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 techniek om veiligheidsproblemen met betrekking tot geheugencorruptie bij software te ontdekken. Ze voeren verschillende varianten (replicae) van eenzelfde programma uit en geven ze dezelfde invoer. Door de verschillende varianten te monitoren op het niveau van de *system calls*, kunnen MVEE's aanvallen een applicatie ontdekken. Als zo'n aanval ontdekt wordt, kan een programma, nog voor er schade toegericht wordt, gestopt worden. Om de *system calls* te kunnen monitoren, worden ze onderschept. Dat gebeurt vanuit een ander proces met een debug-API, wat voor een *overhead* zorgt.

ReMon biedt de optie om een aantal 'ongevaarlijke' *system calls* op een andere manier uit te voeren. De trage omweg wordt vervangen door een aparte component te injecteren in de adresruimte van het te testen programma. Op die manier worden *system calls* veel sneller afgehandeld. Deze specifieke implementatie is niet aanwezig in de reguliere Linux kernel maar moet via een kernelpatch toegevoegd worden, wat veel mensen liever niet doen. Daardoor wordt de ReMon applicatie en zijn snelle implementatie beperkt gebruikt.

## Doelstelling van het project

In deze thesis gaan we ReMon aanpassen zodat de functionaliteit van de kernelpatch (en de interactie die de monitor daarmee heeft) wordt vervangen door functionaliteit die gebruik maakt van nieuwe technologieën in de Linux kernel. Het is belangrijk dat na die aanpassingen dezelfde snelheden behaald kunnen worden als de implementatie met de kernelpatch.

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

| Week | Type | Beschrijving |
|---|---|---|
| 27 sep – 3 okt | Planning | Literatuurstudie:<br>• Secure and Efficient Application Monitoring and Replication (ReMon) – paper<br>• Informatie opzoeken over eBPF en seccomp |
| | Scriptie | |
| | Deadline | |
| 4 okt – 10 okt | Planning | Literatuurstudie:<br>• Brede context van MVEE's begrijpen<br>• Informatie opzoeken over eBPF en seccomp |
| | Scriptie | |
| | Deadline | |
| 11 okt – 17 okt | Planning | Literatuurstudie:<br>• Brede context van MVEE's begrijpen<br>• Gebruik van eBPF in combinatie met seccomp |
| | Scriptie | |
| | Deadline | |
| 18 okt – 24 okt | Planning | Literatuurstudie:<br>• Kernelpatches<br>• Hoe kan ik eBPF in deze technologie gebruiken? (seccomp en ptrace)<br>Technologieverkenning:<br>• ReMon<br>  ○ Installeren van applicatie<br>  ○ Installeren van kernelpatch voor IP-MON<br>  ▪ Kernelpatch begrijpen |
| | Scriptie | |
| | Deadline | • Werkplan indienen |
| 25 okt – 31 okt | Planning | Literatuurstudie:<br>• Kernelpatches<br>• Advanced Techniques for Multi-Variant Execution en verwante literatuur<br>Technologieverkenning:<br>• ReMon<br>  ○ Code begrijpen |
| | Scriptie | |
| | Deadline | |
| 1 nov – 7 nov | Planning | Literatuurstudie:<br>• Advanced Techniques for Multi-Variant Execution en verwante literatuur<br>• Syscall User Dispatch<br>Technologieverkenning:<br>• ReMon |

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

| | | |
|---|---|---|
| | | • Experimenteren met eBPF<br>  ○ System calls uitvoeren in verschillende contexten |
| | Scriptie | • Literatuurstudie beschrijven |
| | Deadline | |
| 6 dec – 12 dec | Planning | Presentatie:<br>• Tussentijdse presentatie maken<br>  ○ Inleiding en eginsituatie schetsen. |
| | Scriptie | • Literatuurstudie beschrijven<br>• Technologieverkenning beschrijven |
| | Deadline | |
| 13 dec – 19 dec | Planning | Presentatie:<br>• Tussentijdse presentatie maken<br>  ○ 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 | • Technologieverkenning beschrijven |
| | Deadline | |
| 20 dec – 26 dec | Planning | Presentatie:<br>• Tussentijdse presentatie inoefenen |
| | Scriptie | |
| | Deadline | • Scriptie: Literatuurstudie laten nalezen<br>• **Tussentijdse presentatie over vorderingen** |
| | | |
| 7 feb – 15 feb | Planning | Ontwerp:<br>• Ontwerp van implementatie aanpassen zodat system calls naar IP-MON niet meer worden doorgestuurd. |
| | Scriptie | |
| | Deadline | |
| 14 feb – 20 feb | Planning | Ontwerp:<br>• Ontwerp van implementatie aanpassen zodat bepaalde system calls, die niet gemonitord moeten worden, door Syscall User Dispatch (eBPF en seccomp) worden afgehandeld. |
| | Scriptie | |
| | Deadline | • Ontwerp nieuwe implementatie afhebben voor controle. |
| 21 feb – 27 feb | Planning | Ontwerp:<br>• 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:<br>• Nieuw ontwerp in bestaande ReMon-applicatie implementeren<br>  ○ IP-MON uit het systeem halen |
| | Scriptie | • Analyse beschrijven |

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

| | | |
|---|---|---|
| | | Presentatie:<br>• (Tussentijdse) presentatie maken<br>  ○ Is mijn oplossing goed? Reflectie en resultaten bespreken. |
| | Scriptie | • Kort abstract schrijven (5-10 lijnen in html voor Plato)<br>• Extended abstract schrijven |
| | Deadline | • **Tussentijdse presentatie voor onderzoeksgroep** |
| 18 apr – 24 apr | Planning | Praktisch:<br>• Benchmarking op kleine (bestaande) programma's<br>  ○ Om bugs op te sporen<br>  ○ Om snelheidsmetingen (overheadsmetingen) te doen |
| | Scriptie | • Inventaris van alle software en technologie beschrijven<br>• Duurzaamheidsreflectie (sdgs en global goals) – reflectie |
| | Deadline | |
| 25 apr – 1 mei | Planning | Praktisch:<br>• Benchmarking op grote (bestaande) programma's<br>  ○ Om bugs op te sporen<br>  ○ Om snelheidsmetingen (overheadsmetingen) te doen |
| | Scriptie | • Beschrijven van resultaten (benchmarking)<br>• Nieuwe bronnen toevoegen |
| | Deadline | |
| 2 mei – 8 mei | Planning | Praktisch:<br>• Benchmarking van grote (bestaande) programma's<br>  ○ Met nieuwe implementatie<br>  ○ Met oude implementatie<br>  ○ Met gewone MVEE |
| | Scriptie | • Beschrijven van resultaten (benchmarking)<br>• 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)<br>• Uitschrijven van alle tests die gebruikt werden om oplossing te evalueren<br>• Eigen beoordeling van het geleverde werk – reflectie |
| | Deadline | |
| 16 mei – 22 mei | Planning | |
| | Scriptie | • Besluit uitschrijven<br>• Woord vooraf uitschrijven<br>• Nalezen |
| | Deadline | |
| 23 mei – 29 mei | Planning | |
| | Scriptie | • Nalezen |
| | Deadline | Scriptie: 1e versie laten nalezen |
| | Planning | |

| | | |
|---|---|---|
| 30 mei – 5 jun | 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:<br>• Presentatie afwerken<br>  ○ Feedback tussentijdse presentatie gebruiken om openbare verdediging op punt te zetten. |
| | Scriptie | |
| | Deadline | |
| 20 jun – 26 jun | Planning | Presentatie:<br>• Presentatie afwerken<br>  ○ Controle: juiste grafieken en grammaticale fouten verbeteren |
| | Scriptie | |
| | Deadline | |
| 27 jun – 3 jul | Planning | Presentatie:<br>• Inoefenen<br>• Voorbereiden door alles nogmaals grondig door te nemen |
| | Scriptie | |
| | Deadline | • Logboek/e-mailrapportering indienen op Plato<br>• Presentatie indienen op Plato<br>• **Openbare verdediging** |