SuperviZ

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Intent-Based Attack Mitigation through Opportunistic Synchronization of Microservices

SOTERN TEAM - IRISA, IMT ATLANTIQUE

- Do Duc Anh NGUYEN (Presenter) - do-duc-anh.nguyen@imt-atlantique.fr
- Pierre ALAIN - pierre.alain@irisa.fr
- Fabien AUTREL - fabien.autrel@imt-atlantique.fr
- Ahmed BOUABDALLAH - ahmed.bouabdallah@imt-atlantique.fr
- Jérôme FRANÇOIS - jerome.francois@uni.lu
- Guillaume DOYEN - guillaume.doyen@imt-atlantique.fr

IEEE NetSoft 2024, PhD Symposium, Saint-Louis USA, June 24-28, 2024

Outline

Outline

Introduction

11/2019

Graduated
Ho Chi Minh City University of Technology
Computer Science

VNG Corporation (1y)
Backend Engineer

9/2020

University of Limoges
Master's student in Cryptology and CyberSecurity

3/2022

IMT Atlantique
Master internship - ***PhD***

1/2023

Outline

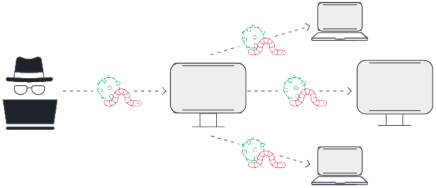
Context: Requirements of Security Management

Implementing security policies is challenging due to

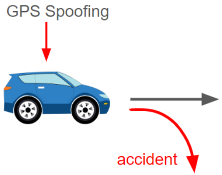
- Complexity of current and future IT systems
- Requirements of quick reaction to cyberattacks

Multiple threats

Malware propagation



Attack against self-driving vehicles

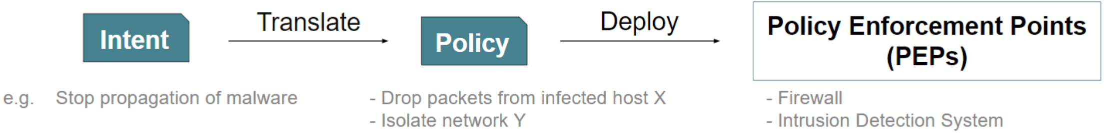


→ Reducing complexity and reaction time against attacks are important to make current and future IT systems secured and robusts

Intent-Based Networking (IBN)

Intent-Based Networking (IBN)

IBN allows the user to specify intents, which stands for the desired outcome, without the need for detailed operations to automate configuration orchestration

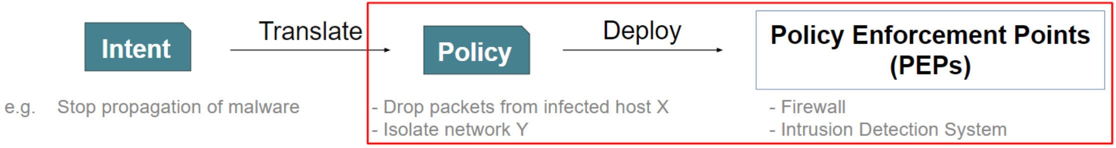


$\Rightarrow$ Avoid time-consuming and error-prone tasks, facilitates the expression of a security policy

Intent-Based Networking (IBN)

Intent-Based Networking (IBN)

IBN allows the user to specify intents, which stands for the desired outcome, without the need for detailed operations to automate configuration orchestration

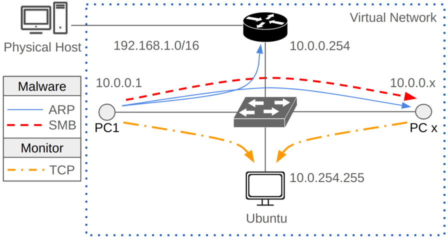


$\Rightarrow$ Avoid time-consuming and error-prone tasks, facilitates the expression of a security policy

Problematic

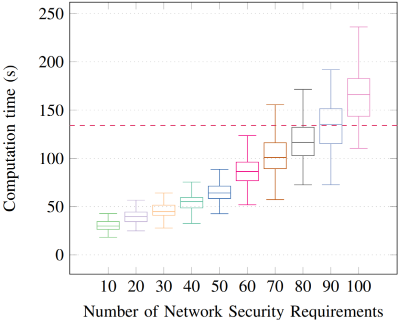
IBN systems may experience degraded performance and limited scalability

Research question: How can we leverage IBN systems to react to rapid security attacks, such as the fast propagation of malware?



In a 50-node LAN, approximately 60% of infections are processed $\leq$ 100 s [1]

><



VEREFOO [2]

[1] Do Duc Anh Nguyen et al. "How Fast does Malware Leveraging EternalBlue Propagate? The case of WannaCry and NotPetya". In: SecSoft Workshop. 2024

[2] Daniele Bringhenti et al. "Automated Firewall Configuration in Virtual Networks". In: IEEE Transactions on Dependable and Secure Computing 20.2 (2023)

Outline

State of the Art: IBN frameworks

Existing open source frameworks

Language Level	Intent		Policy
Framework	Lumi [3]	I2NSF [4]	Verefoo [5]
Language	Natural Language	YANG policy	Firewall rules and network topologies
Automation	Network configuration	Security function deployment and configuration	Compute optimal solution and configure firewalls

Unscalable due to a multitude of complex computations

Outline

Scalability Assessment of IBN Systems for Security: the case of I2NSF

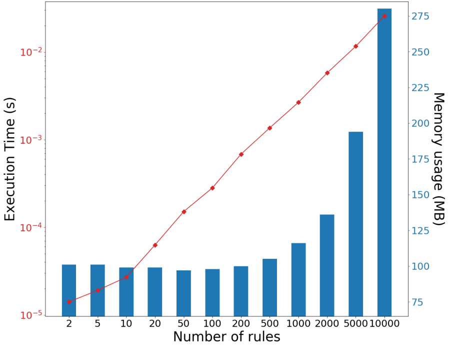
What is the minimum time required for a robust security controller to compute and deploy a novel security configuration?

I2SNF, a standard framework proposed by the IETF, is selected for study

→ Propose a conflict detection and resolution approach for the I2NSF framework [6]

- Robust implementation for the security controller
- The result has polynomial complexity

→ The system is not scalable

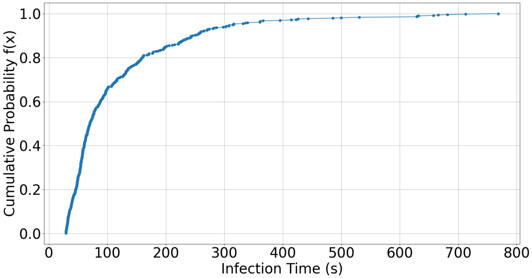


Fast Propagation of Malware

What is the fastest malware propagation time one might encounter?
WannaCry and NotPetya are selected to study fast-spreading strategies

EternalBlue exploit [7]

Allows attackers to execute a remote code on the infected hosts by sending specially crafted Server Message Block version 1 (SMBv1) packets to unpatched Windows systems



Experiments

- 50 Windows 7
- Star topology

WannaCry: Approximately 20% of infections $\leq$ 50 seconds

$\rightarrow$ Centralized approaches are too slow to promptly react against malware propagation

Proposal: Opportunistic Decentralized Mitigation

Against fast malware propagation, we consider a decentralized and autonomous reaction of PEPs

Microservices [8]

Microservices are software-based functions that are decomposed from a large, complex application into independent services (e.g., Unikernels boots in 200 ms)

→ Deploy microservices as PEPs to enable

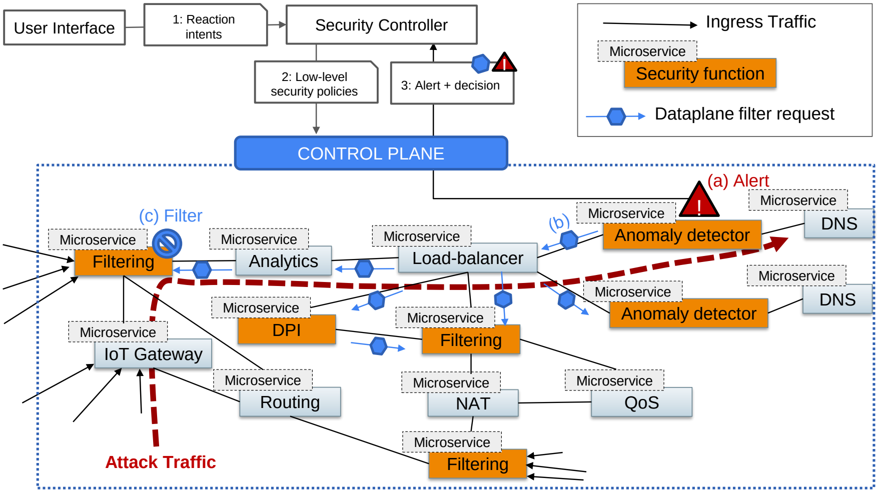
- Scalability
- Flexibility

A synchronization mechanism is required to enable autonomous reactions

Opportunistic synchronization

Leveraging existing data packets as an opportunity to share reaction information and synchronize their responses (e.g., using data plane programmability)

Global Architecture



Challenge and Research Question

Challenges for the opportunistic decentralized mitigation approach

- Need to handle heterogeneous capabilities of microservices
- Quick synchronization are required

Research questions

- How can we autonomously deploy IDS as microservices to ensure a complete view of a network activity?
- How can microservices perform opportunistic synchronization in response to attacks?

Methodology

Vertex cover problem: Allocate a minimum number of IDSs, but to be able to monitor all traffic paths

Lots of algorithms are proposed, such as [9] but they require knowledge of network topology

$\rightarrow$ We first consider a distributed algorithm proposed [10] to solve the problem

- Consider the local view of nodes (no initial information is required)
- Provide a near optimal solution (1% difference in their evaluation)

---

[9] Selman Yakut et al. "A new robust approach to solve minimum vertex cover problem: Malatya vertex-cover algorithm". In: The Journal of Supercomputing. 2023

[10] Vahid Khalilpour Akram and Onur Ugurlu. "A localized distributed algorithm for vertex cover problem". In: Journal of Computational Science 58. 2022

Methodology

Synchronization: Reaction decisions are embedded in packets on the data plane
Different strategies can be leveraged to deliver the embedded packets

- Sent backward along the attack path
- Broadcast

$\rightarrow$ Directly impact the reaction time and the induced overhead
$\Rightarrow$ Faster if existing packets can be leveraged
The state of the art in collaborative methodologies are considered [11]

- Embedding of data plane information used for IP traceback [12] may be considered

[11] N. Bougueroua et al. "A survey on multi-agent based collaborative intrusion detection systems". In: Journal of Artificial Intelligence and Soft Computing Research. 2021

[12] R. Wang et al. "In-band network telemetry based fine-grained traceability against IP address spooling attack". In: ACM ICEA. 2021

Outline

Conclusion

The idea of a microservice-based IBN system is proposed for security management

- Microservices are deployed based on intents
- Opportunistic approach for fast synchronization and autonomous reaction

Future work: Design an initial opportunistic mechanism, solving two main problems

- Solving the vertex cover problem to deploy IDSs efficiently and facilitate investigation for reaction
- Selection of suitable strategies that can be leveraged to deliver attack information quickly

SuperviZ

Acknowledgement: This work is funded by the French 2030 Cybersecurity - Superviz project

References I

[1] Do Duc Anh Nguyen et al. „How Fast does Malware Leveraging EternalBlue Propagate? The case of WannaCry and NotPetya“. In: SecSoft Workshop. 2024.

[2] Daniele Bringhenti et al. „Automated Firewall Configuration in Virtual Networks“. In: IEEE Transactions on Dependable and Secure Computing 20.2 (2023), pp. 1559–1576. DOI: 10.1109/TDSC.2022.3160293.

[3] Arthur S Jacobs et al. „Deploying natural language intents with lumi“. In: Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos. 2019, pp. 82–84.

[4] Jinyong Kim et al. „IBCS: Intent-Based Cloud Services for Security Applications“. In: IEEE Communications Magazine 58.4 (2020), pp. 45–51. DOI: 10.1109/MCOM.001.1900476.

References II

[5] Daniele Bringhenti et al. „Automatic, verifiable and optimized policy-based security enforcement for SDN-aware IoT networks“. In: Computer Networks 213 (2022), p. 109123.

[6] A. Nguyen et al. „A Robust Approach for the Detection and Prevention of Conflicts in I2NSF Security Policies“. In: IEEE/IFIP NOMS 2023 2023. IEEE. 2023.

[7] Zian Liu et al. „Working mechanism of eternalblue and its application in ransomworm“. In: International Symposium on Cyberspace Safety and Security. Springer. 2022, pp. 178–191.

[8] Irakli Nadareishvili et al. Microservice architecture: aligning principles, practices, and culture. ” O'Reilly Media, Inc.”, 2016.

References III

[9] Selman Yakut, Furkan Öztemiz, and Ali Karci. „A new robust approach to solve minimum vertex cover problem: Malatya vertex-cover algorithm“. In: The Journal of Supercomputing 79.17 (2023), pp. 19746–19769.

[10] Vahid Khalilpour Akram and Onur Ugurlu. „A localized distributed algorithm for vertex cover problem“. In: Journal of Computational Science 58 (2022), p. 101518.

[11] Nassima Bougueroua et al. „A survey on multi-agent based collaborative intrusion detection systems“. In: Journal of Artificial Intelligence and Soft Computing Research 11.2 (2021), pp. 111–142.

[12] R. Wang et al. „In-band network telemetry based fine-grained traceability against IP address spooling attack“. In: ACM ICEA. 2021, pp. 229–233.

Detection

Input: $R0$, $R1$
Output: True if $R0$ conflicts with $R1$ and False otherwise

```
if is_different_action(R0, R1) then
    if not exist_nonoverlapped_ap(R0, R1) then
        return True
    end if
end if
return False
```

Algorithm 1: *detect*

Function *detect* follows ABAC proposal [**liu2021novel**]

- *is_different_action*$(R0, R1)$: compare actions
- *exist_nonoverlapped_ap*$(R0, R1)$: compare all attributes

$\rightarrow$ The complexity: $O(A)$ where
A: the attribute number defined in the DM

$\Rightarrow$ The real-time conflict checker uses *detect* to check a new rule against the installed rule set

[**liu2021novel**] A novel conflict detection method for ABAC security policies, Journal of Industrial Information Integration, 2021

Deployment of an I2NSF testbed

A ground architecture to allow the deployment of any subsequent contribution

- Selection of a testbed implemented and presented at IETF Hackathon (#104 to #113)
- Installation and setup of an underlying Devstack distribution
- Reproduction of the standard scenario considered in [4]

Several bugs and issues which made the testbed setup and standard test scenario difficult to implement

- Installation errors in inconsistent version between Devstack plugins
- NSF database of Security Controller is inconsistent in capabilities compared to their instruction
- NSFs do not send IP address to DMS after being initiated
- Service chaining failed because NSFs do not process the incoming packet

[4] IBCS: Intent-Based Cloud Services for Security Applications, IEEE Communications Magazine, 2020