Eindhoven University of Technology

MASTER

Cyber-Attack Containment through Actionable Awareness

van Leeuwen, Roy M.J.

Award date:
2022

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Cyber-Attack Containment through Actionable Awareness

Roy van Leeuwen
Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, Netherlands
royvanleeuwen@live.nl

***Abstract*—Cyberattacks are becoming increasingly automated. Modern cyber attacks can take mere minutes to hours to achieve their goal, while the response to these incidents often take hours to days. This shows a need to speed up the incident response process. A promising way to achieve this is by automating (part of) the incident containment process. We identify the biggest challenge bringing the academic research to the real world are the limitation on when which containment technique can be used. By enabling to apply automated containment in practise we reduce the impact of cyber attack automation, as the response will then be able to keep up with the speed of attackers.**

In this paper we propose to split containment actions into containment techniques and the actuators that execute the actions. Containment techniques are collected from research-oriented sources, established best practises and interviewing professionals. This gives a reasonably complete overview of the technique that are employed. We had experts generate a mapping between actuators and the identified technique to generate actionable containment actions. By including which actuator executes the action we can ensure that the action can be executed in a specific context. The proposed method is showcased by generating a matrix of the actuator and containment technique mapping. A demonstration is given of how the matrix can be used in practical cases to select all the containment measures available in the given network.

***Index Terms*—Cyber security, Cyberattack countermeasures, Intrusion containment, Intrusion response, Security Automation, Option Awareness**

1. Introduction

As more and more of our lives shifts into the digital domain, cybercrime is becoming an increasingly dangerous threat to us all, especially to the entities that operate and use the information technology (IT) infrastructure. When an incident or cyber-attack occurs, an incident response team is generally tasked with creating and implementing a response plan to stop the incident and resume normal operations. There are four general phases for incident response plans, namely ”preparation phase”, ”detection and analysis phase”, ”containment, eradication and recovery phase” and ”post-incident activities” [26]. The ”preparation phase” is a continuous process that takes place before an incident occurs. The ”detection and analysis phase” is crucial, as one cannot react if one does not know that something is wrong. We look at some of the challenges in containing a detected yet uncompleted cyber-attack. To contain a cyberattack means to remove the adversary’s control or capability within the victim’s network, the goal is to seize the initiative from the adversary. [26] To accomplish this, the incident response team draws up a containment plan, consisting of containment actions and when and where these actions should be taken to contain the adversary threat. In theory, this is a simple task, but in practice it becomes exceedingly complex. A cyber attack can take many forms and the information available is often incomplete. There is also a time pressure; the adversary will not wait for a response before continuing the attack. On top of these ever-present challenges, the IT landscape is evolving and changing rapidly, and the defenders infrastructure and capabilities can change just as rapidly, making it much more difficult to maintain a clear overview of the available containment options

Cyberattacks are becoming increasingly professionalized and automated. Modern cyberattacks can take anywhere from a few minutes to hours from start to finish, while responders to these incidents often need hours to days to come up with and execute an appropriate containment plan. This shows that the incident response process needs to become faster to be able to stop future cyberattacks before they are concluded. An additional challenge for the cyber security industry as a whole is the shortage of qualified and skilled personnel to respond quickly and accurately to incidents. A promising approach to solving both of these challenges is to automate the containment phase of incident response. By automating the containment of a cyberattack, the immediate risk of the incident is greatly reduced and the incident response team is given more time to develop a plan for eradication of the threat and recovery from the incident.

In recent decades, numerous proposals have been made for containment automation systems. However, we believe that these proposals are not sufficient to provide a solution that can be applied in the real world. The evaluated containment automation systems lack applicability as they offer no relationship between the proposed incident response and where or how these actions should be executed. The granularity of the network models and the response is far too coarse to automate in a real world network. There is a gap between the academic world that performs cost- and benefit-aware analyses of various containment actions and the real world where constraints exist in terms of which containment actions can actually be performed in the given network. This will lead to automated containment systems providing containment plans without ensuring they can be executed in the real world environment they are modelling.

In this paper, we propose a way to bridge the gap between the more abstract academic state of the art and the real world. We propose to do this by breaking down containment actions into the containment techniques they use, and the actuators responsible for executing the containment actions. By making this division, we allow reasoning about the containment capabilities of the individual techniques, while being able to ensure actionable containment plans by only allowing actions for which a capable actuator is in place. Mapping containment techniques to containment actuators that can apply those techniques also offers a structured way of defining containment actions.

We showcase the proposed method by creating a Actuator-Technique matrix using the proposed framework. We examine two real-world cyber-attacks to see which containment actions could be performed using the proposed framework. This will show both how the proposed framework is used to create a mapping between techniques and actuators, and how this mapping is used in a real world setting.

This manuscript proceeds as follows: Section 2 discusses related work; Section 3 the methodology; Section 4 the results; Section 5 showcases the framework application; Section 6 discussion; Section 7 the conclusion.

2. Related work

Cyber-attack containment. Proposals for automated response systems in the academic space are plentiful, dating back from at least 1996. A characteristic of earlier proposals [7] [10] is the idea of defining some response action to be executed when a predefined scenario or anomaly is detected. These response actions can be defined by the system administrators [7]. Allowing a system administrator to define action gives greater control, however it requires a great effort to generate a substantial set of containment actions. Next to the effort required to set the system up the administrator will need to update the actions when any change is made in the network. This greatly reduces the flexibility and scalability of such an automated containment system. Other work proposed the response actions to be defined in an exchangeable format and shared between trusted parties, after which the results of a executed response can again be shared withing the group of trusted parties [19]. Common modeling languages to formally define observation(attacks) with desired responses(containment actions) have been proposed [10], but these proposed exchangeable languages are rarely seen reused in other research or proposals. Formally defining cyberattacks and the desired responses to them can rarely match the real world in terms of accuracy and detail without becoming far too complex to remain workable. Another drawback of defining responses and sharing these between different containment automation systems is that no two networks are identical, resulting in situations where a containment plan that worked as desired in one network, might not lead to desired results in other networks.

In order to contain a cyberattack one or multiple containment actions need to be taken. The MITRE corporation [13], known in the industry for the MITRE ATT&CK knowledge matrix [16], recently proposed the D3FEND knowledge Matrix [14] and the shield knowledge matrix [15]. These matrices both provide actionable measures that cybersecurity professionals can employ in modern IT systems. While the matrices do not focus on containment specifically, some of the proposed measure are relevant for incident containment. In addition to this framework many academic papers have mentioned and introduced some different containment actions in their proposals to automate incident response. Some provide explicit lists of containment actions covered [3] [11], others give some examples of possible containment actions in text [12] [4], however most papers that propose an automated containment system do not explicitly list any containment actions or techniques.

Incident modeling. A downside of defining containment responses by hand before an incident occurs is that a system is not able to respond to unforeseen situations. A survey on reaction frameworks [21] suggests that a more generalised approach would resolve this limitation. [21] proposes an approach to generalise the selection of containment actions to generate a model of the possible attack states and responses. Using a attack graph model of the cyberattack allow to find a minimal set of attack steps [9]. A more recent approach is to define attack steps based on those in the MITRE attack framework, and add pre- and post- conditions describing capabilities required or provided by an attack step [2]. Similarly, attack tree approaches are proposed to show the different requirements of attacks before a next steps can be executed [20]. These models can also take the form of an attack graph, which shows different pathways an attack could follow through a network [22]. The attack models can include sets of actions which are available on certain nodes in the model graph. This logically leads to containment actions from detected attacks. This however still has the same downside of a state explosion for more complex networks as concluded in the survey discussing reaction frameworks [21]. A limitation common to all proposals applying modeling techniques is that all included containment actions are assumed to be available at all times. There is no consideration for what containment actions might be feasible to execute or whether actuators to execute the action are in place. For some works this is entirely reasonable, for example when the action is performed by a member of staff and the only considered actions are shutting down a system, stopping a service or traffic filtering. However when wanting to automate the response execution some automated actuator capable of executing the evaluated action needs to be present. All actuators can not realistically be assumed present on any machine in any network. To solve this challenge we propose a way to determine which containment actions are available on a network location based on the actuators that are present.

Containment plans evaluation. Attack modeling in itself does not include proposals on how identified attack steps could be contained or the cost of executing specific containment actions. A way to select more appropriate responses cost-benefit analysis based on some set of metrics [5]. By evaluating and attaching scores to their different quality attribute, containment actions can be compared and the optimal response can be chosen. Which evaluation would result in the optimal reaction can differ from minimizing total cost [6] to choosing the (set of) containment actions that generate the best return-on-investment [20]. One of the returning challenges found in multiple approaches is increasing computational and time complexity when introducing more containment actions [21]. In a system where combinations of containment actions are evaluated any additional actions included in the evaluation have a non-polynomial increase in computational complexity [3, section 6.C p.2552]. As such, limiting the set of available containment actions would greatly reduce the computational complexity of the evaluation.

2.1. Research Gap and problem statement

A plenitude of approaches, ranging from manual one-to-one mappings of detection and response to cost sensitive models that calculate optimal responses have been proposed. These proposals however lack a clear and exhaustive analysis of available containment actions. A Framework to relate the academic world of abstract models to the real world which includes limitations does not yet exist. There is currently no structured way to express the relation between techniques, the actuators and when or if the actions can be employed during a detected cyber attack. This prevents any automation proposal from being directly actionable in the real world.

2.1.1. Scoping. In this paper we focus on the containment actions. Specifically we aim to find out which containment actions exist and propose a way to determine which actions are available in the network. We introduce a framework that allows mapping the containment techniques to actuators. This mapping results in containment actions that we can be assured to be available when the corresponding actuators is. We assume that cyberattacks can automatically be detected and analysed to a degree that the presence of and (some of) the adversaries capabilities are known during a cyberattack. We assume that there is full access and knowledge of the network and there is enough time to prepare and analyse the capability of the network. We do not offer a way to select which containment action would be best suited to counter a detected cyberattack or how to tailor a response to a given attacker.

2.2. Problem statement

To automate the selection of containment actions, it is necessary to know which actions exist. Currently, there is no method available to exhaustively construct the containment actions that can be applied in practice. As a result, there are only incomplete lists of containment actions and a complete overview is missing. Furthermore, there is currently no relationship between actuators and the containment actions they perform. This leads to complications when implementing automation proposals for containment, as there is no way to automatically determine which actions can be performed. A framework for relating real-world actuators to abstract containment techniques is needed to bridge the gap between academia and the real world where automation is so desperately needed.

Main research question: **What are possible containment actions and how to determine their availability in a given network?** From this main research question we derive the following research questions:

RQ1: What are the containment technique that can be employed using a containment action?
RQ2: What are the containment actuators that can execute containment actions in a network?
RQ3: How to generate a mapping between containment actuators and containment techniques?

3. Methodology

In order to answer our research questions we devise the following methodology. Figure 1 provides a bird's eye view of the overall approach and its link to the research questions. To create a framework mapping containment actions, techniques, and actuators, we first define a set of techniques and a set of actuators. We use these to build an Actuator-Technique matrix, where each intersection forms the basis of a containment action. Containment actions that exist can then be filled in in this matrix. This Actuator-Technique matrix can then be used to find available containment action given a set of or actuators that are present in the network. This results in a smaller set of actions, as well as ensure that all identified containment actions can be executed in the given context.

3.1. RQ1: What are the containment techniques that can be employed using a containment action?

To capture both research-oriented as well as well-established practices we review the academic literature, the MITRE frameworks and consult domain experts. Only technique or actions that can be used in or as a containment action are included. As we only want to include techniques into the containment technique matrix we need to extract the used technique from any identified containment actions.

3.1.1. Academic literature. During the literature research phase related academic papers have been identified. These are papers that discuss a topic related to automating incident containment, including all papers mentioned in the related works section. We searched on google scholar with the following set of key words: *cyber security; cyberattack countermeasures; intrusion response systems; intrusion prevention and response systems; optimal countermeasure strategy; dynamic reaction selection; containment; intrusion response; automation; automatic mitigation.* In addition to searching with keywords, we also examined the sources of relevant papers. A paper was relevant if it was about incident containment. This was intentionally broad, because both papers on containment planning, and papers on response modeling can cover containment actions.

The academic papers returned by our queries are selected first by reading title and abstract and, if fit, are selected. A paper is deemed relevant if it is about incident

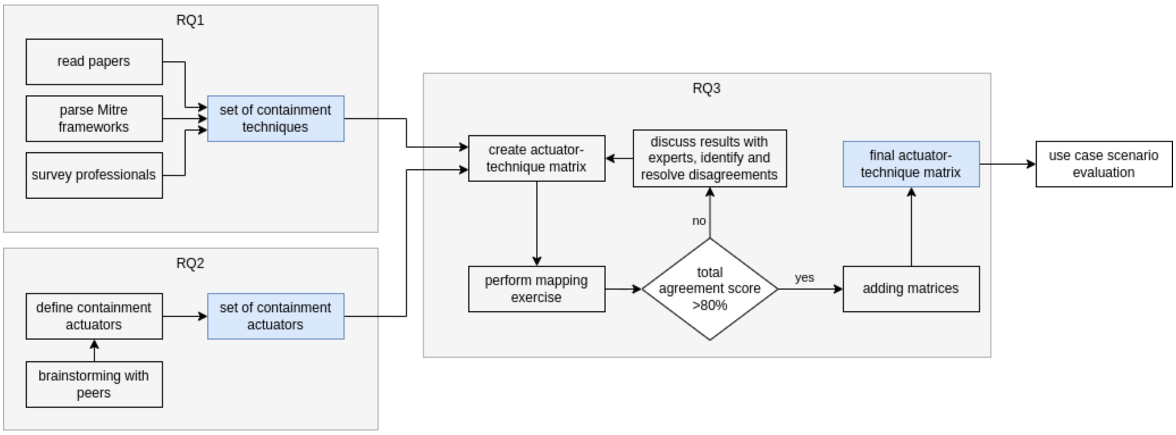

RQ1
read papers
parse Mitre frameworks
survey professionals
set of containment techniques
RQ2
define containment actuators
brainstorming with peers
set of containment actuators
RQ3
create actuator-technique matrix
discuss results with experts, identify and resolve disagreements
perform mapping exercise
total agreement score >80%
no
yes
adding matrices
final actuator-technique matrix
use case scenario evaluation

Figure 1. Visual representation of method. In the grey rectangles represent steps to answer the RQ in which it is placed. The blue rectangles answer the related RQ.

containment. This was intentionally broad, because papers on both containment planning, and papers on response modeling can cover containment actions. Selected papers are read in full, and identified containment techniques or actions are extracted and included in our selection. To keep as wide an inclusion criteria as possible, we do not impose additional requirements on how many or in which specific publications a techniques must be mentioned to be selected. Identified techniques are added to the containment technique matrix. We also check if found containment actions use one of the existing techniques, if not we define a new technique used by the containment action.

3.1.2. Existing frameworks. The MITRE organisation offers two frameworks dedicated to security these are called `MITRE D3FEND` [14] and `MITRE SHI3LD` [15]. Additionally the `MITRE ATT&CK` framework offers mitigations [16] to attack steps defined in the framework. As those frameworks serve a wider purpose than containment alone, it is necessary to go through the frameworks and manually selecting only techniques and action relating to containment. Identified techniques are added to the containment technique matrix. We also check if found containment actions use one of the existing techniques, if not we define a new technique used by the containment action.

3.1.3. Domain experts. In order to fill possible gaps left behind by the literature review and MITRE frameworks, we include domain experts in our source collection. To collect techniques employed or known by domain experts we conduct an exploratory survey. The contacted experts are security professionals active in the domain and within the network of researchers in the TNO organisation. 25 security experts were contacted and asked to participate in a short survey. All experts were selected on the basis of current or previous cooperation with the supervisors within TNO. A reminder was sent to all experts one week after the original invitation.

The exploratory survey consists of seven scenarios to provide context to help the experts provide answers, each scenario regarding the same network model. To achieve a good coverage of the containment actions conducted by domain experts each incident portrays a different scenarios. After consultation with matter experts at TNO we came up with a set of scenarios including: abused stolen credentials, detected command and control communication, detected customer data exfiltration and detected malware on machine. For each scenario a network diagram is given and the affected machine is highlighted. The expert is asked to order a set of containment actions. This listed containment action are the following: add network filtering, disconnect machine from network, change compromised credentials, disable service(if applicable), shutdown machine, break TCP connection, add a trigger or add targeted monitoring. These options where given as these action where already identified in academics and the best practise sources. They are meant to give the experts an starting point to build their containment plan such that they did not have to start from scratch. We ask the expert is there is an action that would be more appropriate in the given situation. This gives room for the experts to provide containment action and techniques which were no identified in other sources. We also ask the expert to explain the made assumptions and reasoning for their choices. After the 7 scenarios we ask if the experts knows any more interesting scenarios we should take into account for the research. We also ask if the experts knows more containment actions that were not mentioned in the survey or their answers. Any containment technique mentioned or described by an expert is included.

3.2. RQ2: What are the containment actuators that can execute containment actions in a network?

We define an actuator as a machine, a software or a human operator which can execute some containment action(s). The set of all actuators is, in practice, very large, as every brand of physical device, computer program or person can have different properties, functions or skillsets. In order to circumvent this challenge we propose a more generalised set of containment actuators. Rather than listing all, say, Antivirus software programs, we adopt general concepts describe the category in which the actuator sits; for example, all Antivirus software programs can be considered ”Host-based security agent”, regardless of their specific functionalities. The process of collecting these generic actuators consisted of multiple rounds of brainstorming sessions with different researchers (including senior researchers at the involved organizations, and peers). The goal of those sessions was to collect as complete a list of actuator categories as possible. Whereas we cannot assure a complete coverage, we engaged with experts in different fields (system security, networking, network security ...) to maximize the scope of the collection. External sources are also used by performing Internet searchers using terms such as ”network security systems” and ”security software agents”. Any actuator we came up with or found such as a antivirus software program is then abstracted to the related level (e.g., ”Host-based security agent”).

3.3. RQ3: How to generate a mapping between containment actuators and containment techniques?

We create the Actuator-Technique matrix by intersecting the collected set of actuators and techniques. We first conduct a pilot mapping exercise with three domain experts, these where two security researchers working at TNO and a Data center/networking Engineer working at Google. To gain insight in how much the experts agree we calculate agreement scores between each experts mapping. An agreement score is defined as the percentage of intersection where all experts agree on applicability($intersections_{agreed}$), over the total number of actuator-technique intersections($intersections_{total}$).

$$\frac{intersections_{agreed}}{intersections_{total}} * 100 \tag{1}$$

Agreement scores can be calculated over the entire Actuator-Technique matrices or parts thereof. It can provide interesting insights to calculate agreement scores across individual techniques or actuators. To do this, count the agreed intersections within an actuator column and divide this over the total number of intersections in this same column. The same process can be done for techniques by adding up the intersections in corresponding rows.

We discuss responses with the domain experts to get a qualitative understanding of their interpretations of the actuator and technique definitions. In cases where the interpretation differs between experts we refined the definition of the techniques/actuators to the point were the experts sufficiently agreed in interpretation. The experts perform the mapping exercise again, this time using the refined actuator and technique definitions. We calculate the agreement scores between the mapping again. In line with common practice, we consider an agreement score of at least 70% as acceptable and an agreement score above 80% as very good or excellent [24].

We ask each expert to fill in the Actuator-Technique matrix indicating which technique each actuator can perform. For the consolidation of the experts matrices we added them together by counting for each intersection of actuator and technique how many experts indicated applicability. The final Actuator-Technique matrix is then constructed by placing a ● at each intersection where all exerts indicate applicability, a ◐ where two indicate applicability, a ○ where one indicates applicability and it is left empty where the experts all agree it is no applicable.

3.4. Evaluation strategy

To showcase the proposed method in action, we define two case studies based on real world cyberattacks. To do this, we select cases that are well documented, and successful from the perspective from the attacker. To showcase the method over diverse scenarios, we choose cases different in attackers goal (nation-state attack vs ransomware attack) as well as with different network designs (closed/air-gaped vs open design). Having the use cases differ from each other will show how the framework can be applied to different types of attacks across different networks.

Based on these criteria we choose the Stuxnet [25] and the Maastricht University cyberattack [1]. Both of these are well documented by widely recognised information security companies. The two attacks also differ in attacker goals. The goal of the Stuxnet attack was to cripple or even destroy infrastructure, where the Maastricht University hack was a ransomware attack aiming to extort ransom from the afflicted university. The networks of the victims were also very different at the time of the cyberattack. The Stuxnet victim is a nuclear enrichment facility with multiple layers of security in a design pattern often seen in the operational technology sector. Where the Maastricht University is an academic network where it can be assumed that the network design was flat and open at the time of the attack.

As part of each case study we start by defining a network model of the victim/defender. In order to ensure realistic assumptions on these networks we have consulted with domain experts working at Eindhoven University of Technology, specifically a PHD researcher experience with operational technology security and an assistant professor familiar with design and function of academic networks. The network model includes containment actuators at their respective network locations.

To visualise the cyberattack in the context of the victims network we also construct Attack-Actuator diagrams of the cyberattacks. The Attack-Actuator diagram includes the relevant steps of the cyberattack. A complete description of the modeling language we have defined for the Attack-Actuator diagrams is provided in appendix A.

Realistic evaluation. As these case studies have been evaluated extensively, at each step of the attack chain it is known what the following step is, and as such which containment actions could stop the attack from progressing. However, in a real world setting the decision-maker would not typically and readily know whether a detected event is a cyberattack, nor what the following attack steps might be, because this information can generally only be inferred as the attack progress. While we look at multiple steps in the attack, we cannot be certain what the following attack steps will be. As such when evaluating a scenario we make the assumption that the step is detected (e.g. by a monitoring environment such as a SOC). While evaluation a scenario we do not take any knowledge about future steps into account, as in a realistic setting this would not be know either. We do present the case studies in a way where if the first scenario is not contained successfully, the attacker moves on to the second, and after that the third scenario. In this way we can showcase how the framework can be used at multiple stages of the cyberattack and is not limited to a specific attack phase. Given these realistic constraints we apply the method to generate a set of containment actions for each investigated scenario. This set is then evaluated on whether the resulting containment actions can be executed in the environment, and whether the containment actions could contain the given scenario.

4. Results

4.1. RQ1: What are the containment technique that can be employed using a containment action?

Table 1 gives an overview of the results of our research into containment techniques. We list the sources in which a containment technique was found. For example, technique 112 - ”reset/lock user credential” was identified in best practices, by domain experts and in academia, while technique 103 - ”stop host” was only identified in academia. The techniques are also grouped under different goals and tactics. The goals of the techniques are the primary grouping, they indicate what can be achieved by using the technique. The first goal is ”incident containment”, employing tactics that directly contain an incident ”isolation” or prevent the attack from reaching other targets ”shrink attack surface”. The second goal is ”slow down adversary”, This includes tactics that hinder the adversary, such as disrupting an attack step ”disrupt attack”, taking offensive measures ”offensive defense” or actions to deceive the adversary ”deceptive defense. The third goal is ”managing information”, further divided in tactics that deal with information by ”gathering”, ”preserving” or ”sharing” data.

4.1.1. Techniques and Best Practises. The source where we identified the largest number of containment techniques is the academic literature with 33 unique techniques, followed by Domain experts with 19 techniques each and Best practices with 18 techniques. In fact, there were only two included techniques that were not identified in academic literature. These techniques are pausing a virtual host and scanning for known IOCs. Both are very practical applications Pausing a virtual machine has many benefits as it simultaneously contains the threat posed by any malware running on the machine, as well as preserve any information that is present on the machine.

Techniques from the best practise source are identified by going through and reading the techniques described in the MITRE frameworks and selecting techniques that related to the containment phase.

4.1.2. Domain experts. During the first week after the survey request email was sent out there were two responses. After the reminder was sent out a week after the request was sent another response arrived. Two months after the survey was sent a fourth and final response was received. As such a total of 4 experts responded to the survey out of the 25 contacted, i.e. a 16% response rate. Any containment action or technique mention by an expert is included in the containment technique matrix. Most notable in the contribution of domain experts is that every experts advises, wherever it is at all possible, to pause a visualised host. This is in clear contrast both with academics, where no virtualisation technologies are mentioned at all, and best practises in mitre attack there is a description of an attack called virtualisation evasion where it states this attack step can not be easily mitigated, and no mention of pausing vitalised machine is identified in the mitre defend framework. Another noticeable thing is the shortage of deceptive defense actions. Only one expert mentioned using some honey pot to learn more of a target and only if possibilities of learning more about the attack are balanced against the risk of further compromise.

4.1.3. Academics. We identified a set of 19 academic papers that include some mention of containment actions or technique. Notable containment techniques identified in academics are those that belong to offensive the defense tactic. These technique are mentioned in Academics, performing a denial-of-service attack to the identified source of an incident, or going even further and attempt the hack the adversary’s systems to stop the incident in this way. In theory these action are be possible, they are however illegal for most incident responders. This may explains why these actions are not mentioned in best practises or by domain experts.

Other techniques that are only identified in academics are isolating a network, stopping a host, patching vulnerabilities, disabling a service, remote logging and logging to unchangeable media.

4.2. RQ2: What are the containment actuators that can execute containment actions in a network?

Table 2 shows the resulting generic actuators, and their classification in the actuators groups and sub-groups. For further clarity a more concrete example is included for each actuator. The group set of actuators all deal in network management and facilitating the communication within and outside of the network. These are further divided in equipment and security monitoring actuators. The network equipment is categories based on OSI level capabilities, meaning whether they can affect communication on the corresponding layer in the OSI-model. To the security monitoring subgroup belong the network-based security agents, to this actuator belong technologies like intrusion detection and prevention systems(IDS and IPS). After network management come the host systems, first the management systems like the patch management system, infrastructure-wide access control, VPN management and virtualisation orchestrator. The management systems exist on one or multiple machines and provide some function to the organisation. The endpoint systems can be present on every machine in the network, these are the host-based security agent and the endpoint-level access control. The final actuator included is the qualified staff member. They fall into their own category as they are natural human beings and are not part of the network itself, but can influence every part of it.

TABLE 1. CONTAINMENT TECHNIQUE SOURCING MATRIX

Goal	Tactic	ID	Technique	Best practises	Domain experts	Academics
incident containment	isolation	100	traffic filtering/rerouting	X	X	X
		101	disconnect/isolate host	X	X	X
		102	disconnect/isolate network			X
		103	stop host			X
		104	pause virtual host		X	
	shrink attack surface	110	patch vulnerability			X
		111	disable service			X
		112	reset/lock user credentials	X	X	X
		113	restrict user activity	X	X	X
		114	executable allow/denylisting	X		X
slow down adversary	disrupt attack	200	break TCP connection/terminate session		X	X
		201	terminate user session	X	X	X
		202	terminate executable/program		X	X
		203	stop host			X
		204	pause virtual host		X	
		205	disable service			X
		206	force additional authentication	X		X
	offensive defence	210	warn intruder			X
		211	denial-of-service attack			X
		212	system compromise attack			X
	deceptive defense	220	add decoy file	X		X
		221	add decoy network resource	X		X
		222	add decoy user accounts	X		X
		223	enable honeypot/smokepot	X	X	X
manage information	gathering	300	add additional logging	X	X	X
		301	add additional IDS	X	X	X
		302	scan for known IOC's	X	X	
		303	enable honeypot/honey net	X	X	X
		304	allow operation on fake file	X		X
	preserving	310	enable remote logging			X
		311	logging to unchangeable media			X
		312	restrict user activity	X	X	X
		313	create backups	X	X	X
		314	backup tampered with files		X	X
	sharing	320	generate an alarm		X	X
		321	generate a report		X	X

4.3. RQ3: How to generate a mapping between containment actuators and containment techniques?

4.3.1. Actuator-Technique mapping first round. Filling in the complete matrix took a longer time than we had expected, the experts reported taking between 45 minutes and an hour to complete the exercise. When discussing the results with the experts one mentioned that after half an hour the judgement became less thoroughly thought through and more rushed as they wanted to be finished with the exercise. Table 3 shows the agreement scores between each expert Actuator-Technique matrix after the first round. The average agreement score between two experts was 77,6% and the agreement score between all matrices was 66,4%. After this first round we discussed the resulting Actuator-Technique matrices with the respective experts to get the reasoning behind every decision. Here we learned that, despite the already high level of agreement between the experts, the interpretation of some actuators and techniques differed greatly between the experts. Mainly the capabilities of staff members varied from action any employee could take, to assuming they are highly specialized technicians.

4.3.2. Actuator-Technique mapping second round. Table 3 shows the agreement scores between each experts Actuator-Technique matrix after the mapping exercise. The average agreement score between two experts was 90,9% and the agreement score between all matrices of 86,3%. The much improved agreement scores suggest that a substantial part of the disagreement between experts was indeed due to the interpretation of the actuator and technique definition. In some cases all experts did agree on the definition, and still did not agree on which techniques could be performed by a given actuator. A example of this is the capabilities of host-based security agents, an expert was familiar with some instance that is capably of terminating a user session, where the other experts where not familiar with this specific application. Overall, the dividing factor seems to be familiarity with a function of an security agent, which was able to employ a technique.

TABLE 2. Generalised containment actuators

Actuator group	Actuator subgroup	Actuator	Icon	Example
Network management	**Networking equipment**	OSI level 5-7 capable	⬡ 5-7	SDN orchestrator
		OSI level 4 capable	⬡ 4	Firewall
		OSI level 3 capable	⬡ 3	Router
		OSI level 2 capable	⬡ 2	Switch
	Security monitoring	Network-based security agent	◇	IDS
Host systems	**Management systems**	Patch management process	○	patch system
		Infrastructure-wide access control	🔑	Active Directory
		VPN management	VPN	employee gateway
		Virtualisation orchestrator	⚙	virtualisation software
	Endpoint systems	Host-based security agent	🛡	antivirus application
		Endpoint-level access control	🔑	login service
Staff	**Staff**	Qualified staff member	🧍	a SOC operator

TABLE 3. agreement score(%) after first mapping

	Expert 1	Expert 2	Expert 3
Expert 1	X	76.4	75.9
Expert 2	76.4	X	80.6
Expert 3	75.9	80.6	X

TABLE 4. agreement score(%) after mapping exercise

	Expert 1	Expert 2	Expert 3
Expert 1	X	89.1	91.7
Expert 2	89.1	X	91.9
Expert 3	91.7	91.9	X

4.3.3. Mapping actuators and Techniques. Table 5 gives the final resulting Actuator-Technique matrix obtained by adding together the experts matrices after the mapping exercise. Out of the 432 total possible intersection between actuators and techniques all experts agreed on applicability of 68 of these. Additionally all experts agreed that 305 are not applicable. There are 59 intersections the experts did not agree on, of which 27 where two experts indicated applicability, and 32 where only one expert indicated applicability. Notable are two actuators where the experts still disagree to a high degree: the network-based security agent (72% agreement), and the host-based security agent (61% agreement). There are also some techniques with low agreement percentages, these are: disconnect/isolate network(67% agreement), disable service(58% agreement), break TCP connection/terminate session(58% agreement) and terminate user session(42% agreement). The experts are much more in agreement about other actuators: The endpoint-level access control, infrastructure-wide access control and VPN management all achieved 94% agreement.

5. Framework application

5.1. Case study 1 - Stuxnet

5.1.1. Background. The first use case is based on a cyber attack on the Natanz nuclear enrichment plant. The attack used the Stuxnet computer worm, which was designed to target supervisory control and data acquisition (SCADA) systems in an operational technology (OT) environment. The OT environment refers to control systems in factories and industrial processes, for example the systems controlling the different sensors and pumps. This use case is based on descriptions of how the Stuxnet attack was able to penetrate into the most secure parts of the Natanz nuclear enrichment facility, despite the actual OT subnet of the network being airgapped from most of the rest of the network and internet. In order for it to have been able to achieve it's goal the worm was extremely sophisticated. [25] It is believed to be developed by multiple nation state actors and used enormous amounts of resources and time to develop. The developers of the malware took care to ensure that only the intended target would be affected. The malware was capable of performing a man in the middle attack and generated fake industrial process controller signals so no abnormal behaviour would be detected. This level of sophistication is unusual for computer malware. [17] The cyberattack took place over multiple months with multiple patches and updates being applied to the malware. The result of the cyberattack was that the nuclear enrichment program of the Iranian government was delayed and centrifuges where damaged beyond repair. It was reported that over a thousand centrifuges were damaged due to the malware. [18] This was the first real world instance of a cyberattack inflicting physical damage.

5.1.2. Network model description. A network diagram representing the facility is presented in figure 2. The layout follows a multi layer model as is common in OT networks. The network model consists of three subnets

TABLE 5. Actuator-Technique Matrix

		Actuator											
ID	Name	Qualified staff member	OSI Level 2	OSI Level 3	OSI Level 4	OSI Level 5-7	Network-based security agent	Patch management process	Infrastructure-wide access control	Vpn management	Virtualisation orchestrator	Host-based security agent	Endpoint-level access control
100	traffic filtering/rerouting		●	●	●	●	○					○	
101	disconnect/isolate host	●	●	●			○				○	●	
102	disconnect/isolate network	◐	●	●			○		○		○		
103	stop host	●									●	◐	
104	pause virtual host	◐									●	○	
110	patch vulnerability	●						●				◐	
111	disable service	●	◐	◐	◐	◐					◐	●	
112	reset/lock user credentials	●							●			◐	●
113	restrict user activity	●							●			◐	●
114	executable allowlisting/blocklisting											●	◐
200	break TCP connection/terminate session			○	◐	●	○			○		◐	
201	terminate user session			○	○	◐	○	○	○	●		○	●
202	terminate process	●						○				●	
203	stop host	●									●	●	
204	pause virtual host	●									●		
205	disable service	●	◐	◐	◐	◐					◐	●	
206	force additional authentication								●		○		◐
210	warn intruder	●										○	
211	denial-of-service attack	●											
212	system compromise attack	◐											
220	add decoy file	●										◐	
221	add decoy network resource	◐					○				●		
222	add decoy user accounts	●					○		●			○	●
223	enable honeypot/smokepot	○					○				●		
300	add additional logging	●					●					●	
301	add additional IDS						○				●		
302	scan for known IOC's	●					●					◐	
303	enable honeypot or honey net						○				●		
304	allow operation on fake file											◐	
310	enable remote logging						●			○		●	
311	logging to unchangeable media	●					●						
312	restrict user activity					○			●			○	●
313	create backups	●						○			○	●	
314	backup tampered with files	●										●	
320	generate an alarm	●	◐				●					●	
321	generate a report	●					●					●	

● - all experts agreed on applicability.
◐ - more than one but not all experts indicated applicability.
○ - one expert indicated applicability.
  - all expert agreed on no applicability.

which are connected with firewalls between them.
The first subnet is the office network, this is the only section of the network that has a direct connection to the internet. Here are all engineering and any administrative workstations. The second subnet in the control network, connected to the office network through a firewall. This layer of the facility consists of the SCADA workstation that manage and monitor the Programmable logic controllers (PLCs).
Finally, there is the OT network where all the PLCs that directly control the centrifuges(via sensors and actuators, which are abstracted from this model) reside.
As part of set-up and maintenance operations there are the workstations that are used to program and manage the PLCs themselves. These workstations run Siemens Step 7 software. These machine must connect directly to the PLCs in order to apply updates and/or patches. The updates and/or patches are created and distributed by Siemens and had to be transferred to the programming workstations. This could be done by connecting the programming workstations to the office network and downloading them directly, or by downloading the update packages and transferring them to a file server connected to the control network, from which the programming workstation could download it. We however for this case study assume that the programming workstation download these packages directly after connecting to the office network.

The facility is set up in different segments with increasing levels of security. This leads to some specific considerations that would not apply to just any network. The business case is also highly secure and confidential. The used attack delivery method was via a USB drive, this makes it hard for the adversary to perform at a larger scale while staying covert, which was a goal of the adversary.

5.1.3. Attack model description. The Attack-Actuator diagram visualising the cyberattack for this case is given in figure 3. The initial attack vector was via USB drive containing the malware. Once the USB drive was plugged into a engineering workstation in the office network it exploited vulnerabilities in the operating system to run some malicious code and infect the machine. Through privilege escalation the malware gain administrator level user rights on the compromised engineering workstation. The malware can perform a lateral movement steps to infect other machines in the same subnet in the same way. When the PLC programmer is connected to the office network the malware performs a lateral movement step from the infected engineering workstation. After the PLC programmer connect to a PLC in the OT network subnet the malware is able to install malware on the PLC. Once the PLCs are infected with the malware the cyberattack is considered successful.

5.1.4. Scenario evaluation. The following provides a breakdown of the attack and the derivation of the relevant containment actions for each scenario. Figure 3 provides an overview of the method output.

First scenario: early containment at workstation level The adversary has gained user level access rights to a workstation in the network. An attempt at a privilege escalation attack step is detected. Since the only machine involved in this attack is the workstation itself, only actuators at the workstation are applicable at this stage. The containment actuators that are present on the workstation itself are a Host-based security agent and some Endpoint-level access control.

To generate the available containment actions, we look up all the containment actions that the present containment actuators can perform. The selection process is visualised in red in figure 4. The selection process remains the same for all scenarios. These resulting containment actions are listed in appendix B.11 and B.12.

By using the method we generate a set of 17 containment actions that can be employed by the two present containment actuators. The related attack states (”user on engineering workstation i”, ”admin on engineering workstation i”) and the attack step (”privilege escalation”) are indicated in the Attack-Actuator diagram in figure 3 by the green circle with a ”1”.

Some applicable containment actions are executing technique 202 by a Host-based security agent and executing techniques 112 and 113 by Endpoint-level access control. By locking/resetting the user credential that have been compromised by the adversary you prevent further access to the host. Alternatively the user privileges of the compromised account execution right can be revoked to prevent the credentials being used to cause harm. By terminating every process launched by the compromised user account, as soon as it is launched, privilege escalation by executing exploits can be prevented.

Second scenario: containing the intermediary steps The adversary has gained administrator level access to a workstation in the network. An attempt to perform a lateral movement attack to gain access to a different non-compromised workstation in the network is detected. This attack step involves the compromised machine, the machine targeted for lateral movement and the networking equipment between these machines. The containment actuators present are the Host-based security agent, Endpoint-level access control or both the compromised machine and the target as well as the network equipment (with OSI level 2 and 3 capabilities).

To generate the available containment actions, we look up all the containment actions that the present containment actuators can perform. The selection process is visualised in red in figure 5. The selection process remains the same for all scenarios. These resulting containment actions are listed in appendix B.2, B.3, B.11 and B.12.

By using the method we generate a set of 23 containment actions that can be employed by the two present containment actuators. The related attack states (”admin on engineering workstation i”, ”user on PLC programmer i”) and the attack step (”lateral movement”) are indicated in the attack diagram 3 by the yellow circle with a ”2”.

Some applicable containment actions are executing technique 100, 101 or 102 by the network equipment. By filtering or rerouting all traffic from the workstation will prevent lateral movement from the workstation to other machines. Disconnected the host entirely from the network will have the same effect as filtering it’s traffic in regard to containment.
By filtering or rerouting all traffic from the workstation will prevent lateral movement from the workstation to other machines. Disconnected the host entirely from the network will have the same effect as filtering it’s traffic in regard to containment.

Third scenario: late detection and preventing infection of PLCs The adversary has infected the software used by the PLC programmer workstation to program and update the PLCs in the OT network. When the PLC programmer is connected to a PLC, an attempt to carry out a lateral motion attack is detected. Since the PLC programmer is directly connected to the PLC and there are no containment actuators on the PLCs themselves, the Host-based security agent and the Endpoint-level access control actuators on the workstation are the only containment actuators present.

To generate the available containment actions, we look up all the containment actions that the present containment actuators can perform. These resulting containment actions are listed in appendix B.11 and B.12.

By using the method we generate a set of 17 containment actions that can be employed by the two present containment actuators. The related attack states (”infected DLL on PLC programmer i”, ”malware on PLC i”) and the attack step (”lateral movement”) are indicated in the attack diagram 3 by the purple circle with a ”3”.

Some applicable containment actions are executing technique 114, 202 or 203 by the Host-based security agent. By placing the programming software to a denylist and consequently terminating the process the malicious software binaries can not infect the PLCs. Stopping the

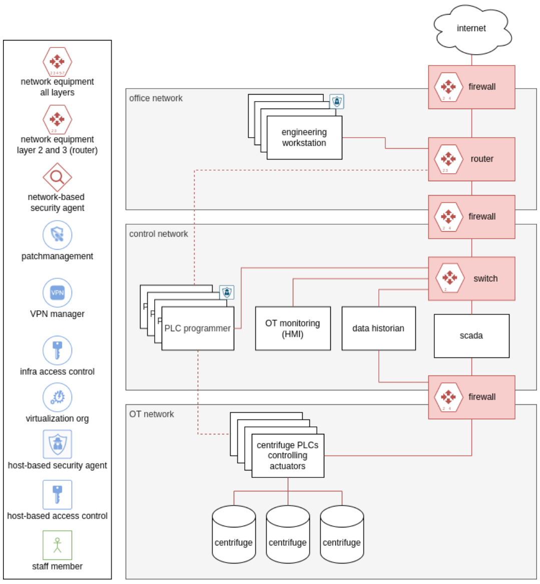


Figure 2. Network diagram for use case 1

host will prevent any further action on or from the machine.

5.2. Case study 2 - Maastricht University hack - from 15 October 2019 until 30 December 2019

5.2.1. Background. The second use case is based on the ransomware attack on Maastricht University(MU) which took place at the end of 2019. During the attack the adversary gained access to the most critical assets of the MU and installed ransomware on over 200 machines. This was the first large scale ransomware attacks on an academic institution in the Netherlands. As a result of the attack 4500 employees and 19000 students weren't able to access their academic resources for some weeks. As both the primary as the back-up servers were affected by the attack the recovery process was extremely complex and some resources were most likely not have been able to be recovered at all. Due to the time pressure to get the students and employees working again, and the prospects of recovery being very poor the MU decided to pay the ransom money of 197000 euro to unencrypt the affected systems.

This use case is based on a descriptions of the attack in a report by FOX IT publish on 5-February-2020 [1].

5.2.2. Network model description. A diagram of an abstract network based on the design of another university network is given in Figure 6. This assumes that both are open networks by definition and that they are likely to be functionally similar at the level of abstraction used in the evaluation. The network model consists of a core router that connects to the Internet and the various subnets that compose the university network. The primary subnet is called the "core network". This is where the administrative machines for the university network are located. The core network subnet contains the Active Directory (AD) servers that act as an infrastructure-wide access control actuator. All other subnets in the network are departmental networks that represent the different departments. A departmental network contains endpoint devices that connect to the university. These devices are, for example, laptops used by students and workstations used by staff members. In addition, a departmental subnet contains servers used by the department behind a firewall. These servers range from department-specific administrative, to research and teaching tasks.

5.2.3. Attack model description. The Attack-Actuator diagram visualising the cyberattack for this case is given in figure 7. The initial attack vector was a phishing email tailored to the organization. The email convinced at least one employee of the targeted organisation to download a malicious Excel document to their endpoint device connected to the department 1 subnet. When the malicious document was opened in the Microsoft Office application, a macro was executed that downloaded and installed the SDBBot remote access Trojan. This Trojan infected the endpoint device and gave the attacker full control of the system. After the initial infection, the attacker performed network scans to find new targets within the department 1 subnet. The attacker infected and gained full control of multiple department servers in the network from the same department. The attacker scraped the memory of the infected department servers and, on one department server, found login credentials with administrator-level user privileges for the Active Directory server in the core network. With these credentials, the attacker had full access to all AD server capabilities. At this point, the attacker would have gained far too much control over the network for any control measure other than (partially) shutting down the network.

5.2.4. Scenario evaluation. The following provides a breakdown of the attack and the derivation of the relevant containment actions for each scenario. Figure 7 provides an overview of the method output.

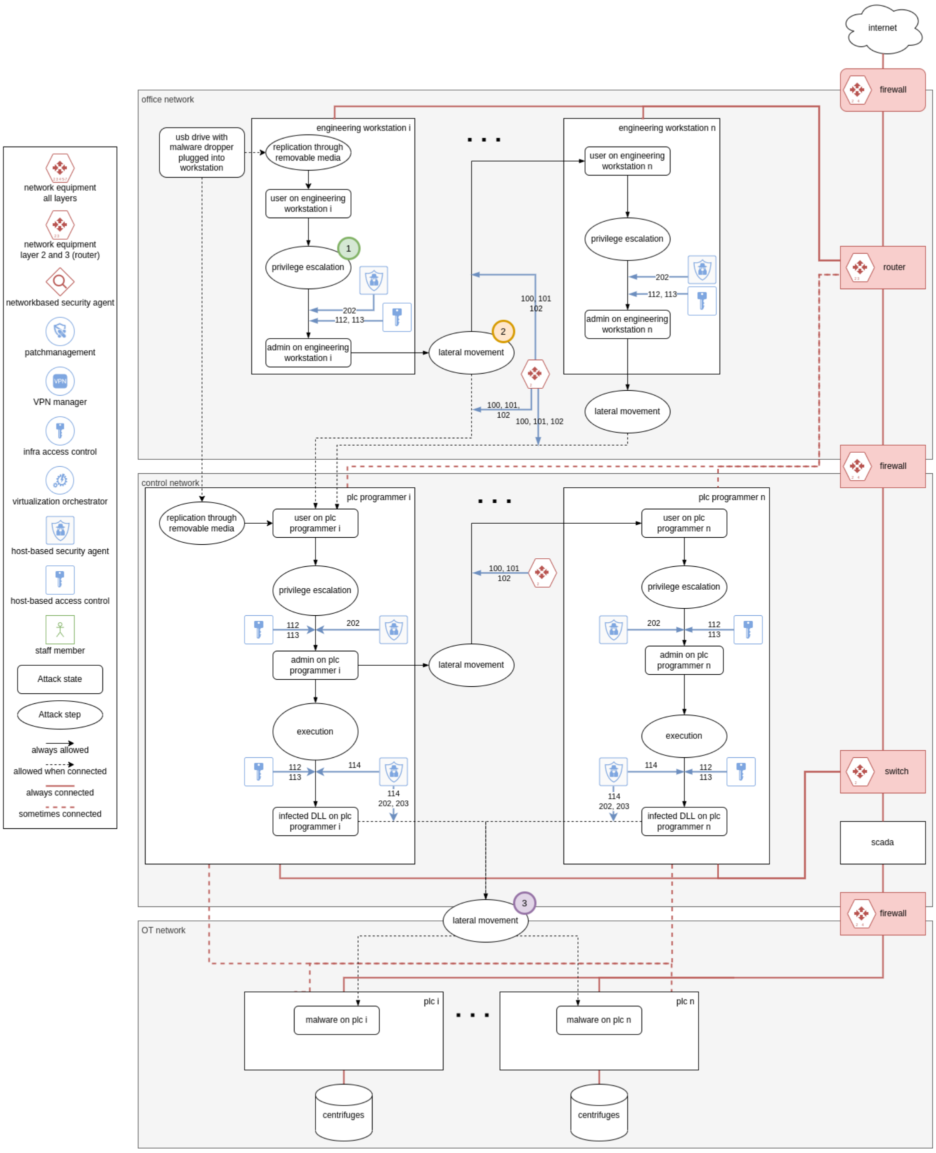


Figure 3. Attack-Actuator diagram for use case 1

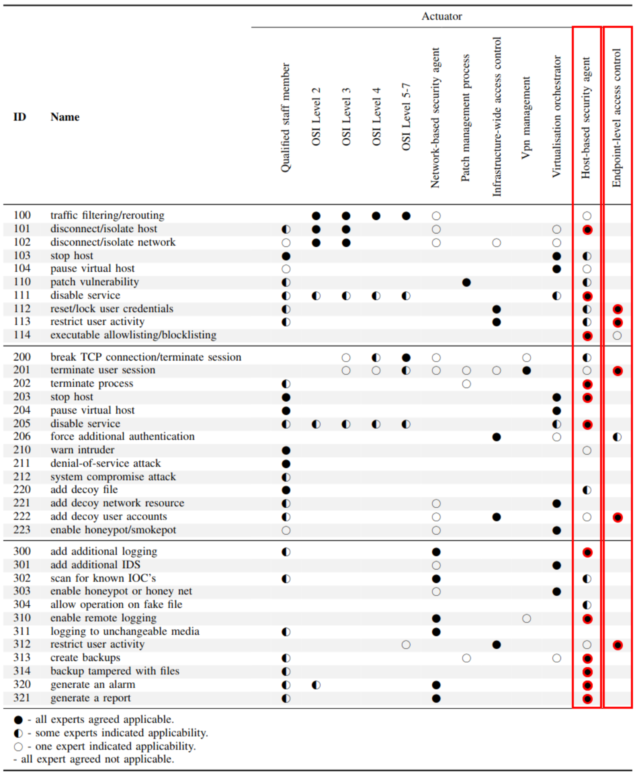
First scenario: containment at first infected workstation The adversary has sent spearphishing emails to employees containing a link that download a malicious file. One of the employees has downloaded the malicious file to their workstation. When the malicious file is opened it is detected that the software opening the files attempts to install some malware. As this attack step happens at the workstation itself the only directly available actuator is the host-base security agent.

To generate the available containment actions, we look up all the containment actions that the present containment actuators can perform. These resulting containment actions are listed in appendix B.11.

By using the method we generate a set of 12 containment actions that can be employed by the present containment actuator. The related attack states ("phishing email containing link to malicious file", "SDBBot malware installed with admin") and the attack step ("Spearphishing link + Office template macros") are indicated in the attack diagram 7 by the green circle with a "1".

Some applicable containment actions are executing technique 101, 114, 202 or 203 by the Host-based security agent. By disconnecting or isolating the host from the rest of the network the incident is guaranteed to be contained to the initial compromised workstation. This does not remove risk to the compromised workstation and resources on the machine. By denylisting the malicious files and/or the application the execution of malicious code will be prevented. This will in this case prevent the the adversaries remote access tool from launching on the workstation. Terminating the application that opened the file as soon as malicious activities are detected, can have the same affect as denylisting the application. Stopping the host will prevent any further action on or from the host.

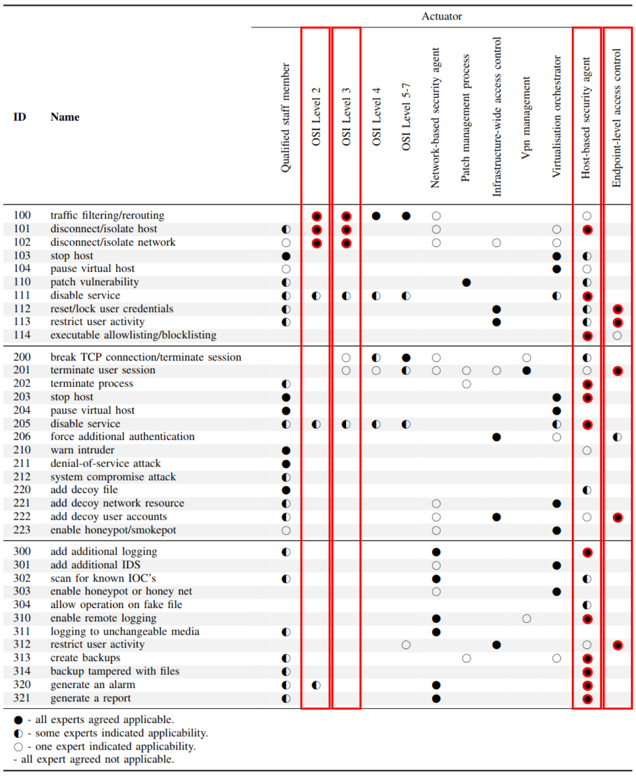
Second scenario: containing lateral movement from initial workstation to other server within the same department Second scenario: containing the intermediary steps The adversary has gained administrator level access to a endpoint device in the department 1 subnet. It is detected that the adversary performs some active scanning activity and attempts lateral movements steps to machines discovered with active scanning. As this attack



		Actuator											
ID	**Name**	Qualified staff member	OSI Level 2	OSI Level 3	OSI Level 4	OSI Level 5-7	Network-based security agent	Patch management process	Infrastructure-wide access control	Vpn management	Virtualisation orchestrator	Host-based security agent	Endpoint-level access control
100	traffic filtering/rerouting		●	●	●	●	○					○	
101	disconnect/isolate host	◐	●	●			○				○	🔴	
102	disconnect/isolate network	○	●	●			○		○		○		
103	stop host	●									●	◐	
104	pause virtual host	○									●	○	
110	patch vulnerability	◐						●				◐	
111	disable service	◐	◐	◐	◐	◐					◐	🔴	
112	reset/lock user credentials	◐							●			◐	🔴
113	restrict user activity	◐							●			◐	🔴
114	executable allowlisting/blocklisting											🔴	○
200	break TCP connection/terminate session			○	◐	●	○			○		◐	
201	terminate user session			○	○	◐	○	○	○	●		○	🔴
202	terminate process	◐						○				🔴	
203	stop host	●									●	🔴	
204	pause virtual host	●									●		
205	disable service	◐	◐	◐	◐	◐					◐	🔴	
206	force additional authentication								●		○		◐
210	warn intruder	●										○	
211	denial-of-service attack	●											
212	system compromise attack	◐											
220	add decoy file	●										◐	
221	add decoy network resource	◐					○				●		
222	add decoy user accounts	◐					○		●			○	🔴
223	enable honeypot/smokepot	○					○				●		
300	add additional logging	◐					●					🔴	
301	add additional IDS						○				●		
302	scan for known IOC's	◐					●					◐	
303	enable honeypot or honey net						○				●		
304	allow operation on fake file											◐	
310	enable remote logging						●			○		🔴	
311	logging to unchangeable media	◐					●						
312	restrict user activity					○			●			○	🔴
313	create backups	◐						○			○	🔴	
314	backup tampered with files	◐										🔴	
320	generate an alarm	◐	◐				●					🔴	
321	generate a report	◐					●					🔴	

● - all experts agreed applicable.
◐ - some experts indicated applicability.
○ - one expert indicated applicability.
- all expert agreed not applicable.

Figure 4. containment action selection using the Actuator-Technique matrix



		Actuator											
ID	**Name**	Qualified staff member	OSI Level 2	OSI Level 3	OSI Level 4	OSI Level 5-7	Network-based security agent	Patch management process	Infrastructure-wide access control	Vpn management	Virtualisation orchestrator	Host-based security agent	Endpoint-level access control
100	traffic filtering/rerouting		🔴	🔴	●	●	○					○	
101	disconnect/isolate host	◐	🔴	🔴			○				○	🔴	
102	disconnect/isolate network	○	🔴	🔴			○		○		○		
103	stop host	●									●	◐	
104	pause virtual host	○									●	○	
110	patch vulnerability	◐						●				◐	
111	disable service	◐	◐	◐	◐	◐					◐	🔴	
112	reset/lock user credentials	◐							●			◐	🔴
113	restrict user activity	◐							●			◐	🔴
114	executable allowlisting/blocklisting											🔴	○
200	break TCP connection/terminate session			○	◐	●	○			○		◐	
201	terminate user session			○	○	◐	○	○	○	●		○	🔴
202	terminate process	◐						○				🔴	
203	stop host	●									●	🔴	
204	pause virtual host	●									●		
205	disable service	◐	◐	◐	◐	◐					◐	🔴	
206	force additional authentication								●		○		◐
210	warn intruder	●										○	
211	denial-of-service attack	●											
212	system compromise attack	◐											
220	add decoy file	●										◐	
221	add decoy network resource	◐					○				●		
222	add decoy user accounts	◐					○		●			○	🔴
223	enable honeypot/smokepot	○					○				●		
300	add additional logging	◐					●					🔴	
301	add additional IDS						○				●		
302	scan for known IOC's	◐					●					◐	
303	enable honeypot or honey net						○				●		
304	allow operation on fake file											◐	
310	enable remote logging						●			○		🔴	
311	logging to unchangeable media	◐					●						
312	restrict user activity					○			●			○	🔴
313	create backups	◐						○			○	🔴	
314	backup tampered with files	◐										🔴	
320	generate an alarm	◐	◐				●					🔴	
321	generate a report	◐					●					🔴	

● - all experts agreed applicable.
◐ - some experts indicated applicability.
○ - one expert indicated applicability.
- all expert agreed not applicable.

Figure 5. containment action selection using the Actuator-Technique matrix

step crosses between the source endpoint device, and all other devices in the department 1 subnet the network equipment, as well as the network-based security agent are available.

To generate the available containment actions, we look up all the containment actions that the present containment actuators can perform. These resulting containment actions are listed in appendix B.2, B.3, B.4, B.6, B.11 and B.12. By using the method we generate a set of 30 containment actions that can be employed by the present containment actuators. The related attack states (”SDBBot malware installed with admin”, ”remote code execution capabilities”) and the attack step (”Active scanning and lateral movement”) are indicated in the attack diagram 3 by the yellow circle with a ”2”.

Some applicable containment actions are executing technique 100, 101 or 102 by the network equipment or 113 by the Host-based security agent. By filtering or rerouting all traffic from the workstation that was scanning the network, the adversary will not learn any additional information about the network. Filtering the traffic from the workstation will also prevent lateral movement from the workstation to other machines. Disconnected the host entirely from the network will have the same effect as filtering it's traffic in regard to containment. The adversary will however be able to notice that the machine is no longer connected. By restricting the relevant, or even all, users capabilities the adversary won't be able to attempt to scan or perform lateral movement in the network.

Third scenario: contain the C&C communication between the adversary and the department server Second scenario: containing the intermediary steps The intrusion detection system detects the outgoing C&C communication originating from one of the department servers in the department 1 subnet. This outgoing connection allows the adversary to perform actions on the machine and extract relevant information. This outgoing connection is initiated by an application running on the department server. As the C&C traffic routes from the server to a different server on the internet the network equipment is available as an actuator. The actuators present on the server are Host-based security agent and Host-based access control.

To generate the available containment actions, we look up all the containment actions that the present containment actuators can perform. These resulting containment actions are listed in appendix B.2, B.3, B.4, B.6, B.11 and B.12. By using the method we generate a set of 30 containment actions that can be employed by the present containment actuators. The related attack states (”remote code execution capabilities”, ”admin on server”) and the attack step (”launch remote access software”) are indicated in the attack diagram 7 by the purple circle with a ”3”.

Some applicable containment actions are executing technique 100 or 101 by the network equipment or executing technique 113, 114 or 202 by the Host-based security agent. By filtering or rerouting the communication with the adversary, the adversary will no longer be able to send commands or extract information directly. This technique can, however, be circumvented and the adversary may notice this. By isolating the server no further information can be extracted from the machine and it cannot infect other devices on the network. Restricting the user privi-

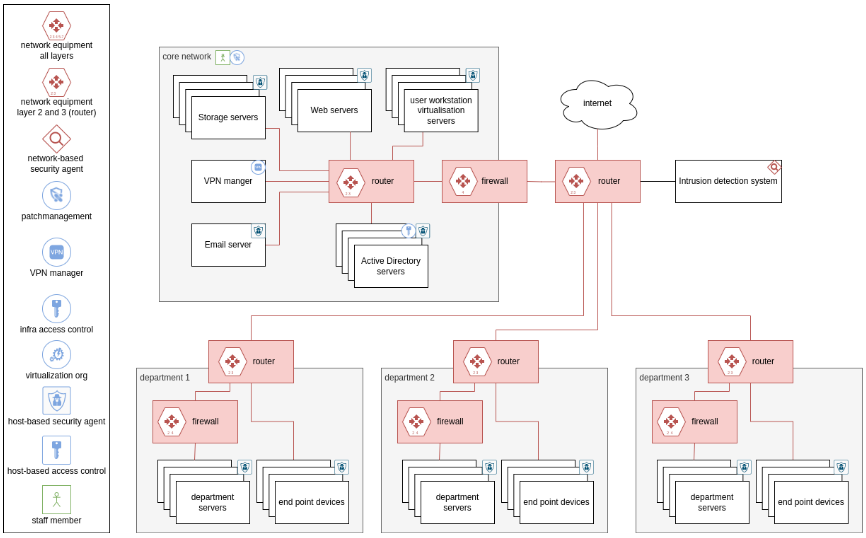


Figure 6. Network diagram for use case 2

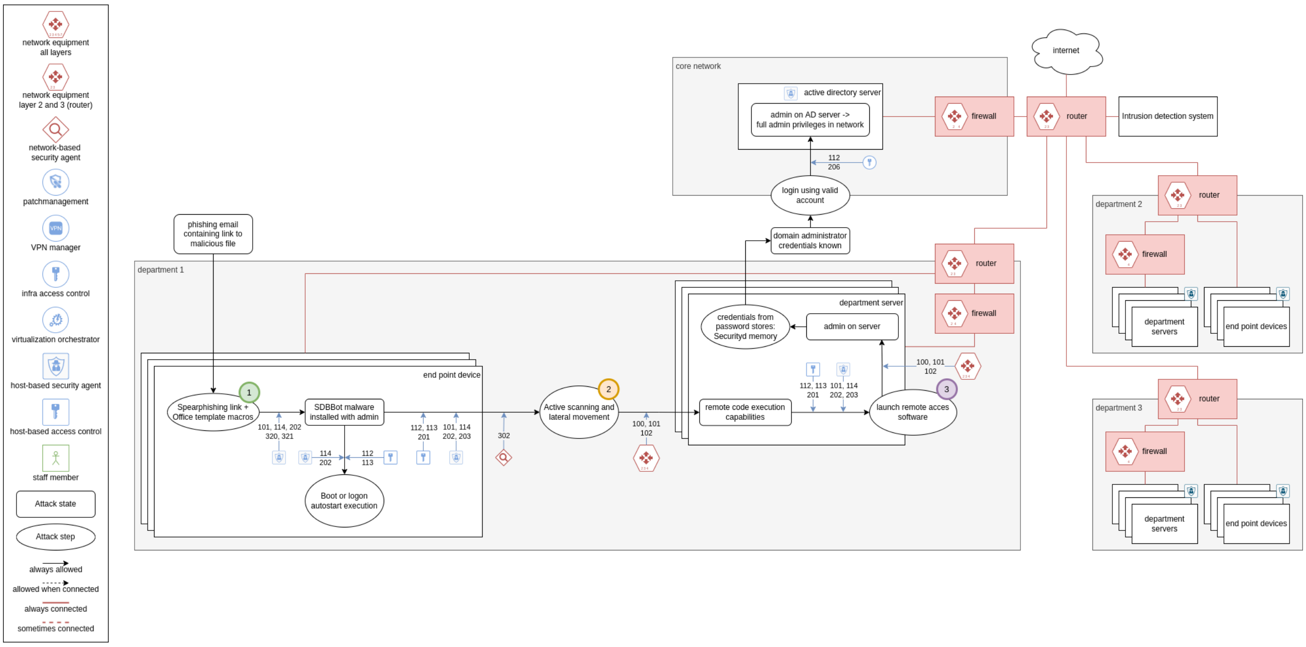

network equipment all layers
network equipment layer 2 and 3 (router)
network-based security agent
patchmanagement
VPN manager
infra access control
virtualization orchestrator
host-based security agent
host-based access control
staff member
Attack state
Attack step
always allowed
allowed when connected
always connected
sometimes connected
phishing email containing link to malicious file
department 1
end point device
Spearphishing link + Office template macros
1
SDBBot malware installed with admin
101, 114, 202
320, 321
114
202
112
113
Boot or logon autostart execution
112, 113
201
101, 114
202, 203
302
Active scanning and lateral movement
2
100, 101
102
core network
active directory server
admin on AD server -> full admin privileges in network
112
206
login using valid account
domain administrator credentials known
department server
credentials from password stores: Securityd memory
admin on server
112, 113
201
101, 114
202, 203
remote code execution capabilities
launch remote acces software
3
100, 101
102
firewall
router
firewall
internet
router
Intrusion detection system
department 2
router
firewall
department servers
end point devices
department 3
router
firewall
department servers
end point devices

Figure 7. Attack-Actuator diagram for use case 2

leges of all user on the system can help preserve data by preventing further harm. By placing the executable that started the communication session with the adversary and consequently terminating the process the malware will not be able to open a C&C channel to the adversary.

6. Discussion

Current academic containment automation proposals lack a way to exhaustively generate and ensure actionability of containment actions. We define containment actions as a technique executed by a specific actuator and as such allow to bridge the gap from abstract reasoning and modeling using techniques to the real world by only allowing actions for which actuators are available. We found that none of the sources we have consulted gave a complete set of the containment techniques. Academia offered the most complete overview, however this is a result of combining many much smaller subsets of techniques. By combining containment techniques from different sources we offer a valuable foundation for future containment automation work.

We see that no domain experts mentions using decoys, while both academics and best practises do mention these techniques. This could be explained by reasoning about the goals of the sources, where academics looks at what is theoretically possible, and best practises aim to learn as much about adversaries as possible. Domain experts on the other hand aim at protecting the victim infrastructure while minimising risk, and allowing an adversary to continue their actions, even inside an honeypot environment, introduces more risk than absolutely necessary.

Another notable difference between the sources is that academic sources mention substantially more techniques than the other sources, 33 vs. 19 and 18. This difference can again be explained by the goals of the source. Whether the action would always be an intelligent option, or even legal is not of concern when testing for theoretic viability of an idea. This less restrictive scope and more exploratory nature allows academics to include techniques that other sources would be less likely to include.

When inspecting the actuators where the experts expressed significant disagreement we see that these actuators are much less narrow defined than the actuators. For example the host-based security agent can range from a simple virus scanner, to software offering complete remote control functionality. As such the difference in experts experience with different security agents can explain the disagreement regarding the actuators capabilities. We believe that more specifically defined actuators would increase the agreement score between experts, as seen with the patch management process and the VPN manager. These actuators both have very high agreement scores (94%). The agreement may be higher with these actuators because they are not too abstract, as there is little variation in the functionality of the implementations of these two actuators. We observe exactly the opposite effect with less narrowly defined actuators such as the network- and host-based security agents with a much lower agreement score (72% and 61%). By keeping the actuators and techniques as concrete as possible, the mapping exercise will be more straightforward with less debatable results.

Disagreement between experts even after ensuring they agreed on the interpretation of the definitions of the techniques and actuators may show an underlying problem with asking experts to identify the capabilities of actuators, it relies of the expert being familiar with all functionality of each possible actuator. What we expect to be a better approach for generating the actuator containment mapping in the physical world would be construct a proof of concept for each containment action. By requiring a practical demonstration there can be no doubt about which techniques can applied using which actuator. This does however require having defined the specific actuators instead of using the abstract actuators as presented in this manuscript. For the purpose of this paper it was not possible to create such a mapping, as we did not have access to a network, nor the time and resources to build one to a realistic scale.

From the application of the framework we learn that the available control actions are highly dependent on the actuators and the design of the network. Defining the actuators in a network can take a lot of effort, but it is necessary to obtain a qualitative optional awareness.

The response to some scenarios may benefit from additional actions not provided by the framework. In the second case study, when it is discovered that the attack vector was a spearfish attack, it would be useful to include other potential victim machines in the containment phase. The framework cannot assist in these types of decisions because a deeper understanding of how a cyber attack propagates is required. The framework only considers control actions and does not provide options outside of this scope. As such, the framework should be considered a useful tool in gaining insight into the available containment options.

6.1. Implications

The work provides a stepping stone for future research on automating incident response. The collected set of containment actions and techniques can be used for multiple purposes in future work. They can form the basis for constructing new and possibly more complex containment plans. Alternatively, the identified set of containment techniques can be used to evaluate proposals for containment automation systems. By creating an Actuator-Technique matrix of all the containment techniques that a containment automation system can perform, one can compare how many of the possible containment techniques are covered by the proposed system. This results in a metric that can be used to evaluate the completeness of a proposed system with respect to the amount of containment techniques covered by the system.

Another result of the research that could prove valuable is the Attack-Actuator diagram modelling language developed for the validation step. The modelling language provides a widely applicable visualisation method of a cyber-attack in the context of a network model, which also displays the available actuators. Future research into automatically extracting network information and generating such a graphical overview would potentially be valuable in providing a clear overview of the options available in the network.

The created set of containment actions and techniques provides a useful set of techniques to implement in an automated containment system. The containment technique matrix gives a fairly complete overview of the current day possibilities in incident containment. Creating an Actuator-Technique Matrix for several popular security devices makes it possible to compare their capabilities in a clear and structured way. This allows security architects to make more informed decisions about which capabilities they would miss in a selection of actuators.

6.2. Limitations

A clear limitation on the domain expert source is that it is based on a very small number of responses to the survey. Therefore, we cannot draw any negative or strong conclusions based on this table. Meaning we cannot say that domain experts do not use a certain technique, only that some domain experts do use a certain technique. As an example regarding technique 103 - ”stop host”, none of our respondents ever mentioned stopping a host, but this does not mean that no domain experts would consider this a viable containment technique.

In this paper we describe a set of abstracted actuators, however if one would apply this framework in the physical world it is advised to tailor the set of actuators to the network. The objective is to map all available actuators in the real world network.

7. Conclusion

The problem we help to solve is to determine how to create optional awareness for containment of an ongoing cyberattack. There is an obvious gap in the definitions of containment actions that exist, and there is also no formalised way to determine when which containment actions can be performed, other than to declare which ones can be performed manually. We want to collect containment actions from various sources and propose a way how only actions that can be performed can be filtered automatically

We provide a framework with which one can define and determine the availability of containment actions. To create this framework, we collected containment techniques and defined abstract actuators that can use these techniques to perform containment actions. By having domain experts perform an exercise where they map techniques to actuators, we can create a qualitative matrix showing for each containment actuator which containment techniques they can apply. These intersections between an actuator and a technique represent containment actions. We reason that if an actuator can affect a certain location in a network, then all containment actions the actuator can perform are available.

To validate the applicability of the proposed framework, we evaluate how the framework can be applied in two real life cyberattacks. We have defined several scenarios that together approximate the real life cyber-attacks. For each of these scenarios, we apply the framework and evaluate the resulting available containment actions and reason whether applicable containment actions are still included in the set generated using the framework.

There is still future work to be done in applying containment automation to the contexts of the Internet of Things, OT or Automotive domains. The infrastructure in these different domains differs significantly from the conventional IT infrastructure, both in terms of the actuators and the containment techniques that are typically used. Moreover, the expectations and interests in these domains are different, e.g. robustness and security are the main priorities in both the OT and Automotive domains, where security is generally not considered a concern in the IT infrastructure. Therefore, an automated containment system that is perfect for a conventional IT infrastructure may not have the desired effect in these other domains. As such more work to identify containment techniques and actuators in different domains still remains.

The proposed framework is based on the knowledge of the available actuators in a network and their capabilities. Currently, an administrator would have to go through the entire network and identify all available actuators to apply the framework in a real world setting. To make this work easier, it would be interesting to create an application that can detect and map actuators in a given network.

References

[1] Dijkstra M., van Dantzig M. (2020, Februari 5). Spoedondersteuning Project Fontana. https://www.maastrichtuniversity.nl/file/foxitrapportreactieuniversiteitmaastrichtpdf

[2] mnemonic, Adversary Emulation Planner, (2021), GitHub repository, https://github.com/mnemonic-no/aep

[3] Li, X., Zhou, C., Tian, Y. C., and Qin, Y. (2018). A dynamic decision-making approach for intrusion response in industrial control systems. IEEE Transactions on Industrial Informatics, 15(5), 2544-2554.

[4] Iannucci, S., and Abdelwahed, S. (2018). Model-based response planning strategies for autonomic intrusion protection. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 13(1), 1-23.

[5] Fessi, B. A., Benabdallah, S., Boudriga, N., and Hamdi, M. (2014). A multi-attribute decision model for intrusion response system. Information Sciences, 270, 237-254.

[6] Zhang, Z., Naït-Abdesselam, F., Ho, P. H., and Kadobayashi, Y. (2011). Toward cost-sensitive self-optimizing anomaly detection and response in autonomic networks. computers & security, 30(6-7), 525-537.

[7] White, G. B., Fisch, E. A., and Pooch, U. W. (1996). Cooperating security managers: A peer-based intrusion detection system. IEEE network, 10(1), 20-23.

[8] Shameli-Sendi, A., Louafi, H., He, W., and Cheriet, M. (2016). Dynamic optimal countermeasure selection for intrusion response system. IEEE Transactions on Dependable and Secure Computing, 15(5), 755-770.

[9] Al Ghazo, A. T., and Kumar, R. (2019, October). Identification of Critical-Attacks Set in an Attack-Graph. In 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0716-0722). IEEE.

[10] Cuppens-Boulahia, N., Cuppens, F., Autrel, F., and Debar, H. (2009). An ontology-based approach to react to network attacks. International Journal of Information and Computer Security, 3(3-4), 280-305.

[11] Carver Jr, C. A. (2001). Adaptive agent-based intrusion response. Texas A&M University.

[12] Heigl, M., Doerr, L., Almaini, A., Fiala, D., and Schram, M. (2018, September). Incident reaction based on intrusion detections' alert analysis. In 2018 International Conference on Applied Electronics (AE) (pp. 1-6). IEEE.

[13] The Mitre Corporation. The MITRE Corporation. (n.d.). Retrieved February 28, 2022, from https://www.mitre.org/

[14] Mitre d3fend knowledge graph. Retrieved February 17, 2022, from https://d3fend.mitre.org/

[15] Mitre shield knowledge graph. Retrieved May 30, 2021, from https://shield.mitre.org/matrix/

[16] Mitre att&ck knowledge graph mitigations. Retrieved May 30, 2021, from https://attack.mitre.org/mitigations/enterprise/

[17] Chen, T. M., & Abu-Nimeh, S. (2011). Lessons from stuxnet. Computer, 44(4), 91-93.

[18] Sanger, D. E. (2012, June 1). Obama order sped up wave of cyberattacks against Iran. The New York Times. Retrieved March 7, 2022, from https://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html

[19] Emerald: Event monitoring enabling responses&nbsp; to Anomalous Live Disturbances. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. (n.d.). Retrieved March 7, 2022, from http://www.csl.sri.com/papers/emerald-niss97/emerald-niss97.html

[20] Dewri, R., Ray, I., Poolsappasit, N., & Whitley, D. (2012). Optimal security hardening on attack tree models of networks: a cost-benefit analysis. International Journal of Information Security, 11(3), 167-188.

[21] Nespoli, P., Papamartzivanos, D., Mármol, F. G., & Kambourakis, G. (2017). Optimal countermeasures selection against cyberattacks: A comprehensive survey on reaction frameworks. IEEE Communications Surveys & Tutorials, 20(2), 1361-1396.

[22] Chung, C. J., Khatkar, P., Xing, T., Lee, J., & Huang, D. (2013). NICE: Network intrusion detection and countermeasure selection in virtual network systems. IEEE transactions on dependable and secure computing, 10(4), 198-211.

[23] Gupta, M., Rees, J., Chaturvedi, A., & Chi, J. (2006). Matching information security vulnerabilities to organizational security profiles: a genetic algorithm approach. Decision Support Systems, 41(3), 592-603.

[24] Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. biometrics, 159-174.

[25] Falliere N., O Murchu L., Chien E. (2011, februari). w32 stuxnet dossier. https://docs.broadcom.com/doc/security-response-w32-stuxnet-dossier-11-en

[26] Scarfone, K., Grance, T., & Masone, K. (2008). Computer security incident handling guide. NIST Special Publication, 800(61), 38.

Appendix A. Definition of visual language used in the verification diagrams

Network diagram:
Machines: Rectangles that have black borders, 90 degree angles and are white filled denote machines. A descriptive name of the machine is inside the rectangle. They can be either physical or vitalized. Machine rectangles that have red borders and are red filled are networking devices. They make up the network infrastructure. A ”stack” of the machine rectangles denote multiple (n) machines that are all connected to the rest of the network in the same way. They have the same relevant properties for the use case.

Physical connections: Red lines are physical connection between machines. The line can be continuous of dashed, the continuous line indicate a permanent connection, where a dashes line indicate a connection that is sometimes present.

Sections: Rectangles with a black border, 90 degree angles and are grey filled denote sections in the network, these are also known as subnets. The sections can be strictly divided sub-nets, or loosely clustered groups of machines. They help in understanding how the network is designed. the sections can contain machines, this is shown by placing the machine rectangle inside the section. Machines inside of a section often have just 1 or two connections to machines outside the network. In most cases this connection point is via a networking machine.

Actuators: Actuators can perform actions relating to one or more machines. Each defined actuator has it's icon. By placing the actuator icon next to, against or on a machine in the diagram, it is show that the actuator can perform actions relating to that specific machines. Actuator icons can also be place in a section rectangle, this show that it can perform actions relating to all machines in that section.

Attack-Actuator diagram: A cyberattack is modeled in an Attack-Actuator diagram. This uses the same components and rules as the networking diagram but adds some more components to denote the attack steps and the containment techniques.

Attack states: An attack state is denoted with a rectangle with black borders, rounded edges and white filling. A description name of the state is inside the rectangle. An attack state can have zero or more black arrows pointing to attack steps that it enables.

Attack steps: An attack step is denoted by a circle or oval with a black border and white filling. A description name of the state is inside the circle/oval. An attack step has at least one black arrow pointing to it and at least on black arrow pointing to a attack state. If and only if all attack states with an arrow pointing to the attack step are true the step can be executed and all states that have an black arrow coming from the attack step will be true as well. It is not allowed to have circles pointing to a previously enabled state. This can be seen as a Directed acyclic graph with the attack states being nodes and the attack steps forming the vertices.

Actuators: Actors can be placed on the graph with blue arrows pointing to the black arrows that connect the attack states and attack steps. On top of the blue arrow one places the containment technique IDs.

Appendix B. actuator to containment technique mapping

B.1. qualified staff member

- 103 - stop host
- 203 - stop host
- 204 - pause virtual host
- 210 - warn intruder
- 211 - denial-of-service attack
- 220 - add decoy file

B.2. OSI level 2 capable

- 100 - traffic filtering/rerouting
- 101 - disconnect/isolate host
- 102 - disconnect/isolate network

B.3. OSI level 3 capable

- 100 - traffic filtering/rerouting
- 101 - disconnect/isolate host
- 102 - disconnect/isolate network

B.4. OSI level 4 capable

- 100 - traffic filtering/rerouting

B.5. OSI level 5-7 capable

- 100 - traffic filtering/rerouting
- 200 - break TCP connection/terminate session

B.6. network-based security agent

- 300 - add additional logging
- 302 - scan for known IOC's
- 310 - enable remote logging
- 311 - logging to unchangeable media
- 320 - generate an alarm
- 321 - generate a report

B.7. patch management process

- 110 - patch vulnerability

B.8. infrastructure-wide access control

- 112 - reset/lock user credentials
- 113 - restrict user privileges
- 206 - force additional authentication
- 222 - add decoy user accounts
- 312 - restrict user activity

B.9. vpn management

- 201 - terminate user session

B.10. virtualisation orchestrator

- 103 - stop host
- 104 - pause virtual host
- 203 - stop host
- 204 - pause virtual host
- 221 - add decoy network resource
- 223 - enable honeypot/smokepot
- 301 - add additional IDS
- 303 - enable honeypot or honey net

B.11. Host-based security agent

- 101 - disconnect/isolate host
- 111 - disable service
- 114 - executable deny/allowlisting
- 202 - terminate process
- 203 - stop host
- 205 - disable service
- 300 - add additional logging
- 310 - enable remote logging
- 313 - create backups
- 314 - backup tampered with files
- 320 - generate an alarm
- 321 - generate a report

B.12. Endpoint-level access control

- 112 - reset/lock user credentials
- 113 - restrict user privileges
- 201 - terminate user session
- 222 - add decoy user accounts
- 312 - restrict user activity

B.13. all techniques

- 100 - traffic filtering/rerouting
- 101 - disconnect/isolate hos
- 102 - disconnect/isolate network
- 103 - stop host
- 104 - pause virtual host
- 110 - patch vulnerability
- 111 - disable service
- 112 - reset/lock user credentials
- 113 - restrict user activity
- 114 - executable allowlisting/blocklisting
- 200 - break TCP connection/terminate session
- 201 - terminate user session
- 202 - terminate process
- 203 - stop host
- 204 - pause virtual host
- 205 - disable service
- 206 - force additional authentication
- 210 - warn intruder
- 211 - denial-of-service attack
- 212 - system compromise attack
- 220 - add decoy file
- 221 - add decoy network resource
- 222 - add decoy user accounts
- 223 - enable honeypot/smokepot
- 300 - add additional logging
- 301 - add additional IDS
- 302 - scan for known IOC's
- 303 - enable honeypot or honey net
- 304 - allow operation on fake file
- 310 - enable remote logging
- 311 - logging to unchangeable media
- 312 - restrict user activity
- 313 - create backups
- 314 - backup tampered with files
- 320 - generate an alarm
- 321 - generate a report

Appendix C. Survey results

We understand that you (main) occupation is within the domain of cybersecurity and incident handling. Which of the following tasks/competencies best describe your day to day activities in that role?
SOC operations;Monitor for security threats;
Making security policies;
SOC operations;Respond to security incidents;
Other;

How many years of experience do you have in the field of cyber security?
1: over 10 years
over 10 years
6-10 years
over 10 years

C.1. scenario 1

Assume that an adversary has gained user level access to workstation W2. This allows the adversary to do everything a user on that system can do. How suitable is each of the following actions in this situation?
1: Add network filtering;Disconnect W2 from the network;Add targeted monitoring;
2: Disconnect W2 from the network;Remove authentication of the adversary on W2;Add targeted monitoring;Add network filtering;Add a trigger;
3: Disconnect W2 from the network;Add targeted monitoring;Add network filtering;Shutdown W2;Remove authentication of the adversary on W2;Add a trigger;
4: Disconnect W2 from the network;

Is there an action that you think would be more appropriate in this situation?
1: Reset all credentials exposed at W2 which are considered compromised.
2: Revoke accces for revelant user-id domain-wide.
3: Make a forensic image of W2 (disk and memory)
4: .

Explain your assumptions and choices for this situation.
1: EDR-level endpoint network isolation; block potential connectivity to command and control server from entire network. Monitor for credential (mis)usage, also in other places in the network. Note: there may be situations

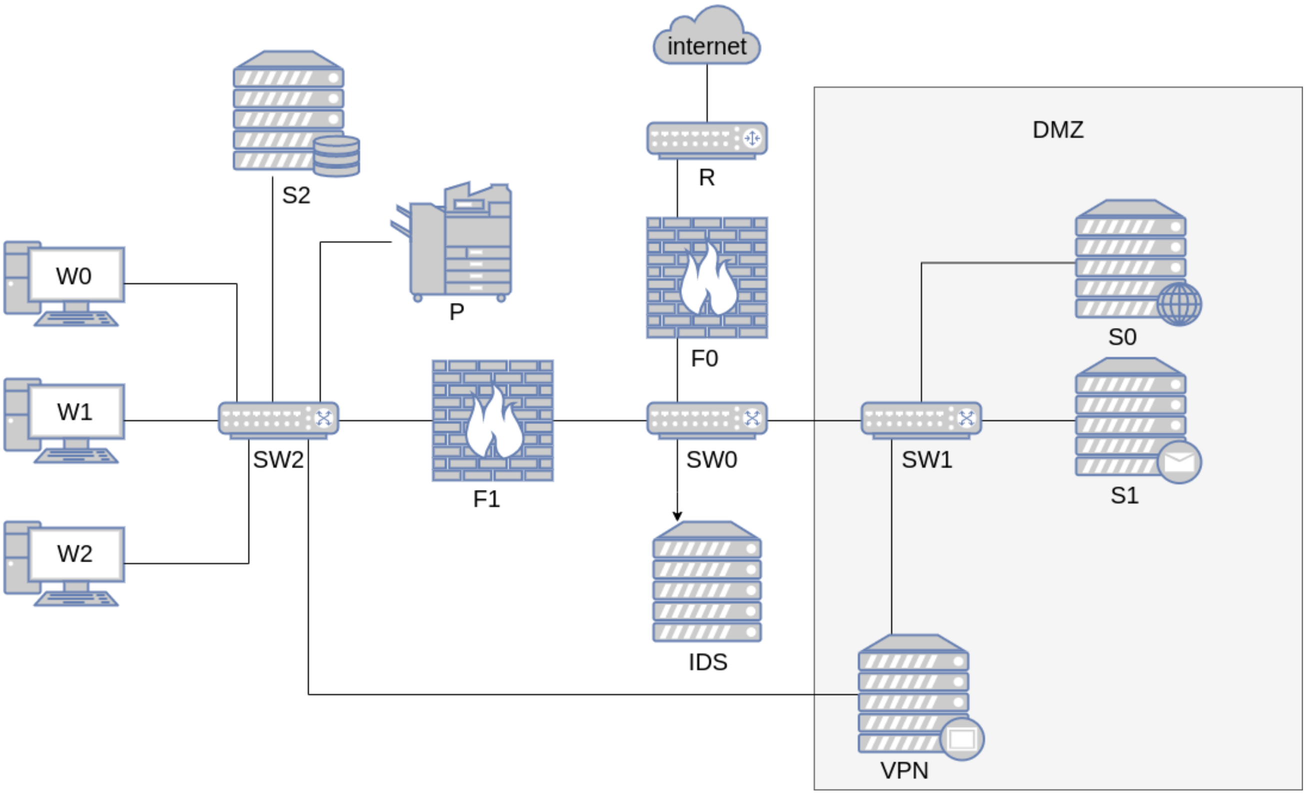


Figure 8. network model used in survey, including the names of each machine

where immediate containment is not advised before getting better understanding of the incident (e.g. in cast of suspected APT attack).
2: There is a chance that lateral movement has taken place, so any use of the compromised account must be detected and stopped. The infected PC must be kept alive, in order to analyze the malware, even if is only resides in memory.
3: Assumption is that the attacker already got further into the network.
4: I assumed that this workstation is on the internal company network, since the DMZ is on the right. In order to be able to do memory forensics, the computer should not be shut down. Network isolation mitigates the immediate threat while preserving the memory. I also assumed that 2. includes logical network isolation through endpoint agents.

C.2. scenario 2

Assume that an adversary has gained root level access to email server S1. This means that the adversary has gained full rights and capabilities on the machine. How suitable is each of the following actions in this situation?
1: Add network filtering;Disconnect S1 from the network;Disable the email service;Add targeted monitoring;
2: Disconnect S1 from the network;Add targeted monitoring;
3: Disconnect S1 from the network;Add targeted monitoring;Add network filtering;Shutdown S1;Disable the email service;Remove authentication of the adversary on S1;Add a trigger;
4: Disconnect S1 from the network;

Is there an action that you think would be more appropriate in this situation?
1: Reset all credentials exposed at S1 which are considered compromised.
2: No. The server must be considered compromised and lost. Find out when the infection occurred, and reinstall the system from a backup earlier then the infection. Also closely monitor other servers for signs of infection.
3: Make a forensic image of the system.
4: .

Explain your assumptions and choices for this situation.
1: Block connectivity to/from intruder IP(s) from entire network. Monitor for credential (mis)usage, also in other places in the network.
2: You can not recover from a root-level compromise. Always wipe the whole system (possibly even swap hardware).
3: I assume that the attacker has access to several machines in the network.
4: .

C.3. scenario 3

Assume that an adversary has acquired one of the employees VPN credentials. And is using then to connect to scan the rest of the network. How suitable

is each of the following actions in this situation?
1: Break the VPN connection used by the adversary;Add network filtering;Remove authorization of the used credentials;
2: Remove authorization of the used credentials;Break the VPN connection used by the adversary;Add targeted monitoring;Add network filtering;
3: Remove authorization of the used credentials;Break the VPN connection used by the adversary;Add network filtering;Disconnect the VPN server from the rest of the network;Disable VPN service;Shutdown VPN server;Add targeted monitoring;Add trigger;
4: Remove authorization of the used credentials;Break the VPN connection used by the adversary;Add targeted monitoring;

Is there an action that you think would be more appropriate in this situation?
1: Reset credentials of affected user.
2: For the future: implement a multi factor authentication mechanism.
3: Forensic investigation on the VPN server
4: If possible, redirect the adversary to deception infrastructure.

Explain your assumptions and choices for this situation.
1: Block connectivity to/from intruder IP address(es) from/to entire network. Monitor for credential (mis)usage, also in other places in the network.
2: The network monitoring and filtering is done in order to determine the source of the attack, and monitor / block any future attacks from this source.
3: Credentials are probably phished or guessed.
4: .

C.4. scenario 4

Assume that the intrusion detection system IDS has detected some command and control traffic coming from server S2 going to a previously unseen IP address which appears to belong to a large cloud provider. How suitable is each of the following actions in this situation?
1: Add network filtering;Add targeted monitoring;Disconnect Server S2 from the network;
2: Add targeted monitoring;Disconnect Server S2 from the network;Add network filtering;Shutdown Server S2;Reset all credentials on server S2;
3: Add targeted monitoring;Add network filtering;Stop the process on server S2 that is communicating;Disconnect Server S2 from the network;Reset all credentials on server S2;Shutdown Server S2;Add trigger;
4: Disconnect Server S2 from the network;Stop the process on server S2 that is communicating;

Is there an action that you think would be more appropriate in this situation?
1: Reset all credentials exposed at S2 which are considered compromised.
2: Just as before with the compromised email server, consider it lost, and rebuild from scratch.
3: Create a forensic image
4: .

Explain your assumptions and choices for this situation.
1: Block connectivity to/from command and control address(es) from/to entire network. Monitor for credential (mis)usage, also in other places in the network. Monitor for network anomalies related with S2. Note: decision to disconnect / take down the server usually needs more information and in some cases business approval to determine risk of interrupting core business process(es) vs risk related keeping compromised host in the network.
2: If malware is found on a server, it is nearly impossible to determine how much information is already stolen. A clean install is always required, followed by a careful restore of the data.
3: Other machines are probably infected
4: .

C.5. scenario 5

Assume an incident where an adversary has gained access to a server (S2) hosting a database service containing customer data. An encrypted data flow using the TCP protocol is detected going from S2 to some destination on the internet. How suitable is each of the following actions in this situation?
1: Disconnect Server S2 from the network;Add network filtering;Change Database (and the server) credentials and reset all connections.;Add targeted monitoring;
2: Disconnect Server S2 from the network;
3: Break the connection by sending a forged TCP reset command to S2;Add network filtering;Add targeted monitoring;Disconnect Server S2 from the network;Change Database (and the server) credentials and reset all connections.;Disable Database service on S2;Shutdown Server S2;Add a trigger;
4: Disconnect Server S2 from the network;

Is there an action that you think would be more appropriate in this situation?
1: Reset all credentials exposed at S2 which are considered compromised.
2: Same answer as the previous 2. Wipe and reinstall.
3: Create a forensic image
4: followed by a forensic investigation and incident response.

Explain your assumptions and choices for this situation.
1: Block connectivity to/from destination IP address(es) from/to entire network. Monitor for credential (mis)usage, also in other places in the network. Monitor for network anomalies related with S2.
2: Same answer as the previous 2. Wipe and reinstall.
3: .
4: .

C.6. scenario 6

Assume an incident where an adversary has been able to install some software on a employees workstation W1. The attack vector was an email attachment opened by the employee. It is unclear at this point what the capabilities of this software are. How suitable is each of the following actions in this situation?

1: Add network filtering;Disconnect Workstation W1 from the network;Change credentials of Workstation W1;Add targeted monitoring;
2: Disconnect Workstation W1 from the network;Change credentials of Workstation W1;Add targeted monitoring;
3: Disconnect Workstation W1 from the network;Add network filtering;Shutdown workstation W1;Add targeted monitoring;Change credentials of Workstation W1;
4: Disconnect Workstation W1 from the network;

Is there an action that you think would be more appropriate in this situation?

1: Reset all credentials exposed at W1 which are considered compromised.
2: Monitor closely for any lateral movement, and analyse the malware on the workstation. After that, wipe the workstation. Try to find out what data or systems were accessed from the workstation and scan for IOC's.
3: Create a forensic image of workstation 1
4: Pause virtual desktop (assuming that the employee is using virtualised workstation)

Explain your assumptions and choices for this situation.

1: EDR-level endpoint network isolation; block potential connectivity to command and control server from entire network. Monitor for credential (mis)usage, also in other places in the network. Note: there may be situations where immediate containment is not advised before getting better understanding of the incident (e.g. in cast of suspected APT attack).
2: A clean install (or restore from clean backup) is the only solution for an infection.
3: This incident is probably (!) only affecting the workstation if the infection was pretty recent.
4: .

C.7. scenario 7

Assume an incident where an adversary has been able to install some malware on server S2. It is currently still unclear what the attack vector was. It is also unclear what the capabilities of this software are. How suitable is each of the following actions in this situation?

1: Add network filtering;Add targeted monitoring;Disconnect server S2 from the network;
2: Disconnect server S2 from the network;Change all credentials on server S2;Add targeted monitoring;
3: Disconnect server S2 from the network;Add network filtering;Shutdown server S2;Add targeted monitoring;Disable services running on S2;Change all credentials on server S2;
4: Disconnect server S2 from the network;Add targeted monitoring;

Is there an action that you think would be more appropriate in this situation?

1: Reset all credentials exposed at S2 which are considered compromised.
2: Same as previous.
3: Forensic imaging
4: (depending on the situation). If there are too many unknowns then the possibility of learning more about the attack has to be balanced against the risk of further compromise.

Explain your assumptions and choices for this situation.

1: Block connectivity to/from potential command and control address(es) from/to entire network. Monitor for credential (mis)usage, also in other places in the network. Monitor for network anomalies related with S2. Note: decision to disconnect / take down the server usually needs more information and in some cases business approval to determine risk of interrupting core business process(es) vs risk related keeping compromised host in the network.
2: Same as previous.
3: A triage on the image should be able to shed some light on the point of entry. Other components should be checked for similar malware or IoCs
4: .

Are there any additional incidents that you think would be relevant to keep in mind for this research? If so please include a description below

1: Ransomware-related
2: .
3: .
4: .

Are there any containment tactics, techniques or actions missing from the Containment Action Matrix? https://docs.google.com/spreadsheets/d/1WGH9oYGT-R0tNQjpz0kMJIF4p-FIJB1ZFet3j8aTbBU/edit?usp=sharing If so please include a description below

1: Reset of all potentially exposed credentials (both for user and service accounts) - not on affected host but centrally managed (e.g. Active Directory); Triggering endpoint network isolation on Endpoint Detection and Response (EDR) tool level
2: .
3: .
4: You might add a ”deceive” tactic (redirect the attack to deception infrastructure/honeypot).