SOCCRATES

SOC & CSIRT Response to Attacks & Threats based on attack defence graphs Evaluation Systems

D3.1

Initial ADG based Attack prototypes

Deliverable type:	DEM
Contributing work packages:	WP3 ADG based Attack Analysis
Due date of deliverable:	28/02/2021
Submission date:	25/02/2021
Dissemination level:	PU

Responsible organisation:	F-Secure
Editors:	Marko Komssi & Ville Alkkiomäki
Revision:	1.0

Abstract	The overall objective of Work Package 3 is to develop the Infrastructure Modelling and Attack Defence Graph analyser & Course of Action Generator components of the SOCCRATES Platform. This deliverable describes the initial versions of the Infrastructure modelling and Attack Defence Graph components.
Keywords:	Infrastructure modelling, Attack Defense Graphs, Attack Simulations

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 833481
Call H2020-SU-ICT-2018 • Innovation Action • Start date: September 1st, 2019

Classification level: Public

Management summary

The main objective of SOCCRATES is to develop and implement a security automation and decision support platform for SOCs and CSIRTs that will significantly improve an organisation's capability to quickly and effectively detect and respond to new cyber threats and ongoing attacks. The SOCCRATES platform consists of eleven components. This deliverable describes the initial prototype implementations of two of the components, the Infrastructure modelling component (IMC) and Attack Defence Graph.

The reader is assumed to be familiar with the three other SOCCRATES deliverables. D2.2 contains a description of the SOCCRATES components and IMC and ADG are introduced in Section 4 and Section 5 of D2.2, respectively. This deliverable also builds on D6.1 and D7.1 that provide the details of deployment environment and the pilot specification & plan, respectively.

Contributor(s)	Marko Komssi (F-Secure) Ville Alkkiomäki (F-Secure) Seamus Moloney (F-Secure) Erik Ringdahl (Foreseeti) Per Eliasson (Foreseeti) Frank Fransen (TNO)

Reviewer(s)	Christophe Kiennert (Institut Mines-Télécom) Andrew Patel (F-Secure)

Security Assessment	
Approval Date	11/02/2021
Remarks	none

TABLE OF CONTENTS

1 INTRODUCTION – RATIONALE OF THIS DOCUMENT 6

1.1 The SOCCRATES project 6
1.2 This deliverable 7
1.3 Structure of this deliverable 7

2 INFRASTRUCTURE MODELLING 8

2.1 Introduction 8
2.2 System architecture 8
2.3 Deployment 10
2.3.1 RADAR Cloud Security Center 10
2.3.2 RADAR Scan node 11
2.3.3 RADAR IMC Adapter 11
2.3.4 IMC Database and REST APIs 12
2.4 Verification of IMC Capabilities 12
2.4.1 Automated infrastructure scanning 12
2.4.2 Continuous updates 12
2.4.3 Time dimension 13
2.4.4 Extendable model 13
2.4.5 Multi-organization 13
2.5 Next steps 13

3 ATTACK DEFENCE GRAPHS 15

3.1 Introduction 15
3.2 Current system architecture 15
3.2.1 Model generation 15
3.2.2 Attack simulation 16
3.2.3 Analysis 16
3.2.4 Threat information platform (TIP) and Course of action generator (COA) 16
3.3 Domains and integration 16
3.3.1 Support for the Meta Attack Language (MAL) framework 16
3.3.2 Support for on-premise environments 17
3.3.3 Support for Amazon Web Services (AWS) 17
3.3.4 Support for Microsoft Azure 17
3.3.5 Integration 17
3.4 Deployment 18
3.4.1 Requirements and installation 18
3.4.2 Target domain 19
3.5 Current capabilities 19
3.6 Next steps 20

4 INTEGRATED PROTOTYPE OF IMC AND ADG 21

4.1 Introduction 21
4.2 Use Case 3 Demo: Response on Newly Discovered Vulnerable Assets 21
4.3 Use Case 4 Demo: Response on Discovered System Configuration Change 23

4.4 Use Case 5 Demo: Response on Deployment of New Systems in Infrastructure 24
5 ABBREVIATIONS 25

1 Introduction – Rationale of this document

This section introduces the SOCCRATES project and defines the goals of this deliverable.

1.1 The SOCCRATES project

SOCCRATES (SOC & CSIRT Response to Attacks & Threats based on attack defence graphs Evaluation Systems) is an EU funded project under the Horizon2020 programme that has the following main challenge:

How can SOC and CSIRT operations effectively improve their capability in detecting and managing response to complex cyber-attacks and emerging threats, in complex and continuously evolving ICT infrastructures while there is a shortage of qualified cybersecurity talent?

The main objective of SOCCRATES is to develop and implement a security automation and decision support platform (**'the SOCCRATES platform')** that will significantly improve an organisation's capability (usually implemented by a SOC and/or CSIRT) to quickly and effectively detect and respond to new cyber threats and ongoing attacks.

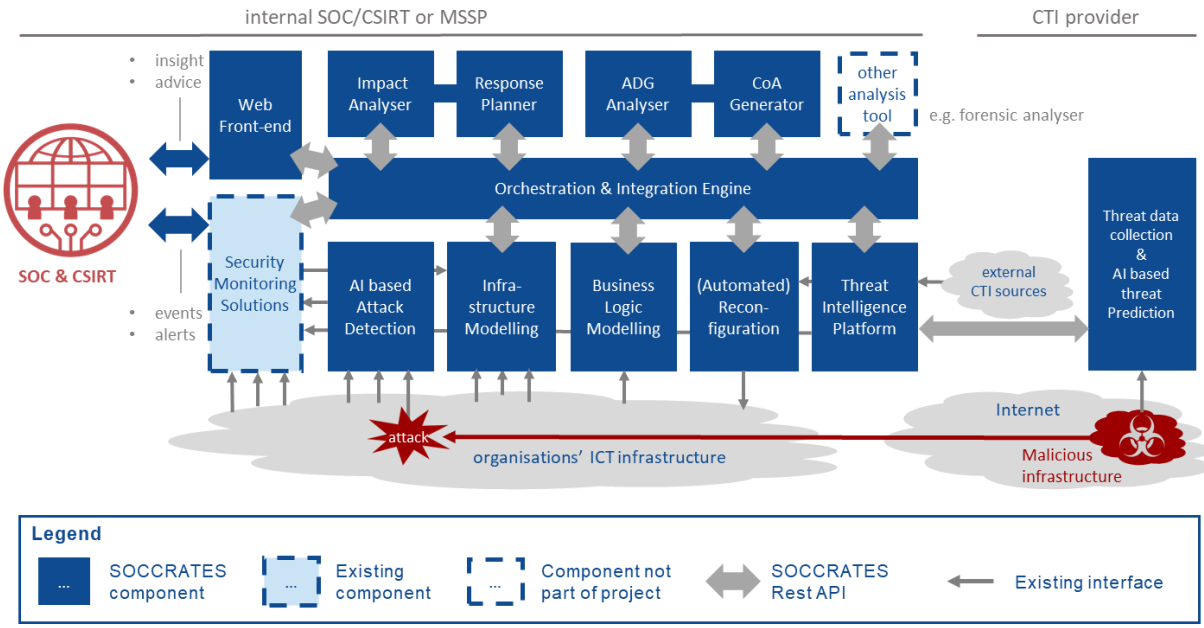

Figure 1 The SOCCRATES platform

The SOCCRATES platform (see Figure 1) consists of an orchestrating function and a set of innovative components for automated infrastructure modelling, attack detection, cyber threat intelligence utilization, threat trend prediction, and automated analysis using attack defence graphs and business impact modelling to aid human analysis and decision making on response actions and enable the execution of defensive actions at machine-speed.

SOCCRATES has the following concrete project objectives:

1. Deliver the SOCCRATES platform consisting of an orchestration function and a unique integration of innovative background solutions that seamlessly work together.
2. Show that the SOCCRATES platform can improve SOC operations by evaluating the SOCCRATES platform in two diverse real-life pilot environments.
3. Examine and illustrate the benefits of automation for selected SOC activities to help manage the cyber security skills gap in organizations.
4. Prepare for successful exploitation by the SOCCRATES partners of the individual innovated components and the integrated SOCCRATES platform in commercial products that are offered to the market and are available for the European (business) community.

Please visit www.soccrates.eu for more information on the SOCCRATES project.

1.2 This deliverable

The SOCCRATES platform consists of eleven components that are introduced in D2.2. Deliverable 3.1 is a demonstrator of the initial prototypes of two components: Infrastructure Modelling Component (IMC) and Attack Defence Graph analyser (ADG). This deliverable document describes the initial implementation and deployment of these prototypes. It is intended to serve as a guidance document to the reader.

IMC provides a model of an organisation's current ICT infrastructure in a common reference description language. The model allows SOC analysts to better understand system structures, configurations, and vulnerabilities and provides insight into how attacks propagate through their ICT infrastructure. An important goal of IMC is to provide accurate and up-to-date model of the ICT infrastructure to the other SOCCRATES components for further analysis. In WP3, however, it was observed that there is no commonly acknowledged ICT infrastructure modelling standard currently available that fulfils the needs of the SOCCRATES project. One such need is support for threat intelligence data. To address this problem, the project started to develop an appropriate reference (meta) model based on the needs of the SOCCRATES project. The new reference model adopts relevant portions of existing initiatives and standards in the cyber security industry. This deliverable introduces the IMC architecture, data model and deployment.

A role of ADG is to predict how attacks propagate over a given ICT infrastructure. ADG can be used to predict, for instance, the potential effect of a new threat, a new vulnerability, or a change in system configuration. ADG-based analysis can also be used to determine if and how the attacker may compromise other systems when an attack is detected. ADG utilizes data provided by the infrastructure modelling component (IMC) and performs an analysis based on probabilistic attack simulations that are further enriched with cyber threat information from a threat intelligence platform (TIP). As a result, ADG provides quantitative metrics on risk exposure and time to compromise (TTC), the most likely attack paths to high-value assets, and suggested mitigations. This deliverable introduces the characteristics and deployment of ADG as well its linkage with IMC.

1.3 Structure of this deliverable

This deliverable is structured as follows. Section 2 presents the initial prototype of IMC and Section 3 that of ADG. Section 4 provides the preliminary results of deployment and validation.

2 Infrastructure modelling

2.1 Introduction

The main objectives of the Infrastructure modelling component (IMC) are automated infrastructure scanning and successful generation of a correct model of the pilot infrastructure, which is defined as KPI1 in deliverable D7.1. Furthermore, IMC acts as a vulnerability scanning solution for SOCCRATES, providing information about detected vulnerabilities as a part of the infrastructure model for other SOCCRATES components to use.

2.2 System architecture

IMC and ADG are two of the three components that are developed in WP3 (see Figure 1). Figure 2 illustrates the current system architecture of IMC and its linkage to ADG that are used in Pilot 1. IMC creates a model that describes the current ICT infrastructure using a common reference description language (IMC graph database schema). The model is accessed via a REST API. To build this model, the component collects data from a variety of available IT and security management tools. Moreover, once Orchestration and Integration Engine (see D2.2) is available, IMC will trigger it in selected use cases, e.g. new assets discovered, changes in the configuration of an asset.

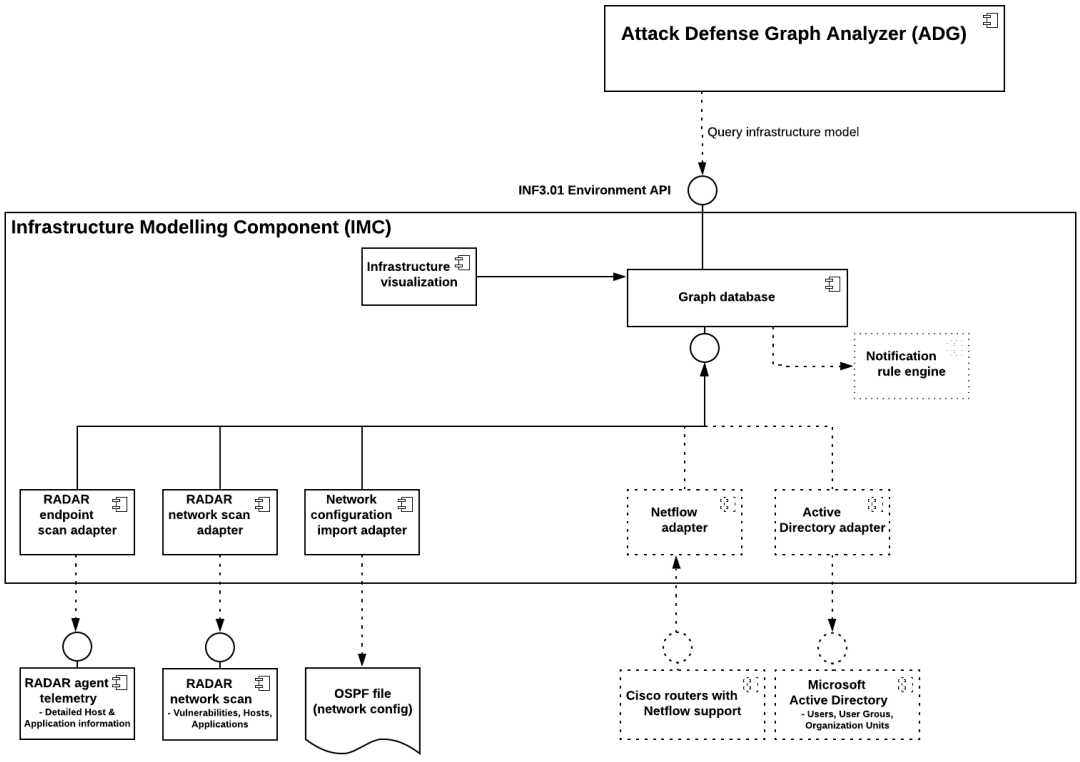

Figure 2. The current system architecture of IMC

Adapters and data sources used in the pilot 1 are:

- **RADAR network adapter**
  - Provides a comprehensive view to all hosts found from the configured target networks

- Collects Software versions visible with the network scan together with their known vulnerabilities, i.e. only running software with open ports is visible.

- **RADAR agent adapter** (requires SSH access to the hosts)
  - Provides detailed host information, e.g. host type and hardware information
  - Installed software and their known vulnerabilities, including software, which does not have any ports open.
- **Network configuration file importer**
  - Target network configuration file is imported with separate adapter, providing network routes, firewall rulesets and rules.

At the core of IMC there is a graph database, which stores the data in normalized data format. Figure 3 illustrates the objects used in pilot 1 and their relationships. All data is tagged with a unique organisation identifier to support multi-tenant deployment, but pilot 1 is using single-organization setup, i.e. all data is stored with the same organization identifier.

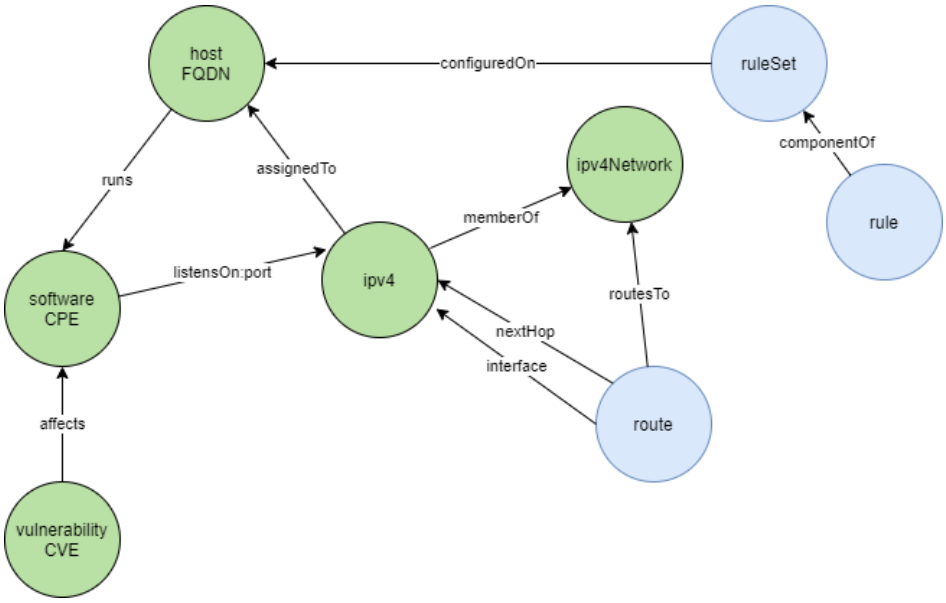

Figure 3. IMC graph database schema

Data objects are:

- **Host** is a device connected to the target organization's (IP) network:
  - Identified with FQDN and with related IPv4 addresses.
- **Software** is software product installed to one or more Hosts:
  - Identified with CPE when available, otherwise library name and version.
  - Software installed to a host may have open network port(s).
- **Vulnerability** is a known vulnerability:
  - Identified with CVE ID.
- **IPv4 Network** is a network segment within the target organisation's ICT infrastructure that has one or more Hosts connected to it:
  - LAN is identified with network segment, e.g. 192.168.0.0/24.
- **Route** is network route between IPv4 Networks
- **Firewall ruleset** limits the network traffic passed through the route

2.3 Deployment

This chapter describes the testbed environment, IMC components installed, and their configuration. The testbed is a virtual environment representing a company network with a set of virtual LANs, routers and virtual machines representing workstations in a company. Our initial testbed was comprised of networked Linux hosts. Future versions will add Microsoft Windows work-stations and an Active Directory infrastructure.

IMC and ADG are deployed in a hybrid mode, meaning that RADAR Cloud is used to manage RADAR scan nodes installed to the testbed network and similarly cloud version of the ACT platform is used as a core of IMC.

Figure 4 illustrates the testbed network monitored by IMC and the PCs deployed to it. Networks with blue colour illustrate LAN networks, red indicates routes between the LANs and green networks are used to administer the network and devices in it. IMC components and their configuration are described further in the chapters below.

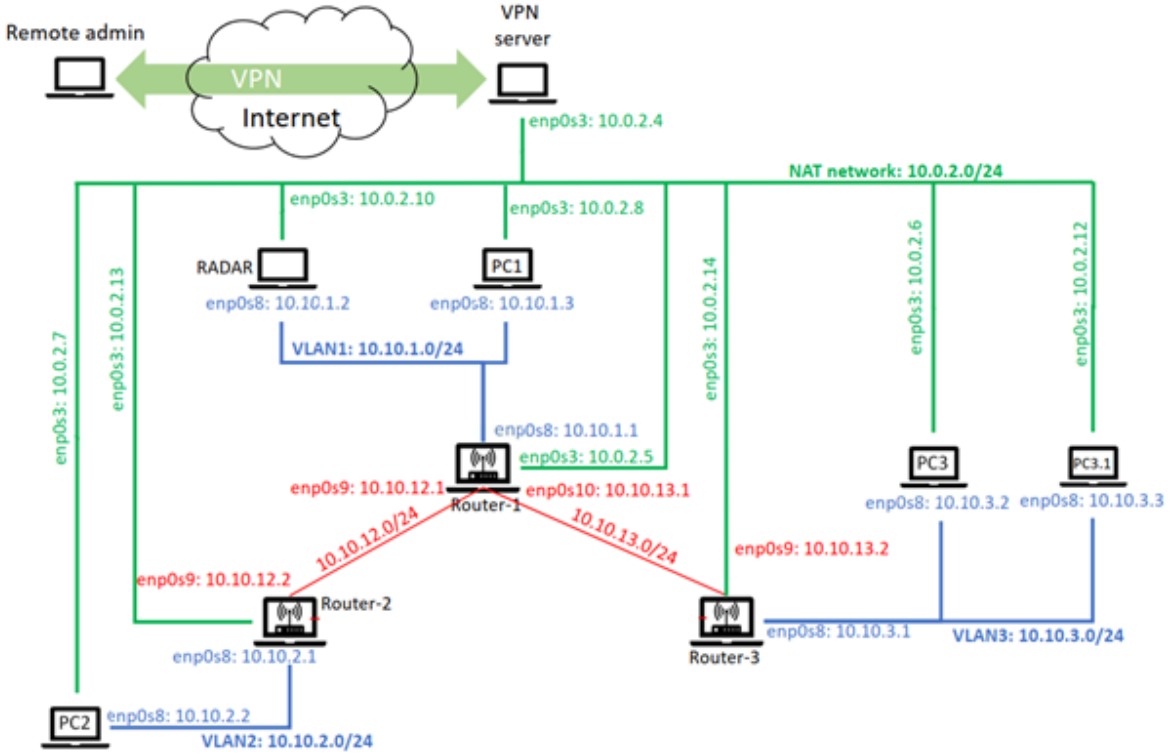

Figure 4. Testbed network and PCs

Linux user accounts with RADAR ssh public keys are needed in PC1, PC2 and PC3 for RADAR endpoint scans. Network routers and PC3.1 are out of scope of the endpoint scan and this enables the comparison of the scan results with and without endpoint scan.

2.3.1 RADAR Cloud Security Center

RADAR Security Center is running in multi-tenant F-Secure Cloud and licensed with normal RADAR Cloud licenses. It is configured to be used as a data source both for network and endpoint scans.

Configuration:

- Discovery scans are configured for all target networks
- Default TCP 0-65535, UDP 0-1024 template is used for network scan
- Settings / Encryption keys are created for storing authenticated scan credentials
- Template for authenticated scan for Linux/Windows endpoints (SSH / WinRM)
    - SSH credentials used in authenticated scan towards Linux devices are stored in template
    - WinRM is set to use scan node service account
- Scan Group is created for pilot 1 for devices without authenticated scan
    - System scan enabled
    - Using (Default) TCP 0-65535, UDP 0-1024 template is used for network scan
    - Asset update sources are set to use Discovery Scan as a source and triggering immediate scan for new hosts and new network services
    - Scan schedule is set to Daily
    - Web scans are disabled
- Scan Group is created for pilot 1 for devices with authenticated scan
    - System scan enabled
    - Using newly created template with SSH credentials
    - Asset update sources are set to use Discovery Scan as a source and triggering immediate scan for new hosts and new network services
    - Scan schedule is set to Daily
    - Web scans are disabled
- API Keys created for RADAR-IMC Adapter

RADAR does require some administration for it to be used as an IMC data source, as the target VLANs and high value assets need to be manually configured and maintained. VLAN1, VLAN2 and VLAN3 are configured to be target networks for the testbed. PC3, which represents the company's CEO's laptop is defined as having a critical confidentiality rating.

2.3.2 RADAR Scan node

A normal Linux RADAR scan node is installed to the RADAR server in the testbed and configured to be administered through RADAR Cloud following the installation guide available on F-Secure web pages: https://help.f-secure.com/product.html#business/radar/3.0/en/concept_AFCE4EB5D8434AEA9D6C79496D3609D4-3.0-en

2.3.3 RADAR IMC Adapter

RADAR-IMC adapter is installed on the same server as RADAR scan node together with Python and required pyyaml & act-api libraries. RADAR-IMC adapter is configured to use the RADAR Cloud Security Center and ACT cloud platform URLs with API credentials stored in the configuration file.

Source code and configuration file examples are stored in the SOCCRATES project gitlab: https://ci.tno.nl/gitlab/soccrates/wp3-adg-based-attack-analysis/-/tree/master/Task%203.1%20Enterprise%20ICT%20Infrastructure%20modelling/radar-adaptor

2.3.4 IMC Database and REST APIs

IMC graph database and IMC Environment API are deployed to a mnemonic's multi-tenant cloud environment and API credentials are created for RADAR-IMC adapter and ADG. IMC data model schema is used as a configuration for the database.

IMC graph database schema is stored in the SOCCRATES project gitlab: https://ci.tno.nl/gitlab/soccrates/wp3-adg-based-attack-analysis/-/tree/master/Task%203.1%20Enterprise%20ICT%20Infrastructure%20modelling/ACT-IMC/act-types-imc

2.4 Verification of IMC Capabilities

In this section we introduce the main capabilities of IMC and our approach to verify them. Verification was carried out in our testbed. Verification was necessary to ensure that the capabilities we developed behaved as planned. Full validation will be performed in pilot infrastructure at a later stage.

The main capabilities of IMC are:

- Automated infrastructure scanning
- Continuous updates
- Time dimension
- Extendable model
- Multi-organization

Verification of IMC's capabilities in our testbed environment resulted preliminary findings that are presented together with each of the capabilities.

2.4.1 Automated infrastructure scanning

The main way to test IMC's capability to automatically scan infrastructure was to empty the IMC database and execute full scans manually in order to cross-check if all assets in the testbed network were detected and imported into graph database correctly.

Result was that IMC detected all devices from VLANs 1-3 in our testbed, as expected. The RADAR scan node was scanned via localhost and was detected to have multiple IP addresses.

Another finding was that the software information available through network without authenticated endpoint scan was quite limited for attack path simulation purposes. Furthermore, in our test setup, RADAR was configured to scan on a daily basis, which may be too infrequent for real-world environments.

2.4.2 Continuous updates

IMC has the capability to normalize and update its model simultaneously from multiple data sources. This was tested in our environment by scanning some endpoints from the network and others from both the network and via authenticated endpoint scans.

Finding reliable logic to combine network and endpoint scan results turned out to be more complicated than expected. The challenge was that information available for host and software objects differ between

network and end-point scans. These differences were resolved by making some of the collected facts optional in the data model. Note that these differences need to be addressed by ADG and other components that source this data.

Additionally, the network routes and firewall rules were imported from a configuration file enriching the ipv4network information gathered from RADAR.

2.4.3 Time dimension

Time dimension refers to the ability to query the past IMC models via timestamp and this was tested by rescanning hosts after deleting them from the database, adding hosts into authenticated scan group one by one, and changing the criticality settings of the hosts.

No issues were found and the history of the changed objects and facts is also nicely visible through the IMC graph visualization.

2.4.4 Extendable model

The IMC data model was extended and changed a few times during the development phase, for example when changing the software cardinality rules and adding new facts for the software. Updating the IMC data model schema required coordinated schema changes to several components, e.g. ACT database, ACT UI, RADAR – IMC adapter and to ADG.

None of these changes required customer data to be deleted or migrated, making these kinds of changes much easier in customer environment. However, making schema changes in customer environments would require change management processes to be in place, otherwise the customer facing service would be broken until all necessary components are updated one by one.

2.4.5 Multi-organization

In testbed only one organization was used, but all IMC components are developed to support multi-tenancy as developing it afterwards is known to be difficult. All data was created under that one organization and all access grants were granted for that single organization only.

Fundamentally all current IMC components can be used in multi-tenant setup and the fact that the same end customer data could be fetched from ADG, using the ACT credentials scoped to testbed only, suggests that the basic support for multi-organization is already working.

2.5 Next steps

The next planned extensions to the IMC data model are shown in Figure 5 with yellow and red colours:

- Data flow (red)
    - Netflow adapter will provide network connection information between hosts based on Cisco Netflow protocol.
    - This will complement the static network route and firewall rule information defining what connections are possible in general with automatically gathered information of the actual network connections between the ipv4 addresses.

- Active Directory users, organization units and groups (yellow)
    - Active directory adapter will fetch user and user access related information from the target organization Microsoft Active Directory.

Furthermore, commercial F-Secure RADAR Cloud is planned to be enhanced with the ability to scan Windows devices using F-Secure endpoint agent, which will greatly simplify the deployment work needed to get authenticated endpoint scan results in RADAR and makes the basic host data more real-time. RADAR endpoint agent could improve the IMC data timeliness and simplify deployments with Windows devices.

This commercial product feature is based on similar logic as prototyped in IMC to combine network and endpoint scan results and for F-Secure customers, agent-based scanning complements traditional network-based vulnerability scanning with a more comprehensive view of vulnerabilities present in installed software.

Figure 5. Next planned IMC data model extensions

Classification level: Public

3 Attack Defence Graphs

3.1 Introduction

The Attack Defence Graph analyser (ADG) component can be used to predict how attacks propagate over a given ICT infrastructure. In SOCCRATES this is used to predict the potential effect of a new threat, a new vulnerability, a change in the system configuration, etc. In addition, when an attack is detected, ADG-based analysis can be used to determine if and how the attacker might compromise other systems. SOCCRATES will use this to provide the SOC / CSIRT analyst with detailed insight into what systems can be compromised, via which attack path(s) and how likely that is. The ADG component is based on Foreseeti AB's securiCAD solution for automated threat modelling and predictive attack simulations. In the ADG component role of the SOCCRATES platform, securiCAD will also be adapted to a continuous operational monitoring context.

3.2 Current system architecture

This section describes the high-level architecture of the current ADG component and its main components and functionality (see Figure 6). The ADG component is based on a service-oriented architecture where different services with specific responsibilities are communicating using REST APIs or via a Message broker service. The internal functionality can be divided into three main processes: Model generation, attack simulation and analysis.

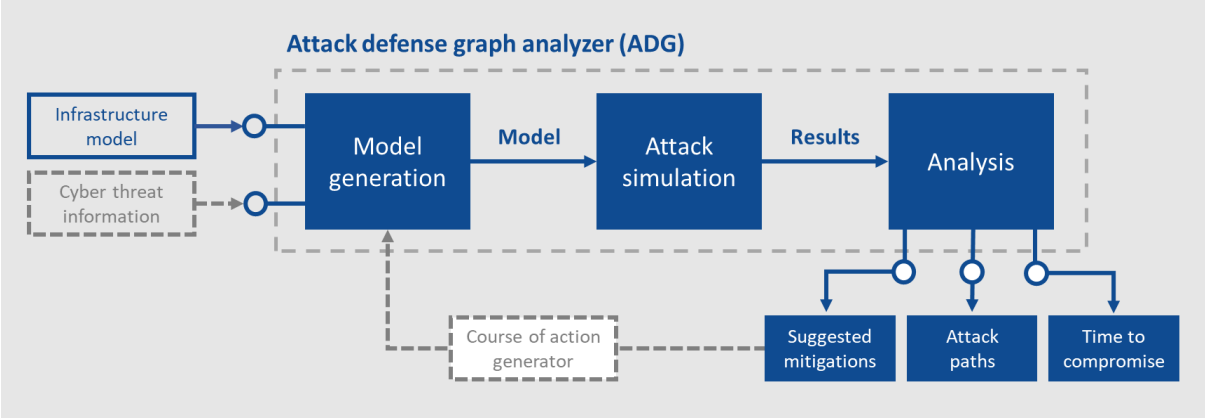

Figure 6. High-level architecture of the current ADG component

3.2.1 Model generation

The ADG component performs attack simulations on models of IT environments. These models can be generated automatically, created manually, or by using a combination of both. In the context of the SOCCRATES project, models are generated using IMC data. Models are created by adding assets and connecting them to each other to represent machines, users, networks, cloud services or any other type of asset that you typically find in your domain. The model definition, i.e., what asset types there are and how they relate to each other is defined in a domain specific language definition. This specification also contains the attack logic, i.e., how an attacker can behave and move in the model, e.g., by exploiting vulnerabilities, lateral movement through valid accounts, phishing of login credentials etc.

3.2.2 Attack simulation

The ADG component utilizes the securiCAD simulation engine to perform the attack simulation on the generated model. The attack simulation generates an attack graph based on the model with all possible steps and paths between the attacker's entry point and the high value assets in the model. Monte Carlo simulations are used to sample the attack graph and its statistical data on success rates and expected effort required to conduct attack steps. Shortest path calculations are then used to find the most likely paths through the graph.

3.2.3 Analysis

Based on the results of the simulation, the securiCAD engine will return the most likely attack paths to high value assets, the probability of success, the Time-To-Compromise (TTC) and suggested mitigations in the form of a web-based report or as JSON data.

3.2.4 Threat information platform (TIP) and Course of action generator (COA)

The current prototype does not integrate directly with the TIP or COA components. However, CTI can be used with the current prototype via manual configuration. The COA component currently uses securiCAD directly for conducting simulations based on suggested mitigations and is not using the ADG component yet.

3.3 Domains and integration

To be able to model and simulate modern IT architectures, one must acknowledge the fact that most IT architectures are actually hybrid environments. Many organizations migrate their datacentres and workloads to the cloud, but even if they run their own on-premise solution, many tools and services used are often hosted by other suppliers, effectively making it a hybrid environment. The securiCAD simulation engine only had built-in support for general on-premise environments with little or no opportunity to extend or customize the language definition.

3.3.1 Support for the Meta Attack Language (MAL) framework

A key capability to support additional domains and any modern hybrid environment is making the underlying language definition dynamic and customizable. Meta Attack Language (MAL) is an open-source framework for specifying assets and associations of models as well as the attack steps, defences and probabilities of the attack graphs. MAL specifications are compiled into a software package that can be used directly with the securiCAD simulation engine without any additional tweaks or modifications. By using and extending MAL, hybrid environments can be modelled and simulated in the ADG component.

New MAL specifications for specific domains such as cloud environments can be combined with specifications for "traditional" IT architectures to facilitate cross-domain analysis. Having support for MAL has allowed the modelling and attack simulation logic of different domains to be decoupled from ADG, allowing domain-specific language to be developed independently from, and dynamically loaded into ADG. In SOCCRATES, the scope of the ADG component ranges across both cloud environments and on-

premise infrastructures, but the focus is on on-premise environments as this is the target of both project pilots. The currently supported cloud service providers are Amazon Web Services (AWS) and Microsoft Azure.

3.3.2 Support for on-premise environments

On-premise ICT environments are the target domain of the SOCCRATES project, and this is in many ways a much more challenging target than a cloud environment, since there is no single consistent and homogenous data source from which such environments can be modelled. This is exactly what the Infrastructure Modelling Component (IMC) solves, and ADG is using the IMC as a single data source for continuously running the modelling – attack simulation – analysis loop to support SOCCRATES use cases. coreLang (https://github.com/mal-lang/coreLang) [2] is an open-source initiative to create a MAL-based threat modelling and attack simulation language based on few but powerful abstractions that allow flexible representation of many different environments. coreLang was chosen for the ADG component over the securiCAD-native securiLang due to its superior fit to the SOCCRATES cyber operational context in on-premise environments. In particular, coreLang has explicit support for software vulnerabilities, including automatic mapping from CVSS v2/v3 vectors to probability functions in the attack graphs and support for more direct and intuitive modelling of supply chain attacks.

3.3.3 Support for Amazon Web Services (AWS)

The ADG component is also capable of generating models and running attack simulations of AWS environments. This includes a MAL-based language specification that defines the AWS attack logic as well as a parser that can automatically build infrastructure models from standard AWS APIs. AWS is a comprehensive platform with a large set of functions and services and the AWS support package currently covers the most important set of core services: IAM, KMS, EC2, VPC, RDS, S3, DynamoDB, Lambda, API Gateway, ECS, ECR, Glue, DataPipeline, SageMaker, CloudFormation, CodeStar and Amazon Inspector.

3.3.4 Support for Microsoft Azure

The ADG component supports Microsoft Azure cloud environments in the same manner as it supports AWS. The Azure features currently supported by ADG are: RBAC (IAM - via Management Groups, Subscriptions and Resource Groups), Virtual Machines, Virtual Networks, Network Security Groups, Key Vault, Cosmos DB, App service Apps, Function Apps and Service Bus.

3.3.5 Integration

In order to achieve continuous and near real-time modelling and analysis of the environments where the SOCCRATES platform is deployed, the ADG component must also be able to automatically and efficiently generate models from IMC data or other data sources for any target domain. These integration points between environment data and the ADG component are called parsers. Parsers effectively transform data collected from the target environment into the correct language specification format.

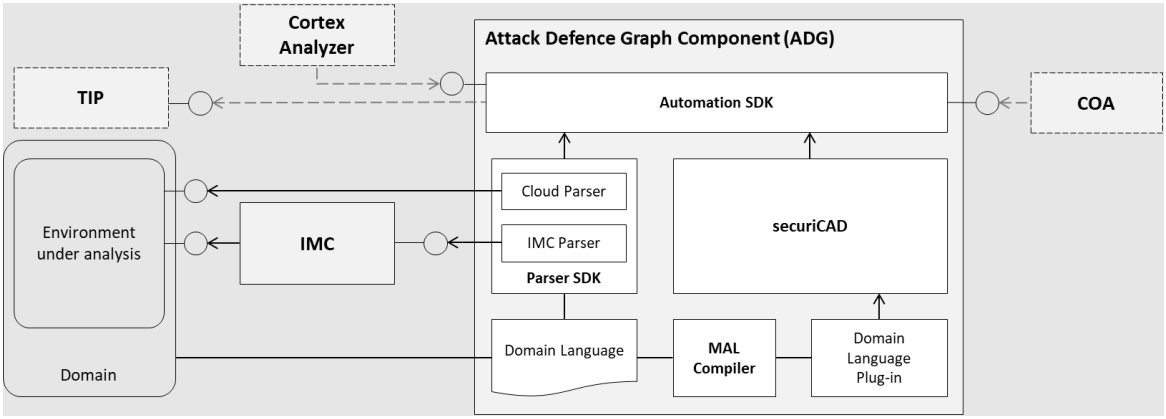

Figure 7. Domain support and auto-modelling

In Figure 7 above, **Domain** denotes a particular type of environment which the ADG component is targeted to support. Selection of the environment is in theory arbitrary, but in practice chosen by aspects of cohesion as the domain is to be formally represented in a MAL based language and models auto-generated through a parser. There are currently two types of parser; The **IMC Parser** which parses data from the **IMC** and **Cloud Parsers** which parses data directly from cloud service providers.

- The connection between **Domain** and **Domain Language** denotes the representation of a particular type of environment in a formal Domain language specified in the MAL framework
- The connections between **Domain Language**, **MAL compiler** and **Domain language plug-in** illustrates that the domain language is compiled into an securiCAD "plug-in" i.e., the simulation and modelling logic which is used for the attack simulation.
- The **Domain Language plug-in** and **ADG platform** connections show that the plug-in is loaded by securiCAD to provide the simulation functionality for that domain.
- **Parsers** are linked to a domain languages and are written to utilize data sources describing a real environment of a particular domain. The source data is transformed into a model that can be used simulated by securiCAD in the ADG component. Parsers are written with a standard **Parser SDK** to help creating tailored parsers for different domains.
- The **Automation SDK** component illustrates the possibility to automate both the modelling process as well as the simulation and analysis process of the ADG component. This is utilized to integrate the ADG component in the larger SOCCRATES system context and illustrated with the connections to the TIP, IMC, Cortex analysers of the Orchestrator and the COA.

3.4 Deployment

The ADG component is a multi-user and multi-tenant web-based platform. An ADG deployment includes a centralized server instance holding the project and user databases. System services and other components are accessed via a browser, APIs, or the SDK. The ADG component can be installed directly on dedicated hardware or in a VM with a supported guest OS and can be deployed on most modern servers, workstations or laptops.

3.4.1 Requirements and installation

The ADG component requires a machine (dedicated hardware or virtual) with at least 8GB RAM and 30GB of storage, and currently supports the following operating systems: Ubuntu Server 16.04/18.04/20.04, RHEL 7 and CentOS 7. The installation requires a dedicated account with sudo privileges. By default the installer connects to the Internet to download needed packages. If required,

foreseeti can provide installations with all packages included where Internet access is not available or allowed.

3.4.2 Target domain

The ADG component currently requires that the intended target domain is selected before deployment so the required language specification can be installed i.e., if the environment that is being analysed is deployed in AWS, Azure or on-premise. Support for deployments for hybrid-environments will be added.

3.5 Current capabilities

In this section we introduce the current capabilities of the ADG component and our approach to verifying them. Verification was carried out in the SOCCRATES testbed. Cloud environments were used to verify AWS and Azure integration functionality. The functions provided by the current ADG component prototype are:

- Perform probabilistic attack simulation on a given infrastructure model provided by the IMC and supported cloud service providers
  - perform attack simulations for a particular set of high-value assets and specific starting point and capabilities of the simulated attacker (Use Case 1)
  - assess the probability of a successful attack with regards to specific security measures and model configurations
  - find the most probable attack paths to high-value assets in the model.
- Quantify and provide a metric based on the most probable attack path to each of the high-value assets
  - Time to compromise (TTC) i.e., the time to a successful attack, will be provided for each high value asset
- Compare and quantify the difference between simulation results with regards to the aforementioned metric. Difference with previous analysis could be due to:
  - Newly discovered vulnerable assets from the IMC (Use Case 3)
  - Change in the infrastructure model from the IMC (Use Case 4 and 5)

The above capabilities performed as expected and delivered the expected data to the SOCCRATES testbed environment as well as tested cloud environments. However, there were some expected challenges for more complex environments where not all data points are as neatly curated as in the testbed:

- The ADG component relies on a statistical simulation engine to calculate the most likely attack paths. These attack paths' likelihood often depend on security controls or configuration points in the environment e.g., the existence of a vulnerability scanner or 2FA on a account. If these data points are not available in the IMC, the ADG component makes assumptions about the model. The ADG component must be able to communicate these assumptions together with the results as well as giving the recipient the opportunity to change them.
- Changes to the underlying simulation logic can greatly affect results and create large changes in the quantified results as they are compared over time. These changes must be communicated by the ADG component to the recipient to note that changes have been made to the simulation logic and that they can be the reason behind changes in the results.

3.6 Next steps

The functions currently not yet provided by the ADG component are:

- Integration with the TIP and COA components
  - Containment CoA generation (Use Case 1)
- Integration with the orchestrator via a Cortex analyzer
- Compare and quantify the difference between simulation results with regards to:
  - New threat information from the TIP (Use Case 2)
- Update or add new attack steps to the attack simulations based on, for example, increased use of a particular attack technique (Use Case 2.1.1)
- Update the parameters of vulnerabilities in the model based on new information about the vulnerability. For example, the availability of public exploits or released patches (Use Case 2.1.2)
- Define a threat actor with specific set of preferred attack techniques or capabilities (based on Adversary Emulation Plan) (Use Case 2.1.3)

Known improvements include:

- Extended language definitions to keep up with the IMC development and cloud service additions
- Better performance for more efficient sampling and graph generation as well as improved data ingestion and model generation to be able to deliver models and simulations in near real-time
- Support for hybrid domains and multiple simultaneous language definitions in the ADG component

4 Integrated prototype of IMC and ADG

4.1 Introduction

The purpose of the integrated prototype is to verify that the initial data provided by IMC can be used by ADG to make attack path simulations and to demonstrate how IMC and ADG work together.

The use cases 3-5 were chosen to demonstrate the integration of ADG and IMC as these could be tested before the Orchestration and Integration Engine is available by manually triggering the ADG analysis.

4.2 Use Case 3 Demo: Response on Newly Discovered Vulnerable Assets

This use case was used in the IMC/ADG webinar as a storyline, where a new vulnerability is found from the testbed network by RADAR, which updates IMC making it available for ADG, and finally ADG is triggered manually to use the IMC data in its simulations. This process flow is illustrated in the Figure 8. Orchestration and Integration Engine, Business Impact Analyzer (BIA) and Course of Action generator (COA) will be added in a later stage of the project, adding more components to the overall flow.

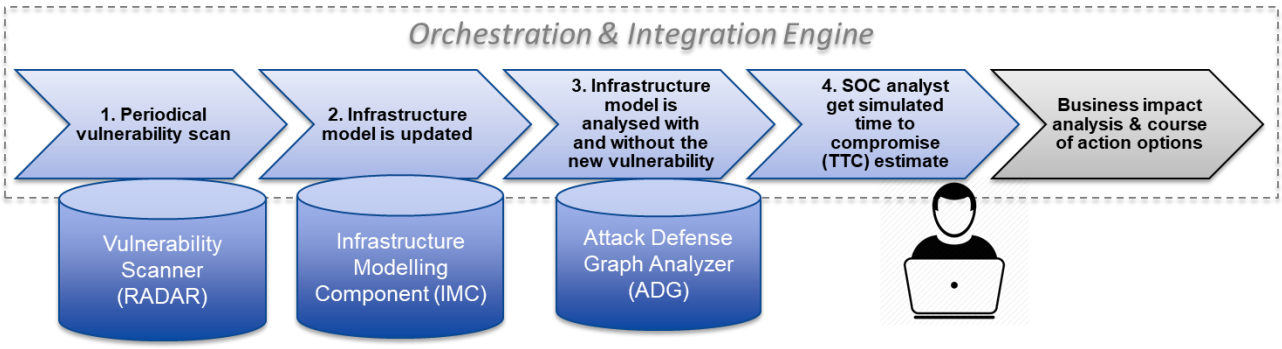

Figure 8. Process flow of the Use Case 3 without BIA & COA

Initially the testbed virtual machines did not have any critical vulnerabilities, which could have been used and thus a set of vulnerable software was installed for testing purposes as illustrated in Figure 9.

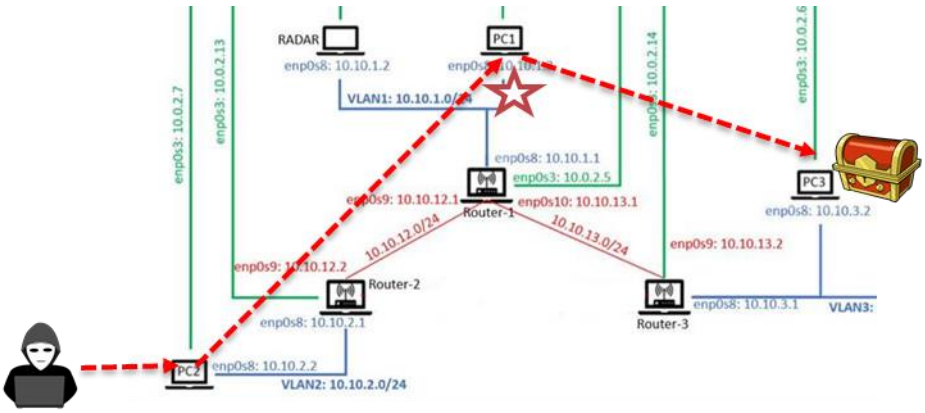

Figure 9. Attack path simulation after new vulnerability found from PC1

- PC1 represents “IT Admin PC” with a newly discovered vulnerability in mod_perl
- PC2 represents “Web server” with unpatched Apache Web server with multiple vulnerabilities
- PC3 represents “CEO laptop” being the most critical asset in the testbed network with unpatched samba

This combination of vulnerabilities enabled the new attack paths to be simulated with ADG. This process flow is illustrated in Figure 10, Figure 11, Figure 12 and Figure 13 that are screenshots showing how testbed looks in RADAR, IMC and ADG after the new vulnerability was found from PC1.

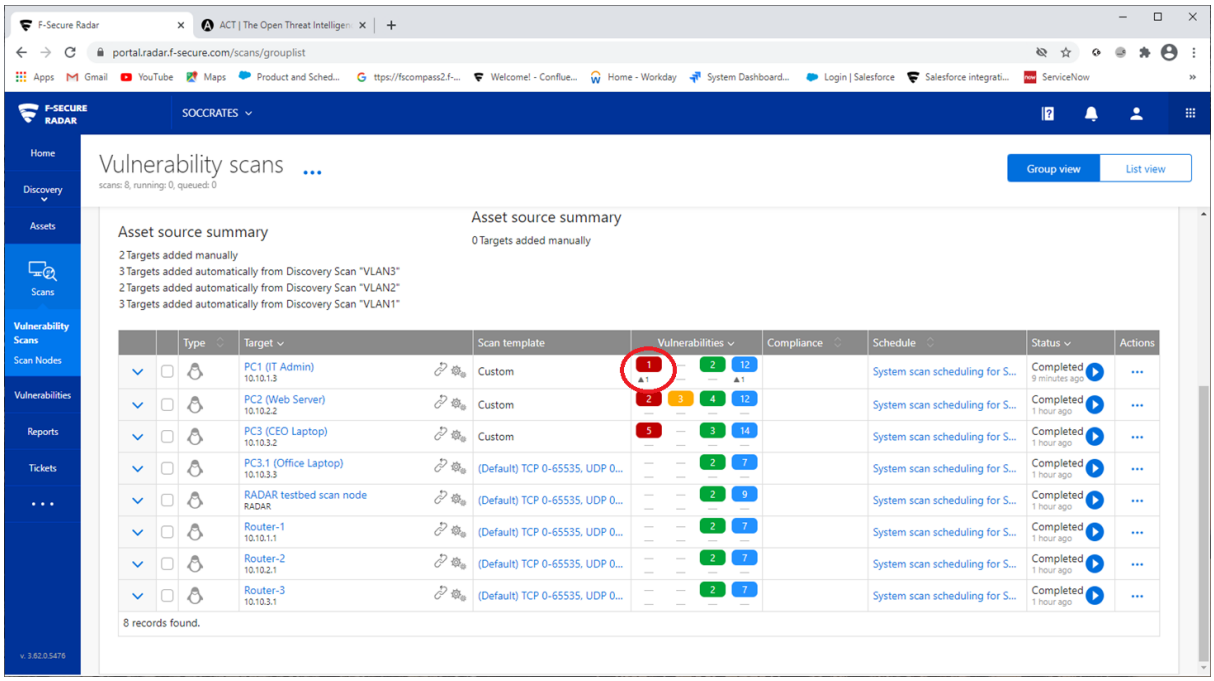

Figure 10. Scan results after the identification of the new mod_perl vulnerability in PC1

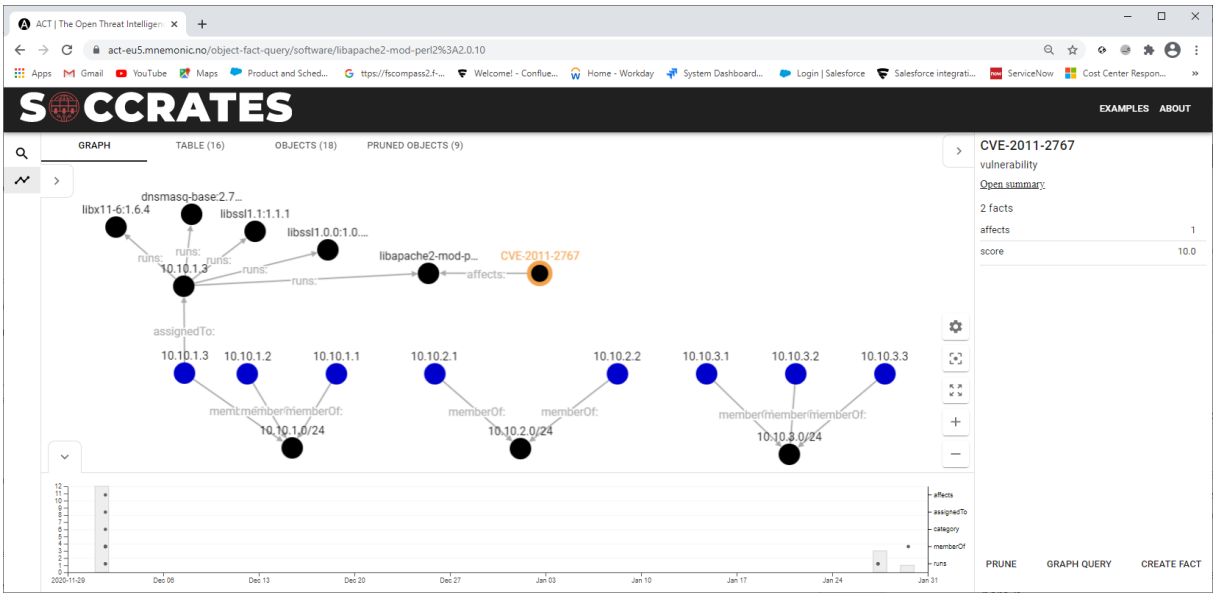

Figure 11. IMC visualization of the testbed network and the new vulnerability CVE-2011-2767 in PC1

The above screenshot shows the IMC internal structure for networks, IP addresses, hosts, software and vulnerabilities. In the current version, only software with known vulnerabilities is imported into

the IMC graph database as there is no use for the software listing of tens or hundreds of installed libraries without any vulnerabilities. The right-side window shows the facts linked to the selected object, e.g., the new vulnerability CVE-2011-2767 and the bottom part shows the timeline when the facts have been inserted to the IMC database.

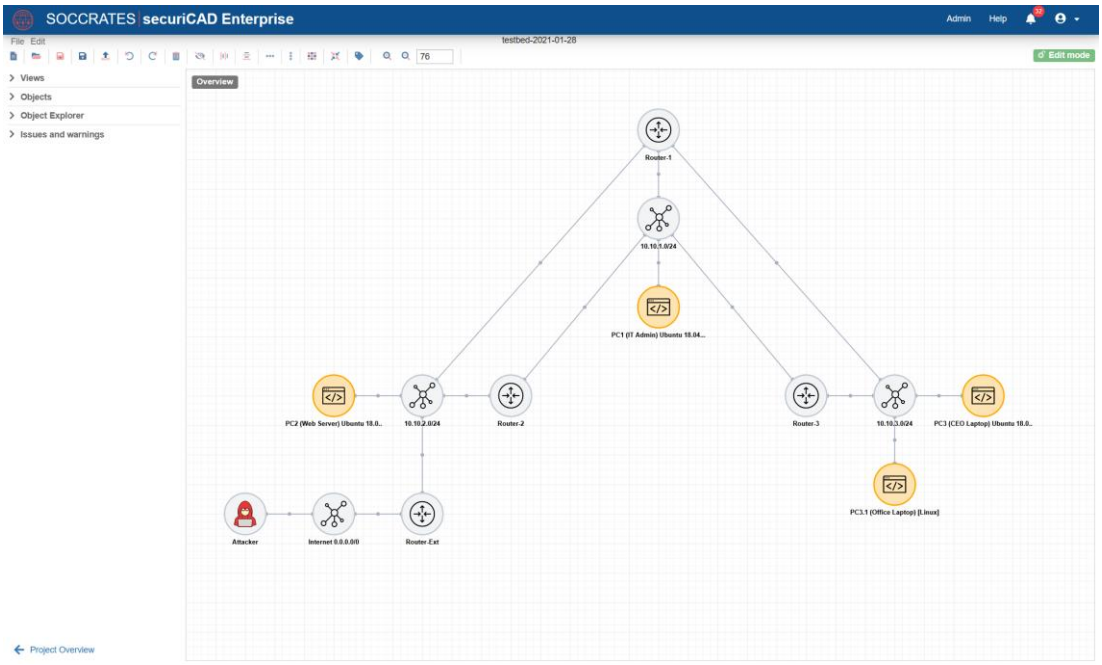

Figure 12. Testbed infrastructure as seen in ADG

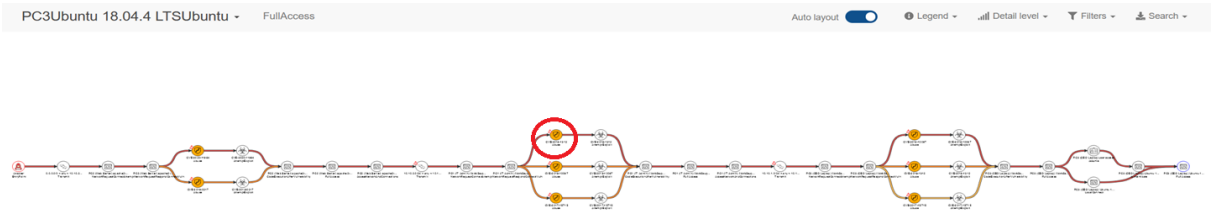

Figure 13. Most critical attack paths for testbed based on the simulation

Figure 12 above shows the ADG internal structure after the infrastructure model has been loaded from IMC and the newly found vulnerability is part of the simulated attack paths as shown in Figure 13. The inclusion of an exploit of a new vulnerability in such a path means that the system has found increased risk to business-critical assets as a direct consequence of the vulnerability.

4.3 Use Case 4 Demo: Response on Discovered System Configuration Change

Router configuration changes were used to verify that configuration changes in testbed were passed from IMC to ADG. The original testbed setup allowed network traffic between Router 1 and Router 3, but this route was blocked as illustrated in Figure 14, and the new network configuration file was loaded into IMC.

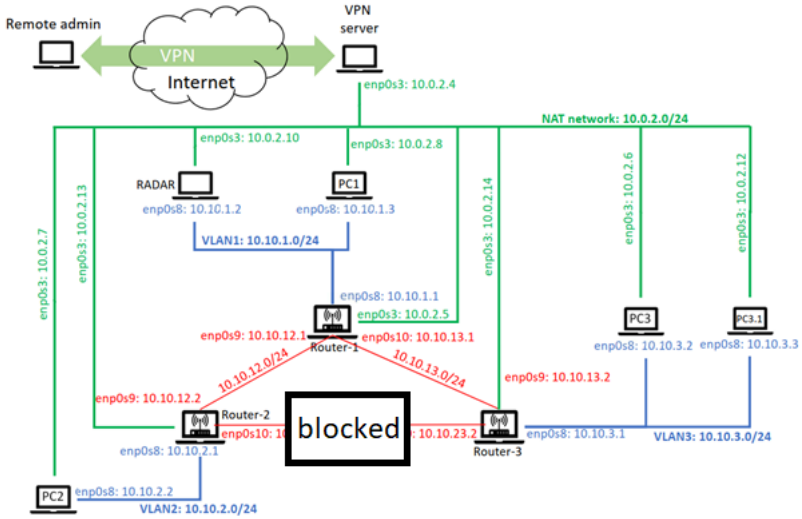

Figure 14. Network configuration change

This change made the testbed network more interesting for attack path simulation as previously network access within the network was allowed between all PCs. Blocking network connections between PC2 and PC1 requires the attacker to gain access to PC1 for lateral movement.

4.4 Use Case 5 Demo: Response on Deployment of New Systems in Infrastructure

To test how comprehensively IMC can detect new assets in the testbed, the IMC testbed database was cleaned up and the RADAR scans executed manually to detect all assets from testbed VLANs 1, 2 and 3.

Figure 15 shows that IMC was able to identify all eight IP addresses and related hosts from the target VLANs, plus one additional IP address, 172.17.0.1 used by the RADAR scan node server itself. The RADAR scan node server (10.10.1.2) had this additional interface and as a RADAR scan was made locally, it was able to identify this IP address in addition to the VLAN1 address 10.10.1.2.

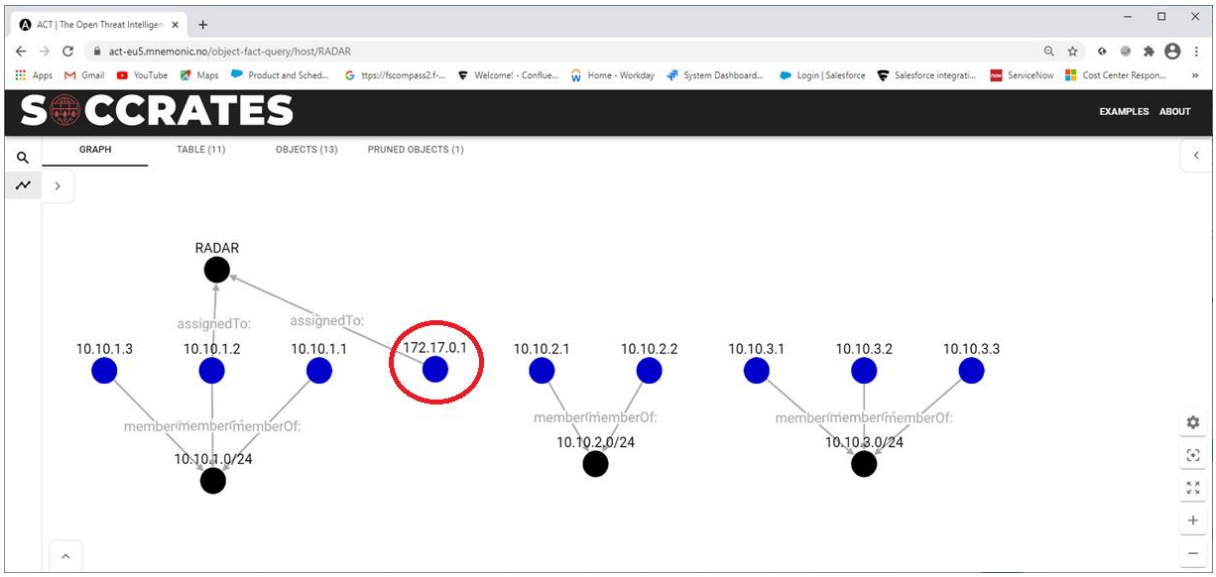

Figure 15. ipv4networks and ipv4 addresses of testbed in IMC graph database

Classification level: Public

5 Abbreviations

This glossary serves as inventory of abbreviations used in the document.

Acronym	Description
ACT	semi-Automated Cyber Threat intelligence
ADG	Attack Defence Graph
AEF	Argus Event Format
AI	Artificial Intelligence
AIT	AIT Austrian Institute of technology
API	Application Programming Interface
APT	Advance Persistent Threat
ATOS	ATOS Spain
AV	AntiVirus
AWS	Amazon Web Services
BPMN	Business Process Model and Notation
CC	Command and Control
CERT	Computer Emergency Response Team
CMDB	Configuration Management Database
CSIRT	Computer Security Incident Response Team
CoA	Course of Action
COA	Course of Action generator
CTI	Cyber Threat Intelligence
CVE	Common Vulnerabilities and Exposures
DC	DataCentre
DGA	Domain Generated Algorithm
DNS	Domain Name System
EDR	Endpoint Detection and Response
ELK	Elasticsearch/Logstash/Kibana
FRS	Foreseeti
FSC	F-secure
ICT	Information and Communication Technology
IDS	Intrusion Detection System
IMC	Infrastructure Modelling Component
IMT	Institut Mines Télécom – Télécom SudParis
INTF	Interface
IoC	Indicators of Compromise
IP	Internet Protocol
IPS	Intrusion Prevention System
IRM	Incident Response and Management
ITIL	Information Technology Infrastructure Library
KTH	Kungliga Tekniska högskolan - Royal Institute of Technology
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MAL	Meta Attack Language
ML	Machine Learning
M_n	Infrastructure Model (at time *n*)

MNM	mnemonic
MSSP	Managed Security Service Provider
MTTD	Mean Time To Detection
NOC	Network Operations Centre
OT	Operational Technology
OS	Operating System
RORI	Return on Response Investment
SDN	Software Defined Network
SHS	Shadowserver
SIEM	Security information and event management
SOAR	Security Orchestration, Automation and Response
SOC	Security Operation Centre
SOCCRATES	SOC & CSIRT Response to Attacks & Threats based on attack defence graph Evaluation Systems
SSH	Secure Shell, a cryptographic network protocol
SSL	Secure Sockets Layer
TAP	Test Access Point
TIP	Threat Intelligence platform
TLS	Transport Layer Security
TNO	Nederlandse Organisatie voor toegepast natuurwetenschappelijk onderzoek
TTC	Time To Compromise
UC	Use Case
VLAN	Virtual LAN
VM	Virtual Machine
VTF	Vattenfall