatch Technologieverkenning: • ReMon

		○ Testen van programma ○ Code begrijpen
	Scriptie	• Overleaf ○ Juiste .latex-bestand (UGent, fea) ○ Structuur van scriptie vastleggen ○ Opmaak
	Deadline	
8 nov – 14 nov	Planning	Literatuurstudie: • Advanced Techniques for Multi-Variant Execution en verwante literatuur • Syscall User Dispatch • System call monitoring ○ Snelheid en verschillende technieken Technologieverkenning: • ReMon ○ Code begrijpen
	Scriptie	• Woord vooraf schrijven
	Deadline	
15 nov – 21 nov	Planning	Literatuurstudie: • Advanced Techniques for Multi-Variant Execution en verwante literatuur • Syscall User Dispatch • System call monitoring ○ Security Technologieverkenning: • Experimenteren met eBPF: ○ Hoe schrijf ik een programma voor eBPF? ○ Hoe voer ik een programma in eBPF uit?
	Scriptie	• Inleiding schrijven
	Deadline	
22 nov – 28 nov	Planning	Literatuurstudie: • Verschil in implementatietechnologieën van eBPF ○ Snelheid in ontwikkeling ○ Snelheid in gebruik ○ Veiligheid? Technologieverkenning: • Experiementeren met eBPF ○ Voorbeelden uitvoeren (seccomp en ptrace)
	Scriptie	• Doelstelling beschrijven
	Deadline	
29 nov – 5 dec	Planning	Literatuurstudie: • seccomp, eBPF en ptrace ○ Security m.b.t. system calls Technologieverkenning:

		• Experimenteren met eBPF ○ System calls uitvoeren in verschillende contexten
	Scriptie	• Literatuurstudie beschrijven
	Deadline	
6 dec – 12 dec	Planning	Presentatie: • Tussentijdse presentatie maken ○ Waarover vertel ik? Inleiding. Beginsituatie schetsen.
	Scriptie	• Technologieverkenning beschrijven
	Deadline	
13 dec – 19 dec	Planning	Presentatie: • Tussentijdse presentatie maken ○ Wat heb ik tot nu toe gedaan? Wat ga ik doen in het tweede semester? Hoe zal ik dat aanpakken? Aan welke resultaten kan ik me verwachten na inwerking en literatuurstudie?
	Scriptie	• Bronvermelding van literatuurstudie
	Deadline	
20 dec – 26 dec	Planning	Presentatie: • Tussentijdse presentatie inoefenen
	Scriptie	
	Deadline	• Scriptie: Literatuurstudie laten nalezen • **Tussentijdse presentatie over vorderingen**
7 feb – 15 feb	Planning	Ontwerp: • Ontwerp van implementatie aanpassen zodat system calls naar IP-MON niet meer worden doorgestuurd.
	Scriptie	
	Deadline	
14 feb – 20 feb	Planning	Ontwerp: • Ontwerp van implementatie aanpassen zodat bepaalde system calls, die niet gemonitord moeten worden, door Syscall User Dispatch (eBPF en seccomp) worden afgehandeld.
	Scriptie	
	Deadline	
21 feb – 27 feb	Planning	Ontwerp: • Overleg nieuw ontwerp/architectuur van de implementatie met begeleider en promotor.
	Scriptie	
	Deadline	• Ontwerp nieuwe implementatie (met Syscall User Dispatch i.p.v. kernel patch) klaar.
28 feb – 6 mrt	Planning	Implementatie: • Nieuw ontwerp in bestaande ReMon-applicatie implementeren ○ IP-MON uit het systeem halen
	Scriptie	• Analyse beschrijven ○ Ontwerp van implementatie

		○ Motivatie van keuze
	Deadline	
7 mrt – 13 mrt	Planning	Implementatie: • Nieuw ontwerp in bestaande ReMon-applicatie implementeren ○ De doorgestuurde system calls uit IK-B ophalen in eBPF
	Scriptie	
	Deadline	
14 mrt – 20 mrt	Planning	Implementatie: • Nieuw ontwerp in bestaande ReMon-applicatie implementeren ○ Met Syscall User Dispatch (eBPF en seccomp) de doorgestuurde system calls uit IK-B afhandelen
	Scriptie	
	Deadline	
21 mrt – 27 mrt	Planning	Implementatie: • Nieuw ontwerp in bestaande ReMon-applicatie implementeren ○ Met Syscall User Dispatch (eBPF en seccomp) de doorgestuurde system calls uit IK-B afhandelen • Andere stukken van de ReMon-applicatie op nieuwe manier implementeren (eBPF) voor meer snelheidswinst
	Scriptie	
	Deadline	
28 mrt – 3 apr	Planning	Implementatie: • Controlleren of nieuwe implementatie werkt zoals het hoort ○ Implementatie bijsturen ○ Bugs uit code halen • Andere stukken van de ReMon-applicatie op nieuwe manier implementeren (eBPF) voor meer snelheidswinst Presentatie: • (Tussentijdse) presentatie maken ○ Waarover vertel ik? Inleiding. Beginsituatie schetsen.
	Scriptie	
	Deadline	• Scriptie: Laten nalezen van ±25 pag. • Invullen gegevens op Plato
4 apr – 10 apr	Planning	Implementatie: • Afronden concrete implementatie (in grote lijnen) Presentatie: • (Tussentijdse) presentatie maken ○ Mijn oplossing uitleggen.
	Scriptie	• Uitschrijven van concrete implementatie (corpus)
	Deadline	• **Werkend prototype**
11 apr – 17 apr	Planning	Praktisch: • Benchmarks schrijven ○ Voor randgevallen ○ Om overhead goed te kunnen meten Presentatie:

		• (Tussentijdse) presentatie maken ○ Is mijn oplossing goed? Reflectie en resultaten bespreken.
	Scriptie	• Kort abstract schrijven (5-10 lijnen in html voor Plato) • Extended abstract schrijven
	Deadline	• **Tussentijdse presentatie voor onderzoeksgroep**

18 apr – 24 apr	Planning	Praktisch:
		• Benchmarking op kleine (bestaande) programma's
		  ○ Om bugs op te sporen
		  ○ Om snelheidsmetingen (overheadsmetingen) te doen

	Scriptie	• Inventaris van alle software en technologie beschrijven • Duurzaamheidsreflectie (sdgs en globalgoals) – reflectie
	Deadline	

25 apr – 1 mei	Planning	Praktisch:
		• Benchmarking op grote (bestaande) programma's
		  ○ Om bugs op te sporen
		  ○ Om snelheidsmetingen (overheadsmetingen) te doen

	Scriptie	• Beschrijven van resultaten (benchmarking) • Nieuwe bronnen toevoegen
	Deadline	

2 mei – 8 mei	Planning	Praktisch:
		• Benchmarking van grote (bestaande) programma's
		  ○ Met nieuwe implementatie
		  ○ Met oude implementatie
		  ○ Met gewone MVEE

	Scriptie	• Beschrijven van resultaten (benchmarking) • Moeilijkheden beschrijven
	Deadline	Klaar met alle benchmarks om resultaten uit te kunnen schrijven in scriptie
9 mei – 15 mei	Planning	
	Scriptie	• Beschrijven van vergelijkende resultaten (benchmarking) • Uitschrijven van alle tests die gebruikt werden om oplossing te evalueren • Eigen beoordeling van het geleverde werk – reflectie
	Deadline	
16 mei – 22 mei	Planning	
	Scriptie	• Besluit uitschrijven • Nalezen
	Deadline	
23 mei – 29 mei	Planning	
	Scriptie	• Nalezen
	Deadline	Scriptie: 1e versie laten nalezen
30 mei – 5 jun	Planning	
	Scriptie	• Laatste correcties in scriptie
	Deadline	• Afgewerkte implementatie doorgeven aan promotors en begeleider

6 jun – 12 jun	Planning	
	Scriptie	• Laatste correcties in scriptie
	Deadline	Scriptie: Indienen op Plato
13 jun – 19 jun	Planning	Presentatie: • Presentatie afwerken ○ Feedback tussentijdse presentatie gebruiken om openbare verdediging op punt te zetten.
	Scriptie	
	Deadline	
20 jun – 26 jun	Planning	Presentatie: • Presentatie afwerken ○ Controle: juiste grafieken en grammaticale fouten verbeteren
	Scriptie	
	Deadline	
27 jun – 3 jul	Planning	Presentatie: • Inoefenen • Voorbereiden door alles nogmaals grondig door te nemen
	Scriptie	
	Deadline	• Logboek/e-mailrapportering indienen op Plato • Presentatie indienen op Plato • **Openbare verdediging**