MARS: uma arquitetura para análise de *malwares* utilizando SDN

João M. ceron[1], Cíntia B. Margi[1], Lisandro Z. Granville[2]

[1]Escola Politécnica – Universidade de São Paulo – São Paulo – SP – Brasil
[2] Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
{ceron,cintia}@usp.br, lisandro@inf.ufrgs.br

Abstract. *To investigate characteristics from malicious code are essential to improve security systems. However, modern malwares require specific conditions to activate their actions in the target system. This thesis presents an specialized architecture to analyze malware by managing the analysis environment in a centralized way, including to control the sandbox and the elements that surrounds it. The proposed architecture enables to determine the network access policy, to handle the analysis environment resource configuration, and to manipulate the network connections performed by the malware. The experimental results showed that our solution can reveals unseen behaviors that are observerd in tradicional analysis solutions.*

Resumo. *Investigar características de códigos maliciosos é um processo essencial para aprimorar os sistemas de segurança. No entanto, malwares modernos requerem condições específicas no ambiente em que são executados para revelar seu comportamento malicioso. Esta tese propõe arquitetura flexível para analisar códigos maliciosos controlando de maneira unificada o ambiente de análise, incluindo o sandbox e os elementos que o circundam. Dessa maneira, é possível gerenciar regras de contenção, configuração dinâmica de recursos, e manipular o tráfego de rede gerado pelos malwares. Os resultados experimentais demonstraram que a arquitetura pode revelar comportamentos de malwares que não são exibidos em soluções tradicionais de análise.*

1. Introdução

Malware é um termo genérico para caracterizar os diferentes tipos de códigos maliciosos (*e.g.* vírus, *worms*, *trojan*). Os *malwares* representam um grande risco à segurança pois podem realizar diversas ações nocivas, causando danos aos sistemas computacionais. Para proteger os sistemas dessas ameaças, é importante desenvolver ferramentas que possam identificar comportamentos dos códigos maliciosos e assim aprimorar os mecanismos de defesa. Sistemas de detecção de intrusão (*IDS - Intrusion Detection Systems*) e antivírus precisam conhecer antecipadamente as características comportamentais dos códigos para desenvolver assinaturas e identificar ameaças. Neste contexto, as ferramentas de análise de *malware* possuem uma função essencial permitindo que as ações dos *malwares* sejam previamente caracterizadas.

Uma estratégia popular de investigação consiste em executar um arquivo suspeito em um *sandbox* – *i.e.*, sistema controlado – de modo a mapear as características comportamentais do *malware* quando executado. Com base no comportamento observado, um analista pode desenvolver assinaturas e aprimorar os mecanismos de

defesa. Sabe-se, no entanto, que certas famílias de *malwares* adaptam o seu comportamento tendo em vista os recursos disponíveis do sistema que são executados. *Malwares* como o Agobot [McLaughlin 2004] podem alterar seu comportamento de execução quando detectam um sistema de análise. A ameaça *Red October*, por exemplo, caracteriza-se por executar funcionalidades exclusivamente em ambientes governamentais ou em redes pertencentes às embaixadas [Virvilis and Gritzalis 2013].

Soluções de análise de *malware* negligenciam o controle do ambiente que circunda o *sandbox*, mesmo sabendo que a configuração deste ambiente pode afetar as ações desempenhadas pelo *malware* [Laurianne 2004]. Em especial, as soluções atuais apresentam limitações ao prover funcionalidades relacionadas com a camada de rede. Algumas soluções abordam a camada de rede no contexto de análise; no entanto, as funcionalidades são implementadas de forma parcial, focando apenas na contenção de tráfego e não na integração dos recursos disponíveis. Diante disso, observa-se que as soluções atuais podem ser aprimoradas, em especial, abordando a seguinte pergunta de pesquisa: como investigar *malwares* avançados que adaptam suas funcionalidades em função dos recursos externos ao *sandbox*?

Com o surgimento da tecnologia SDN, notou-se uma oportunidade para desenvolver uma arquitetura para análise de *malwares* na qual a camada de rede é integrada ao processo de execução do código malicioso. Sendo possível gerenciar a sua execução e controlar o ambiente que circunda o *sandbox* segundo as necessidades demandadas pelo *malware*. Portanto esta tese propõe:

> ***Objetivo:*** *Desenvolver uma arquitetura especializada para analisar códigos maliciosos que permita controlar a execução do malware no sandbox, bem como os recursos que compõem o ambiente de análise.*

Para avaliar a arquitetura foi analisado um conjunto de *malwares* em dois cenários de avaliação. No primeiro cenário de avaliação as funcionalidades descritas pela solução proposta revelaram novos eventos comportamentais em 100% dos *malwares* analisados. Já, no segundo cenários de avaliação, foi analisado um conjunto de *malwares* projetados para dispositivos IoT (*Internet of Things*). Em consequência, foi possível bloquear ataques, monitorar a comunicação do *malware* com seu controlador de *botnet*, e manipular comandos de ataques. Mais do que utilizar uma nova abordagem para implementar uma arquitetura de análise de *malwares*, esta tese revela que o controle centralizado e a manipulação dos fluxos de rede trazem uma flexibilidade até então não disponível nas ferramentas de investigação.

2. Trabalhos Relacionados

A literatura apresenta inúmeras soluções relacionadas à análise dinâmica de *malware* na qual um arquivo suspeito é executado de modo a investigar suas funcionalidades [Kirat et al. 2014, Moser et al. 2007, Holz and Raynal 2005, Dinaburg et al. 2008]. A maioria das soluções, no entanto, foca no aprimoramento de sistemas onde o *malware* é executado, não considerando os demais recursos disponíveis no ambiente de análise. De fato, as ações desempenhadas por um *malware* são altamente dependentes de sua interação com outros sistemas. Sendo assim, uma ferramenta de análise de *malware* deve considerar o ambiente que circunda o *sandbox*. Mais do que permitir acesso à Internet, o ambiente que circunda o *sandbox* provê

recursos fundamentais para a execução do *malware*, tais como: resolução de nomes (DNS), endereçamento de rede, serviços de rede, e política de acesso à rede. De maneira geral, a camada de rede está desassociada do processo de análise de *malware* nas soluções existentes. Boa parte das soluções utilizam funcionalidades associadas com a contenção de acessos à rede, redirecionando de tráfego e filtro de pacotes [Kreibich et al. 2011, Oktavianto and Muhardianto 2013, Kruegel et al. 2006]. Já as demais funcionalidades de rede são negligenciadas pelas atuais ferramentas de análise de *malware*, em particular mecanismos relacionados à configuração dos elementos que compõe a arquitetura, filtro dinâmico de conteúdo e alterações de topologia [Graziano et al. 2012, Norman 2003, Pa et al. 2015]. MARS, por outro lado, vale-se da estrutura centralizada da tecnologia SDN para unificar o controle dos elementos de análise possibilitando com que a camada de rede seja integrada ao processo de análise.

3. Arquitetura MARS

A solução proposta especifica uma arquitetura modular baseada em eventos que integra a camada de rede ao processo de análise de *malware*. Cada evento é associado a uma determinada ação que irá interagir com o ambiente configurando e/ou coordenando a execução do *malware* no *sandbox*. Dessa forma, aspectos comportamentais resultantes da execução do *malware* geram eventos de configuração que atuarão no ambiente de investigação. A Figura 3 ilustra a estrutura da arquitetura MARS e a relação entre os elementos que a compõe. Os principais elementos funcionais da arquitetura são destacados, sendo: *sandbox* (❶) responsável por executar o arquivo suspeito e mapear suas funcionalidades no sistema; o controlador SDN (❷) que provê aspectos de comunicação de rede; e recursos de rede (❸) que constitui um repositório de serviços utilizado para compor o ambiente de análise.

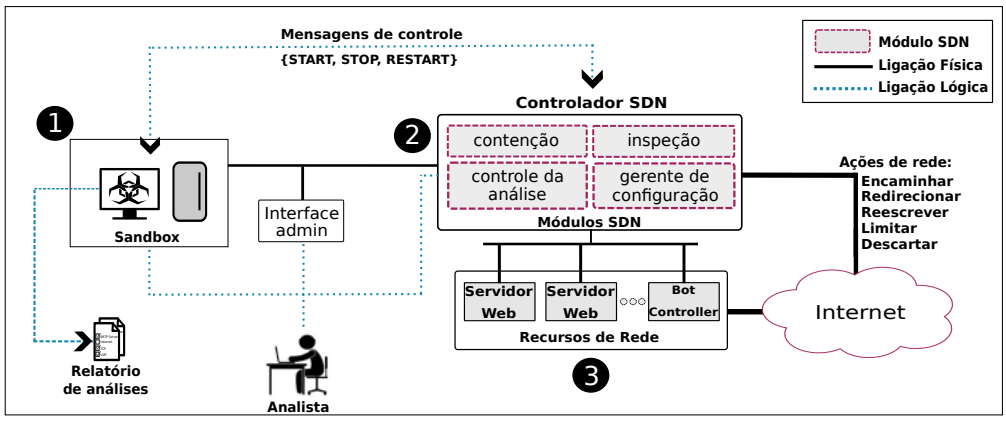

Figura 1. Arquitetura MARS.

O controlador SDN foi estendido com módulos específicos para o processo de análise de *malware*. Em particular, as funcionalidades implementadas possibilitam que o ambiente de análise seja reconfigurado dinamicamente em resposta a comportamentos de rede exibidos pelo *malware*. Através da manipulação de tráfego realizada pelo próprio controlador, é possível redirecionar fluxos de rede, reescrever pacotes, modificar a topologia e introduzir serviços vulneráveis no ambiente para coletar detalhes de ataques.

A arquitetura MARS não apenas introduz novas características no processo de análise de *malware*, mas também busca aprimorar as funcionalidades existentes nas soluções tradicionais. Em especial, por utilizar uma infraestrutura na qual

a camada de rede faz parte do processo de análise de *malwares*, torna-se possível desenvolver mecanismos de contenção de tráfego mais precisos considerando características de fluxos e do próprio conteúdo dos pacotes (protocolos não cifrados). Aliado a isso, a orquestração das funcionalidades de maneira centralizada traz flexibilidade ao processo de análise de *malware* permitindo que diferentes perfis de configuração sejam aplicados a um mesmo ambiente, sem a necessidade de intervenção ou configuração manual requeridas pelas soluções tradicionais. Como resultado, um mesmo *malware* pode ser investigado em diferentes cenários para que variações comportamentais decorrentes da configuração do cenário de avaliação sejam convenientemente mapeadas.

4. Avaliação Experimental

Para avaliar a solução proposta foram definidas duas avaliações experimentais: na primeira, foi analisado um conjunto heterogêneo de *malwares* compostos por códigos maliciosos modernos; na segunda, foram analisados *malwares* projetados para dispositivos IoT. Na sequência as duas avaliações experimentais são detalhadas.

4.1. *Malwares* Modernos

Esta avaliação experimental buscou identificar aspectos comportamentais dos *malwares* exibidos em decorrência da configuração utilizada no ambiente de análise. Como previamente comentado, sabe-se que o conjunto de recursos disponíveis pode fazer com que o *malware* analisado revele determinados comportamentos. Para tal, utilizou a flexibilidade provida por MARS para especificar diferentes configurações para a mesma infraestrutura de análise e mapear variações comportamentais de forma automatizada. Foram definidas três configurações para o mesmo ambiente de análise, detalhadas na tese, onde se utilizou diferentes políticas de contenção de tráfego e distintas regras de manipulação de tráfego de rede. Através da execução de um mesmo *malware* nas múltiplas configurações definidas, tornou-se possível identificar quais configurações podem ser mais adequadas para investigar um determinado *malware* e, em última análise, mapear os recursos demandados pelo *malware* que desencadeiam funcionalidades não observadas nas soluções tradicionais. Foram analisados 100 *malwares* de diferentes famílias na arquitetura MARS e, também, em uma arquitetura referência detalhada na tese. Utilizando um conjunto de parâmetros de avaliação obtidos na análise dos 100 *malwares* foi possível determinar variações comportamentais desencadeadas pela arquitetura MARS.

Parâmetros de avaliação	Referência	MARS	Incremento (%)
Endereços IP	7129	29277	410%
Duração	750	3268	435%
Conversação IP	1863	7617	408%
Dados Transmitidos	870	2264	260%
Portas TCP	281	796	283%
Portas UDP	536	572	106%
Assinaturas Comportamentais	609	630	103%

Tabela 1. Comparação dos parâmetros de avaliação obtidos na execução do *malware* no *sandbox*.

A Tabela 1 apresenta um comparativo dos parâmetros, onde *IP* ilustra todos os endereços IP observados no ambiente de análise, incluindo os endereços demandados pelo *malware* analisado; *conversação IP* corresponde ao tráfego IP realizado entre dois pares observado no ambiente de análise; *porta TCP/UDP* todas as portas TCP/UDP requisitadas pelo *malware*; *assinatura comportamental* são assinaturas fornecidas pelo *sandbox* e descrevem aspectos comportamentais do *malware* analisado; *dados transmitidos* quantifica o número de conversações nas quais foram transmitidos mais de 1000 bytes; *duração* quantifica o número de conversações que possuem duração superior a 2 segundos.

Fica evidente um incremento significativo em relação a todos os parâmetros dos *malwares* quando analisados na arquitetura MARS. Essa discussão é estendida na tese onde são apresentados casos específicos de *malwares* que adaptam o seu comportamento tendo em vista a política de contenção da rede. Da mesma forma, observou-se *malwares* que apenas exibem seu comportamento malicioso quando determinados recursos estão acessíveis a partir da estrutura de análise.

4.2. *Malware* IoT

Nesta avaliação experimental foram analisados *malwares* projetados para dispositivos IoT. Como objetivo, deseja-se demonstrar que as funcionalidades da arquitetura MARS facilitam a investigação de *malwares* mitigando o tráfego de ataque destinado a outros sistemas sem afetar a qualidade dos resultados. Para isso, a arquitetura foi customizada tendo em vista as características comportamentais das duas famílias de *malwares* IoT mais populares, i.e., Mirai e Bashlite [Antonakakis et al. 2017].

Por questão de espaço, esta seção apresenta o experimento realizado com *malwares* da família Bashlite que possuem características de uma *botnet*. Utilizando assinaturas de rede embutidas no controlador SDN foi possível dinamicamente adaptar o ambiente de análise bloqueando ataques realizados pelo *malware*, mas permitindo a comunicação com o C&C da *botnet*. A Figura 2 sumariza a taxa de pacotes por hora originadas na execução do *malware* Bashlite na arquitetura MARS.

Como ilustrado, a maioria dos pacotes são associados aos ataques de propagação, *i.e.*, pacotes destinados à porta 23/TCP (TELNET). O ataque de varredura inicia assim que o *bot* contata o C&C e persiste de forma praticamente constante durante todo o período da análise (24 horas). Observa-se uma pequena quantidade de pacotes associados à comunicação com o C&C da *botnet* que representam basicamente mensagens de *keep-alive* e comandos de ataques. Em particular, durante a análise foram detectadas cinco instruções de ataques encaminhadas para a máquina infectada. Esses comandos representam instruções de ataques de negação de serviços almejando o serviço HTTP (`GET` e `POST`) destinadas para cinco endereços IP distintos. Devido às características do ataque, o número de pacotes não é representativo no gráfico. Para ser efetivo, ataques DoS destinados aos serviços TCP requerem uma sessão estabelecida. No entanto, no sistema implementado, esses pacotes de ataques são subitamente bloqueados impedindo que a sessão TCP seja estabelecida.

Apesar de bloquear esses ataques, a arquitetura ainda assim pode mapear as características dos mesmos, tal como, alvos e aspectos dos pacotes. A tese também

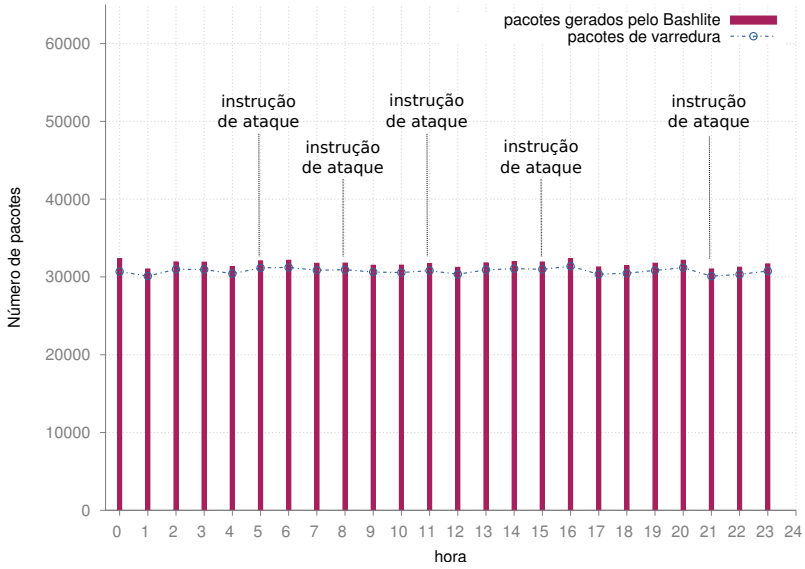

Figura 2. Bashlite: número de pacotes por hora gerados pelo *malware* analisado.

detalha como é possível manipular no tráfego de comunicação da *botnet*. São descritos, por exemplo, experimentos nos quais as instruções de ataques recebidas pelo *malware* são reescritas pelo controlador SDN de modo a inibir ataques.

5. Conclusões

Esta tese apresentou uma arquitetura para análise de *malware* denominada MARS *MAlwaRe Analysis Architecture based on SDN* – que emprega a tecnologia SDN para suprir as limitações das soluções atuais. MARS utiliza a camada de rede para integrar a operação do *sandbox* com os demais componentes do ambiente de análise. Com isso, o ambiente de análise pode assumir múltiplas configurações em termos de disponibilidade de serviços, topologia e política de acesso. Adicionalmente, o ambiente pode modificar suas características dinamicamente baseado nas ações do *malware* e, deste modo, moldar o ambiente conforme recursos demandados.

A análise dos resultados experimentais possibilitou as seguintes conclusões:

- a arquitetura proposta permitiu mapear alterações comportamentais dos *malwares* de maneira automatizada;
- a integração da camada de rede junto ao processo de análise demonstrou-se eficiente para controlar o tráfego gerado pelo *malware* e também para controlar os demais dispositivos presentes no ambiente de análise;
- os recursos demandados pelos *malwares* fazem que novas ações sejam mapeadas e, em última análise, que variações comportamentais sejam identificadas;
- o controle unificado do ambiente de análise permite um analista desenvolver ambiente complexos fazendo o uso de arquivos de configuração. Com isso, um *malware* pode ser analisado em diferentes cenários sem depender da intervenção humana para configurar manualmente os elementos, bem como definir a política de acesso em cada execução do *malware*;
- a avaliação experimental demonstrou que simples modificações no ambiente de análise podem desencadear novas ações comportamentais dos *malwares*;

- a política de acesso disponibilizada para o *malware* tende a afetar as suas ações comportamentais. Alguns *malwares* adaptam suas ações de modo a superar as limitações impostas pela política de acesso presente no ambiente;
- o encaminhamento de tráfego e manipulação de pacotes DNS foram funcionalidades efetivas para desencadear comportamentos inéditos na maioria dos *malwares* analisados.

Por fim, a tese faz uma discussão das principais limitações da arquitetura proposta abordando as especificidades do modelo SDN e também da dependência das atividades realizadas pelo *malware* executado no *sandbox*. Da mesma forma é apresentado as próximas direções desta pesquisa abordando elementos da arquitetura onde serão disponibilizados ambientes pré-configurados para compor processos customizados de análise de *malware* de maneira simplificada.

6. Produção Científica

Ao longo desta pesquisa de doutoramento, os seguintes artigos científicos relacionados aos tópicos analisados foram escritos e publicados:

1. *CERON, J. M.; MARGI, C. B.; GRANVILLE, L. Z. MARS: An SDN-based Malware Analysis Solution. In: IEEE. Computers and Communication (ISCC), 2016 IEEE Symposium on. Messina, Italy, 2016. p. 525-530. DOI: 10.1109/ISCC.2016.7543792.*
2. *CERON, J. M.; MARGI, C. B.; GRANVILLE, L. Z. MARS: From Traffic Containment to Network Reconfiguration in Malware-Analysis Systems. In Computer Networks. ISSN: 1389-1286, 2017. DOI: 10.1016/j.comnet.2017.10.003.*

Adicionalmente, em resposta ao Edital BR-US 2016 - *CyberSec/Malware detection*, foi desenvolvido um projeto em associação com a Professora Jelena Mirkovic da USC (*University of Southern California*) para participação no referido edital que busca incentivar pesquisas na área de segurança entre Brasil e Estados Unidos. Para tal, participei do desenvolvimento e escrita do projeto, o qual propõe uma infraestrutura de análise de *malware* que incorpora as funcionalidades descritas por MARS. Mesmo sem ter sido contemplado, o projeto passou para etapa final da seleção.

Apesar do término do doutoramento, a pesquisa apresentada por esta tese continua em andamento. Os autores buscam estender a arquitetura provendo um conjunto de cenários de avaliação previamente configurados para investigar *malwares* avançados projetados para órgãos governamentais. Em paralelo, a arquitetura MARS está sendo utilizada como base para desenvolver uma solução especializada em *malwares* IoT (subproduto), tendo em vista suas particularidades de acesso. Por fim, acredita-se que a arquitetura MARS possui um grande potencial para revelar *malwares* em diferentes contextos, incluindo códigos maliciosos projetados para dispositivos SCADA (*Supervisory Control and Data Acquisition*) e demais sistemas ICS (*Industrial Control Systems*) que serão investigados em trabalhos futuros.

Referências

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever,

C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., and Zhou, Y. (2017). Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1093–1110, Vancouver, BC. USENIX Association.

Dinaburg, A., Royal, P., Sharif, M., and Lee, W. (2008). Ether: malware analysis via hardware virtualization extensions. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 51–62. ACM.

Graziano, M., Leita, C., and Balzarotti, D. (2012). Towards network containment in malware analysis systems. In *28th Annual Computer Security Applications Conference, ACSAC 2012, Orlando, FL, USA, 3-7 December 2012*, pages 339–348. ACM.

Holz, T. and Raynal, F. (2005). Detecting honeypots and other suspicious environments. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 29–36, West Point, New York, USA. IEEE.

Kirat, D., Vigna, G., and Kruegel, C. (2014). Barecloud: bare-metal analysis-based evasive malware detection. In *Proceedings of the 23rd USENIX conference on Security Symposium (SEC'14). USENIX Association, Berkeley, CA, USA*, pages 287–301. USENIX.

Kreibich, C., Weaver, N., Kanich, C., Cui, W., and Paxson, V. (2011). Gq: Practical containment for measuring modern malware systems. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 397–412. ACM, ACM.

Kruegel, C., Kirda, E., and Bayer, U. (2006). TTanalyze: A tool for analyzing malware. In *Proceedings of the 15th European Institute for Computer Antivirus Research (EICAR 2006) Annual Conference*, volume 4, Vienna, Austria. EICAR.

Laurianne, M. (2004). Bot software spreads, causes new worries.

McLaughlin, L. (2004). Bot software spreads, causes new worries. *Distributed Systems Online, IEEE*, 5(6):1.

Moser, A., Kruegel, C., and Kirda, E. (2007). Exploring multiple execution paths for malware analysis. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 231–245. IEEE, IEEE Computer Society.

Norman (2003). Norman sandbox technical repor. Technical report, Technical report, New York, USA.

Oktavianto, D. and Muhardianto, I. (2013). *Cuckoo Malware Analysis*. Packt Publishing Ltd, New York, USA.

Pa, Y. M. P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., and Rossow, C. (2015). Iotpot: Analysing the rise of iot compromises. *EMU*, 9:1.

Virvilis, N. and Gritzalis, D. (2013). The big four-what we did wrong in advanced persistent threat detection? In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pages 248–254, Regensburg, Germany, September. IEEE, IEEE Computer Society.