



INTERNATIONAL
HELLENIC
UNIVERSITY

Smart grid simulation platforms with cyber- attack capabilities

Eleftheriou Alexandros

SID: 3307150002

SCHOOL OF SCIENCE & TECHNOLOGY
A thesis submitted for the degree of
Master of Science (MSc) in Communications & Cybersecurity

MARCH 2018
THESSALONIKI – GREECE



INTERNATIONAL
HELLENIC
UNIVERSITY

Smart grid simulation platforms with cyber- attack capabilities

Eleftheriou Alexandros

SID: 3307150002

Supervisor:
Supervising
Members:

Committee Prof. Sokratis Katsikas
Assoc. Prof. Name Surname
Assist. Prof. Name Surname

SCHOOL OF SCIENCE & TECHNOLOGY
A thesis submitted for the degree of
Master of Science (MSc) in Communications & Cybersecurity

MARCH 2018
THESSALONIKI – GREECE

Abstract

Smart grid, the next generation power grid, uses two-way electricity and data flow by integrating information and communication (ICT) infrastructure into its legacy system. This allows for a self-monitoring, self-healing, adaptive power grid environment that encourages distributed generation methods and user active participation. Due to the huge network complexity, combined with the benefits that smart grid features offer, there is a need for simulation environments that allow us to understand different aspects of the smart grid ecosystem. The most in-depth attempts use co-simulation frameworks combining both legacy and communication networks.

However, due to the critical nature of the availability of power related services, this massive network of millions interconnected devices creates security concerns and vulnerabilities. This thesis comprehends a description of the smart grid infrastructure and the main simulation platforms that are being used to model its complex environment. Emphasis is given to security related issues, by providing an overview of the different cyber-attack modeling techniques that are used to understand and confront system and network vulnerabilities and threats. Finally, an analysis of both simulation and emulation platforms that attempted to model attacks against parts of a smart grid network, is been given.

Acknowledgements: I would like to thank my supervisor Sokratis Katsikas for his continuous guidance and advice throughout the thesis development.

Eleftheriou Alexandros
March 2018

Contents

ABSTRACT	III
CONTENTS	V
1 INTRODUCTION.....	1
2 SMART GRID OVERVIEW	3
2.1 SMART GRID MAIN CONCEPTS.....	3
2.2 SMART GRID ARCHITECTURE	3
2.2.1 <i>Generation</i>	4
2.2.2 <i>Transmission</i>	4
2.2.3 <i>Distribution</i>	5
2.3 DCS AND SCADA SYSTEM OVERVIEW	5
2.4 COMMUNICATION NETWORKS IN SMART GRID	7
2.4.1 <i>End user/Customer</i>	7
2.4.2 <i>Distribution grid</i>	8
2.4.3 <i>Cellular networks</i>	8
2.4.4 <i>Transmission grid</i>	9
3 SECURITY ISSUES IN SMART GRID.....	11
3.1 SMART GRID VULNERABILITIES/ATTACKS.....	11
3.1.1 <i>SCADA/EMS</i>	11
3.1.2 <i>AMI components</i>	12
3.2 INCIDENTS	15
4 SIMULATION.....	17
4.1 CATEGORIES	17
4.2 POWER GRID SIMULATORS.....	18
4.2.1 <i>MATPOWER/PYPOWER</i>	18
4.2.2 <i>OpenDSS</i>	18
4.2.3 <i>GridLAB-D</i>	19
4.3 NETWORK SIMULATORS	19
4.3.1 <i>OMNeT++</i>	19
4.3.2 <i>Ns-3</i>	19
4.4 CO-SIMULATION APPROACH.....	20
4.4.1 <i>OpenDSS/OMNeT++ co-simulation</i>	20
4.4.2 <i>FNCS framework</i>	22
4.4.3 <i>Mosaik</i>	23
4.4.4 <i>NeSS²</i>	24
4.5 EMULATION APPROACH.....	26
4.5.1 <i>CORE</i>	26
4.5.2 <i>GNS3</i>	27
5 CYBER-ATTACK MODELING TECHNIQUES.....	29

5.1	ATTACK GRAPHS/TREES.....	29
5.2	ATTACK SURFACE/VECTOR	31
5.3	OWASP AND THREAT MODELING TECHNIQUES.....	32
5.4	DIAMOND MODEL	33
5.5	KILL CHAIN MODEL	35
5.6	COMBINING MODELS.....	37
5.7	ATTACK MODELS ON CPS.....	37
5.8	ATTACK MODELS ON SMART GRID	38
6	ANALYSIS OF CYBER-ATTACK MODELING IN SMART GRID RELATED FRAMEWORKS	41
6.1	ASTORIA	41
6.1.1	<i>Mosaik overview</i>	41
6.1.2	<i>NS-3</i>	44
6.1.3	<i>ASTORIA Co-simulation</i>	46
6.1.4	<i>Attacks</i>	47
6.2	NESSI ²	48
6.3	SCORE.....	50
6.3.1	<i>Overview</i>	50
6.3.2	<i>Attacks on a SCORE emulator</i>	51
6.4	GNS3.....	55
6.4.1	<i>Modeling approach</i>	55
6.4.2	<i>DoS with GNS3</i>	56
6.4.3	<i>Hexinject</i>	59
6.5	SUMMARY	61
7	CONCLUSIONS	63
	BIBLIOGRAPHY	65
	APPENDIX	71

1 Introduction

Smart grid is a market term used to describe the act of modernization and evolution of the electrical power grid infrastructure, which did not experience major changes during the last century. This modernization process was achieved with the integration of Information and Communication Technology (ICT) components on top of the conventional power grid, that connect field devices, utilities, operators and end users. This intelligence layer enables full real-time view and remote control of the grid infrastructure and applications to emerge.

The legacy power grid system is isolated and based on large central generation plants, transmission and distribution centers that are one-way and do not efficiently inform end users and customers about their electricity consumption. Generation is highly dependent on conventional resources (thermal, nuclear, chemical) a concerning issue, mainly due to the environmental impact of burning fossil fuels and the waste of almost two-thirds of fueled energy. Equipment is checked and restored manually, and the protection of the systems is limited. Higher demands, lack of effective monitor and control of the subsystems, lack of capacity to meet future demands and the inability to support decentralized energy resources together with the technological advances in communication systems are the reasons that lead to modernization opportunities in the power grid infrastructure.

Smart grid is the result of an effort to build a more robust, self-healing, more automated and human less, environmental friendly power grid infrastructure. There has been a dramatic change in how energy is generated, transmitted and consumed emphasizing in balance between generation and demands as well as moving from central to widely distributed generation concepts. The increasing demand of energy and the protection of electrical devices requires complete control of substations, in order to create a reliable system that can adapt to and withstand failures. ICT systems can achieve full monitoring and control of the power grid ecosystem by offering two-way communication link possibilities.

This thesis contributes to the cyber-attack modeling domain in smart grid related scenarios. An analysis of the most notable co-simulation and emulation efforts is being given, along with custom scenario variations and possible attack extensions. The scenario structures follow known cyber-attack modeling technique patterns for a better understanding of their impacts.

The structure of this thesis goes as follows: section 2 provides a detailed overview of the smart grid ecosystem, by describing its architecture and main concepts, its underlying control system, the communication networks and technologies that are integrated and how all these features interact. In section 3, the main security issues of smart grid are discussed, along with a brief reference to the main related security incidents. Issues concern both the control system and the advanced metering infrastructure. Section 4 describes the ways smart grid environments can be modeled.

Emphasis is given to co-simulation frameworks that use a combination of power and communication network model approach and emulation testbeds that mimic behavior of real devices and can be integrated into real networks. An overview of both simulators and emulators used in smart grid modeling scenarios is also being given. Section 5 includes a descriptive report on the cyber-attack modeling techniques used in ICT, CPS and in smart grid environments. Choosing the most appropriate techniques can help in better modeling, understanding attacks and mitigating risks and threats. Finally, in section 6 an analysis of simulators and emulators used in smart grid scenarios is given, their attack modeling techniques and a brief description of tools used, aiming at finding the most appropriate techniques, compatible with a smart grid scenario.

2 Smart grid overview

2.1 Smart grid main concepts

The core of smart grid [7] is the combination of communication and power grid networks for increased awareness of the system which is now interconnected, a more efficient use of current assets as well as applications that can prove of great importance to transmission, distribution, generation and end user areas.

Smart grid enables the distributed control and monitor of the systems along with the existing central oriented one, that can improve utility services quality.

Supports intelligent automated delivery of energy and error correction minimizing outage possibilities and energy loss that can reduce the amount of work of system operators.

It is also capable of facilitating generation of all sizes, alternative sources of energy and microgrids that can run independently a huge advantage to power management and environmental impact. Energy can be now stored, reused and fed back to the grid.

End users can now be a part of the system optimization by using real time information that is given to them through smart meters and smart appliances, allowing them to manage their electricity consumption efficiently and manually by plug and play services.

Concluding, the main goal of smart grid is a self-healing power grid system that can support energy and demand side management efficiency. The previous, constitute the idea of advanced metering infrastructure (AMI) which can be further defined as real time automated metering and meter data management using Information and Communication Technologies (ICT) equipment and computational intelligence to process and distribute information accordingly.

2.2 Smart grid architecture

For a better grasp of the smart grid infrastructure several conceptual models [1] have been provided. These can be area, technical or utility-oriented models. NIST [3], [5] divides the smart grid into seven domains: power generation, transmission, distribution and customers, markets, operators and service providers.

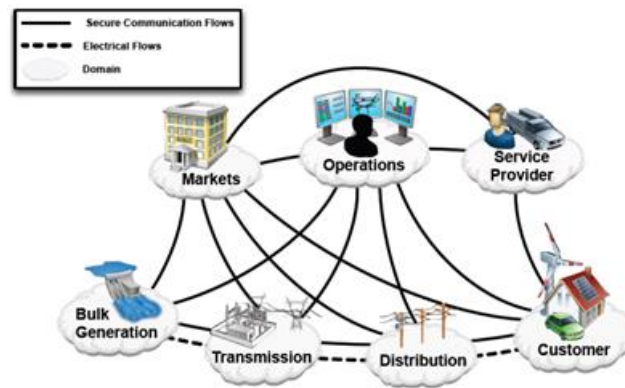


Figure 1 NIST smart grid model, extracted from [3]

2.2.1 Generation

Power plants where electricity is generated through other forms of energy before fed into the grid. The main change in this domain is that the two-way flow of electricity allows for more efficient and environmental friendly power generation using multiple layers of distributed intelligent generation (DG). Using distributed energy resources (DER) that are often based on wind and solar energy we can now move on from the conventional centralized fossil-based generation to small scale distributed generation from alternative energy resources. This concept allows for many independent user-based generators that can stand on their own or act together as a conventional scalable power plant that can be managed remotely. Smart generation also allows for energy storage, control and asset management, a solution to peak availability and demand problems. Despite the many obstacles that can appear smart grid envisions power generation highly distributed and mainly reliable on DERs.

2.2.2 Transmission

Generated power is stepped up to higher voltage and moved to the transmission grid and the transmission substations. Due to higher demands the system must support better performance and energy delivery. The use of communication networks and computational intelligence can improve optimal transmission power quality, capacity and stability of transmission lines.

Substations are now able to remotely control and automate processes which provides great flexibility and a helping hand to transmission system operators. This is achieved through monitoring and measurement equipment such as:

Sensors that are used to detect common errors and failures in the power grid. Sensor networks can provide a real-time full overview of the electrical transmission system, perform state estimation and automate or suggest solutions to operators.

Smart relays, meaning digital microprocessor protective relays that can replace classic relays and store data and provide error correction and remotely control electronic devices like switches.

Widely distributed Phasor Measurement Units (PMUs). PMUs measure and monitor system data in real time using GPS synchronization and provide an overview of the

power and network system. PMUs send collected data to the Phasor Data Concentrator (PDC) which displays it to operators. Along with the rest power grid protection and utility equipment are one of the most important tools for the protection of the grid infrastructure.

Smart control centers collect data from sensors and PMUs and use different data analysis techniques in order to make them meaningful, visualize them and use them accordingly. Different management systems are used depending on whether the data are related to energy transmission flow, utility markets or renewable energy resources.

2.2.3 Distribution

This is the step where transmitted power is stepped down to medium voltages before finally reaching end users. Smart relays are also used here for measurement, recording and control of distribution substations. Distribution control rooms can now provide integrated management and visualization of power grid devices, outage management systems and smart meters.

Smart meters are devices that are used to gather consumption data from end user devices. They register data such as power consumption and are able to connect and disconnect customers from the grid or sound an alarm in case of emergency. The smart metering concept (AMI) differs from traditional automatic meter reading (AMR) in that the former allows two-way information flow with the meter through communications networks. In this way recorded data can be sent back to utility centers for real time monitoring and control of user devices.

Consumers can manage their energy consumption proactively with the use of smart meters. They can generate part of their own energy using home solar panels, view billing prices in real time and program their appliances to run when the price of electricity is relatively low.

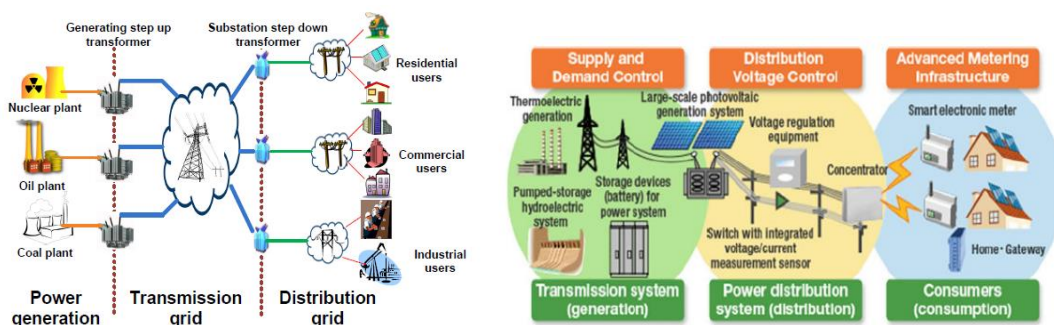


Figure 2 Traditional vs smart grid, extracted from [5], [7]

2.3 DCS and SCADA system overview

Supervisory control and data acquisition [10], [11] is the evolution of the modern ICS control systems. The interconnection of ICT and ICS components brought a transition to

a computer based, remotely accessible through CLI, GUI optimizing control and automation.

General structure: Field devices distributed throughout the plant areas. These devices include: Programmable Logic Controllers (PLCs), that automate basic electrical hardware processes such as relays and switches, based on the sensors feedbacks. Remote Terminal Units (RTUs) which are microcontrollers placed in remote locations that collect analog and digital data and perform simple tasks, sending them back to control centers for storage and visualization. PLCs and RTUs can communicate with multiple intelligent electronic devices. IEDs refer microprocessor-based controllers with sensors and actuators such as circuit breakers and protective relays that are used for smart metering and monitoring. These devices collect and send data to the control center through the communication links. The distributed nature throughout the system in levels of control is the main feature of DCSs. Field sites are usually connected with operation sites over wide area networks.

The main part of SCADA systems is the control center where the collected data from the field devices are gathered, analyzed, generating reports and alarms if necessary. It consists of workstations and servers and Master Terminal Units (MTUs) that communicate, and control distributed remote processes of the field devices, the databases that store the gathered information and the human-machine interface (HMI) for a graphical display of alarms and reports, that helps the operators to monitor and control automated processes manually in case of emergency. HMI could be on workstations, laptops over WLAN or browser based.

ICT equipment is used to connect control centers with field devices and other networks through wired or wireless means.

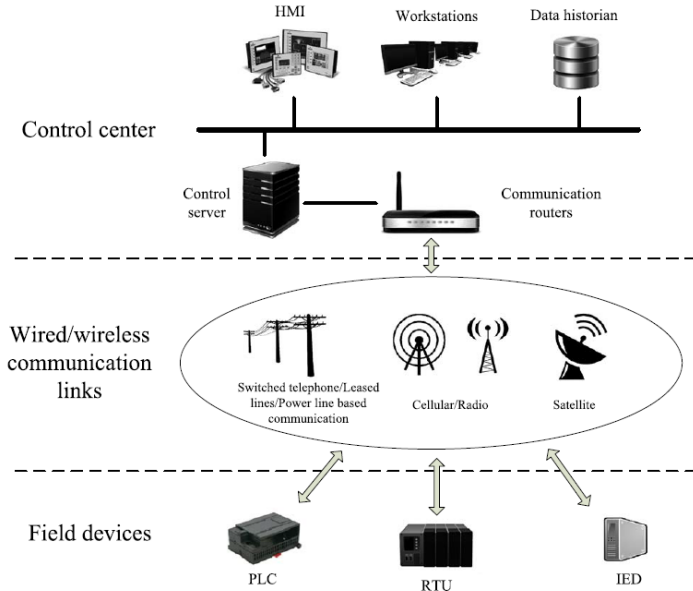


Figure 3 SCADA system overview, extracted from [10]

2.4 Communication networks in smart grid

The main feature of the new generation smart grid infrastructure is the bidirectional communication links between the different parts of the grid [12], [13]. This is achieved with different communication technologies, wired or wireless, that generate and transmit data related to consumption, power flow and devices status, that are useful for analysis, control and monitor of the grid. The capability to access every element of the grid and remotely automate processes renders these networks an essential part of the smart grid infrastructure that however introduces the already know ICT vulnerabilities to the system such as safe data transfer and eavesdropping.

Information that is exchanged in smart grids can be either from sensors and electrical devices to smart meters or from smart meters to utility services and vice versa. The former is suitable for the use of low power wireless communication technologies such as ZigBee and LoWPAN while in the latter cellular or Internet services can be used.

2.4.1 End user/Customer

These networks enable the communication of smart electric appliances with data centers and allow for a real-time consumption display as well as efficient energy management and automation. Depending on the size of the network, it can be divided to Home Area Network (HAN), that connects smart devices and appliances with the smart meter, Business Area Network (BAN) and Industrial Area Network (IAN) where the communication between hardware, software, servers and SCADA/DCS systems is supported. Neighborhood Area Networks (NAN) connect the customer networks with the utilities and the distribution domain.

Technologies used: ZigBee, a wireless communication technology that is low power, low cost, has low data rate and complexity and is suitable for home network metering and energy management demand response. Smart appliances, smart thermostats and photovoltaic panels are example of the communicating devices, interconnected with a smart meter and collector nodes for gathering data. ZigBee low power consumption allows devices batteries to last for long periods of time. Bluetooth can provide an alternative solution of HAN devices connection.

Wi-Fi seems a good choice for the connection of a group of HANS or Business networks where a higher bandwidth with relatively low-cost is required, adding however security issues especially in wireless deployments.

Mesh network topology that uses all nodes as relays that transmit data, a dynamic and self-healing type of network that can adapt easily to changes and failures will probably be highly adopted in these types of networks. Collector nodes communicate with utilities and utility premises through common communication technology mechanisms, including Demilitarized Zones (DMZ) in order to protect sensitive parts of the network.

For Industrial Area Networks an alternative could be the Z-wave network technology, a wireless mesh network that uses lower radio frequencies. End User networks could also be connected to smart meters through Power Line Communications (PLC). There is direct communication with the smart meters therefore no further operational expenses. In a typical PLC scenario, data are transferred to data concentrators through powerlines

and subsequently forwarded to data centers through cellular networks. PLC networks are mostly suitable for densely populated urban areas or geographically dispersed wide area networks.

Home or residential gateways interconnect electronic devices or act individually depending on their type. They are responsible for the security and the translation between private HANs and external IP based networks.

2.4.2 Distribution grid

Due to the change of the conventional power grids to modern ones and the increasing use of distributed energy resources (DER) distribution systems and their operators face significant changes. New requirements demand more human less, adapting of the distribution substations as well as automated monitor and control of loads. Distribution and transmission centers have legacy paths as well as new smart grid communication paths.

Depending on the type of the interconnected devices and the geographical coverage distribution substation and control networks are divided to neighborhood, field area networks (NAN, FAN) and Wide Area Networks (WAN) that are further interconnected to a backhaul network that acts as an intermediate between networks and is responsible for the data collection, automation and control of the distribution process through wired or wireless technologies. Protocols that are or could be used for these types of networks are Wi-Fi ZigBee or WiMax.

Another type is the distribution substation network, a local area network that connects the field devices with a distribution substation and consequently to the backhaul network. Information between field devices is exchanged through wired or wireless communication technologies called feeder networks.

2.4.3 Cellular networks

Another option for smart metering and monitoring is through already existing cellular networks (2G, 3G, LTE, WiMAX). They do not require additional operational costs and deployments and their range could be a good solution for wide area networks and the communication between smart meters and the utilities. Operational and maintenance costs, coverage and license spectrum are managed by the ISPs and different data rates can be chosen depending on the network demands. Their main problems are the congestion due to the large shared use of their services and their lack of availability under certain circumstances.

WiMAX (World Interoperability for Microwave access) can be used as backhaul to increase the capacity of other wireless networks and overcome some of their limitations. Cellular networks are primarily used as backhaul networks between FANs and NANs.

2.4.4 Transmission grid

The implementation of intelligent electronic devices in the generation and transmission subsystems, demand a communication between the substations and the central control systems. This allows for remote control and response to potential system problems during the transfer of electricity from generation to distribution substations (blackouts), that can be manually or automatically controlled by transmission system operators.

The main technologies used are phasor networks which use Phasor Measurement Units (PMUs) for high accuracy metering and synchronization with GPS. These devices can be either independent or integrated into other devices or protective relays. They are the core of the Wide Area Management Systems (WAMS) networks and along with SCADA/DCS control and monitor the transmission power grid subsystem. Ethernet over synchronous optical networking is the main way of establishing a communication link between the transmission substations over the fiber-optic cables of the transmission lines. Where fiber-optic connections do not exist Power Line Carrier for high voltages provides an alternative solution.

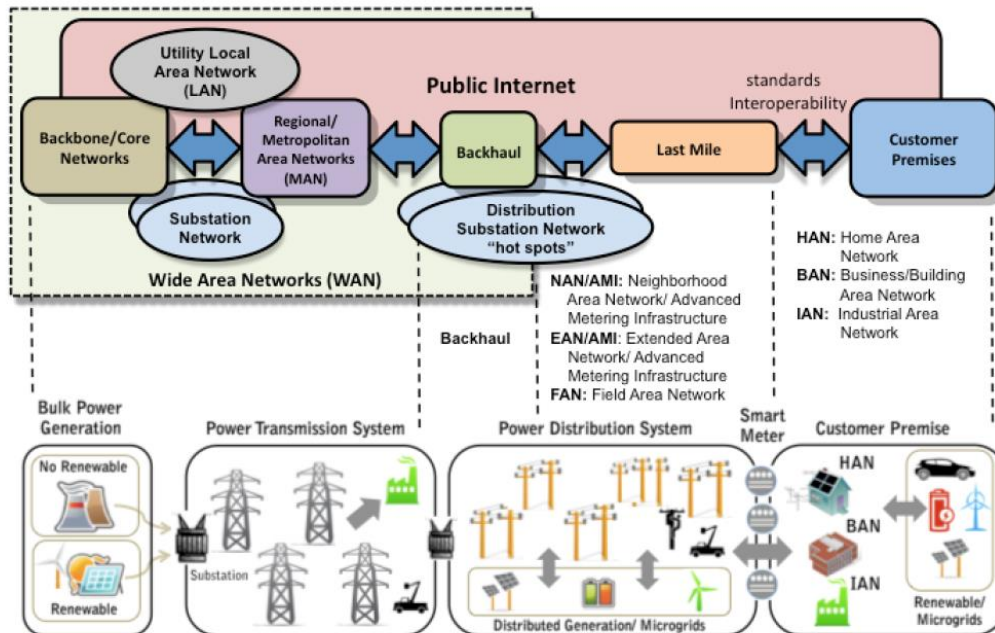


Figure 4 Communication networks in smart grid, extracted from [14]

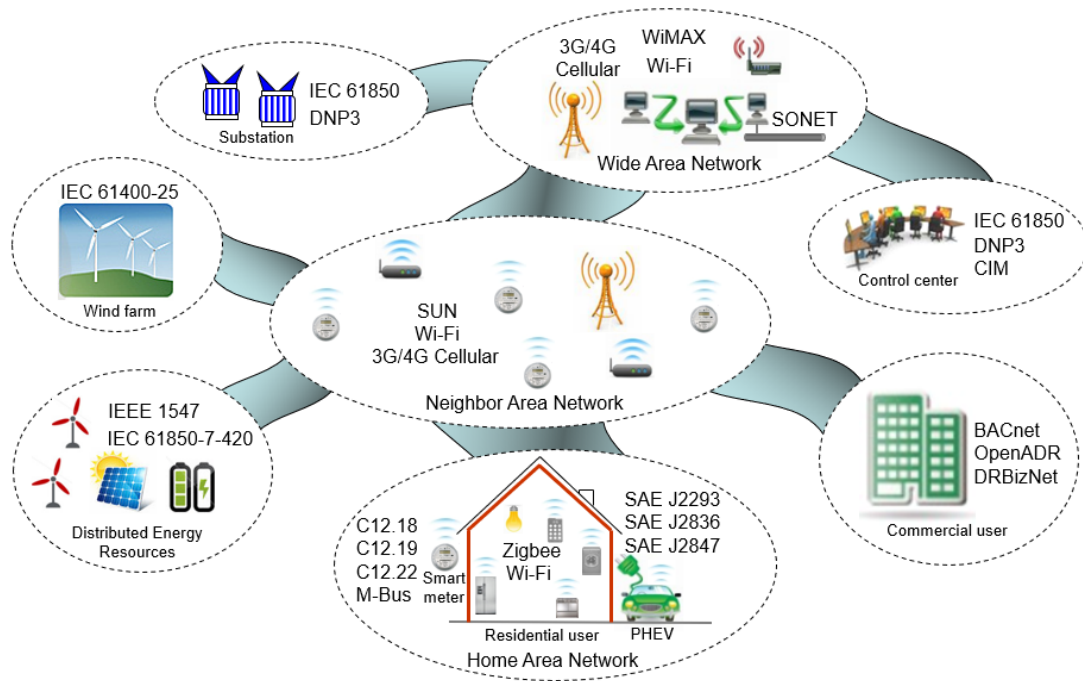


Figure 5 Main communication standards, extracted from [15]

3 Security issues in smart grid

The complexity of the smart grid, which depends on continuously increasing number of interconnected electronic devices, computers, software and communication technologies, form an attractive target that if compromised by an adversary could have devastating effects. Smart grid is recognized as one of the 18 critical infrastructure systems because it is used to control electricity that is required almost everywhere. Therefore, the main security goals are firstly human safety and secondly to maintain integrity and normal process operation. The critical nature of the grid applications adds constraints in terms of delay and data sensitivity. Disruptions could also cause failures or outages and simple solutions such as rebooting an application that work in IT environments aren't acceptable. Security assessments on SCADA systems have identified many cyber security holes, some of them being exposed to the public. The level of automation in substations is constantly increasing, which can lead to more security issues. The deployment of communication technologies and components will also introduce significant risks due to the fact that power grid applications were not designed for an IT based environment. Some of the technologies used are already known while others are under development and security issues and vulnerabilities are currently being discovered. Smart grids' dynamic architecture and size does not foster accurate security assessments, allowing potential attackers to act unpredictably.

3.1 Smart grid vulnerabilities/attacks

3.1.1 SCADA/EMS

Many security assessments have been performed to monitor and control systems of the power grid by Test Bed programs the last decades, exposing a variety of vulnerabilities during the tests [11], [16], [19].

Vulnerabilities that could concern PLCs and RTUs and other ICS equipment are hardware trojans when supply vendors are not reliable. Eavesdropping, tamper and theft of data, modification of firmware and even physical destruction are some of their main functionalities. Furthermore, the replacement of these devices is from 15 to 20 years and leaves a technology implementation gap with the new ones.

HMI vulnerabilities could cause complete loss of control or even physical destruction of components. These could be weaknesses in authentication mechanisms or even coding errors concerning message input validation that could cause buffer overflows.

The constantly increasing level of automation can lead to security concerns to distribution and transmission substations. Automation is related to interconnected electronic devices, hardware and software and communication devices that allow the remote control of switches, circuit breakers, transformers, capacitors and other important electrical devices. Poor security configurations and lack of immediate

security patches from vendors could lead to unauthorized access. The consequences of vulnerability exploitation in this domain would cause a disruption of the power grid operations. Denial of Service (DoS) attacks can be performed by compromised nodes, flooding control centers or RTUs with messages to waste their resources. False data injection could lead to wrong control decisions based on retransmission or modification of messages. Malware infection, access through poor management of databases could also provide entry points for attackers. A compromised center could be a gateway for attacking other transmission grid components.

3.1.2 AMI components

Attacks on smart meters and other AMI devices concern with integrity and confidentiality and privacy information, as they transfer sensitive consumption related data. The great number of devices and their limited computational resources such as bandwidth and memory increase the number of entry points and the risks of a cyberattack [17], [18], [21], [22], [23].

Smart meters

Jamming attacks, DoS like attacks that can be performed by a malicious node to prevent smart meters from connecting to nearby devices by occupying the communicating channel with noise and signals. Wireless signals are transmitted in open space and are susceptible to man in the middle and eavesdropping attacks, the unauthorized sniffing and real-time interception of customer data related to energy consumption and billing prices. Smart meter encryption mechanisms that could protect against these attacks are often simple pattern related and easily deciphered. False data injection attacks, by introducing arbitrary values in data measurements without being detected by the security mechanisms in order to deceive utility service providers. Remote connect and disconnect is a smart meter feature that can malicious users shut down customers meters, steal energy and even cause meter failures that result in fires. Smart meters are also vulnerable to replay attacks due to synchronization issues and delays between transmissions. After gaining access to a vulnerable network, capturing and analyzing transmitted data this attack can be used to redirect energy or even cause physical damage. Finally, the physical accessibility of AMI components renders them susceptible to side-channel attacks, cryptographic attacks which allow information to be extracted from devices' power consumption and processing time.

Home Gateways

Home gateways, an interface that forwards consumption from smart meters and display them in householder devices, are also susceptible to eavesdropping and modification attacks.

PMUs

PMUs are vulnerable to cyberattacks if a node is compromised by an attacker. The attacker may be able to spoof host addresses or alter critical network messages, which could cause damage to transmission and distribution operations.

Communication networks

Vulnerabilities come from different network sources such as enterprise firewalls that filter incoming network packets and monitor traffic, switches and routers and communication systems that transfer data between networks.

The various network layers and the protocols that are adopted for data exchange could also provide potential entry points for attackers. Standard Internet Protocol Suite like TCP/IP or HTTP and common operating systems are being used between corporate and control domain inheriting their vulnerabilities. The Internet Control Center Protocol (ICCP), ModBus and its more efficient version the Distributed Network Protocol 3.0 (DNP) are used for the communication between control centers (MTU) and field devices. These legacy communication protocols were built without security in mind and often come with poor configuration. Reverse engineering these protocols using their publicly available specifications could lead to possible vulnerability discovery and exploitation.

Black hole and selective forwarding attacks that silently drop incoming or outgoing traffic, have been demonstrated against the routing protocol for Low Power and Lossy Networks that HAN, NAN use. ZigBee and WiMAX networks often have encryption issues and can possibly be vulnerable to DoS and sniffing attacks.

Finally, the human factor should always be considered as a security risk. Phishing E-mails with malicious attachments and USB drives with malicious software have started some of the most sophisticated attacks. Security unaware people are highly related to the most known cybersecurity incidents. The main reasons for these kinds of incidents are insufficient training and poor security policies.

Zone	Component	Attacks
SCADA	<i>RTUs</i>	Hardware trojans, eavesdropping, firmware modification, physical
	PLCs	
	HMI	Authentication weaknesses, buffer overflow
	IED and automation	Denial of Service, Malware infection, poor database management
AMI	Smart meters	Jamming DoS attacks, weak encryption, false data injection, replay attacks, physical access
	<i>Gateways</i>	Eavesdropping, data modification
	PMUs	Spoofing, message modification
Communication networks	ModBus, DNP3	Reverse engineering
	HANs, NANs	Black hole, selective forwarding
	ZigBee, WiMAX	DoS, sniffing, encryption vulnerabilities
-	Human factor	Phishing, lack of security awareness

Smart grid main security issues

3.2 Incidents

An attack known as *Aurora* was demonstrated by the Idaho National Lab. It caused explosion of a power generator after gaining access and infecting it with a virus program that turned switches on and off repeatedly, showing that equipment of the grid could be destroyed remotely by a cyberattack.

A *nuclear power generation plant* was forced to shut down for two days. The reason was a software update that caused the rebooting of the computer monitoring system. The safety system translated wrongly the lack of monitoring information, shutting down the power plant, which cost millions of dollars to the company.

Stuxnet, a worm that infected PLCs in Iran is one of the most known attacks in ICS systems. It got into the system via a USB flash drive using a rootkit to hide its presence. It replicated itself through the network due to unpatched vulnerabilities and instructed the victim computers to connect to a C&C server. The server then modified PLCs instructions that ruined thousands of centrifuges.

Another known incident is the *Ukraine power grid hack*. It started with a spear-phishing campaign that targeted IT staff working in different electricity distribution companies in Ukraine. The mails contained a malicious word document that installed a backdoor to their corporate networks. After some months, they managed to gain access to the main controller and collect credentials that led them into the SCADA networks. After writing malicious firmware for serial-to-ethernet converters and replacing the legitimate one, they disabled the operators from sending remote commands. They launched a TDoS attack that prevented the workers from reporting the outage and they finally launched the attack by opening the breakers bringing several substations offline.

A new campaign of attacks called *Dragonfly 2.0*, that targets energy companies has recently been revealed. Hackers gained access to control center interfaces that are responsible for sending commands to critical electronic devices. The intrusion was traced back to 2015 but escalated in mid-2017. Spear-phishing e-mails with malicious attachments targeting unsuspecting victims were the origin of the attacks. These attachments led to compromised websites infected with malware that stole credentials from victims which finally gave them operational access.

4 Simulation

Modern smart grid systems must exchange large data volumes, integrate new features, cope with continuously increasing demands for electricity and be resistant against possible failures and cyberattacks. TCP/IP based communication and technologies will play an important role in smart grids development. This complex and dynamic infrastructure renders simulation frameworks essential for analysis, energy consumption estimation, performance and impact of ICT components on the networks. Simulation enables the evaluation of new architectures that are designed for the smart grid infrastructure before their actual deployment, minimizing risks and costs [24]-[27].

4.1 Categories

Two main categories exist for test platform creation of cyber physical systems such as the smart grid. Real hardware testbeds and software simulation. Real hardware testbeds are further broken down into two categories: Full hardware platforms that offer a complete duplication of ICS hardware. The smart grid testbed in Korea's island Jeju and the renewable energy lab in Greece are some examples. The second category is Hardware-in-the-loop (HIL) simulation that includes both real hardware components and simulated ones are used. At least one real hardware device is needed connected to modeling software to achieve a simulation loop process. This is the best scenario when all layers need to be modeled and full hardware approach is not available.

Modeling smart grid involves different tools that belong to different domains. These tools are often written in different languages, incompatible and in need of a framework that supports diverse simulation technologies. To examine the co-existence of power and communication networks software approach is divided to individual and co-simulation platforms. The former proposes the extension of an individual simulator to enable the simulation of communication networks providing a single integrated simulator, that focus on specific areas of interests, a non-effective solution if scenarios become too complex. The latter proposes the independent use of power and network simulators to form a time synchronized tool that can exchange data between simulators.

The power grid network is modeled by one or more power simulators. The complexity of the network requires simulators that focus on specific domains (transmission, distribution etc.). Wired and wireless communication links, common protocols and control mechanisms are represented by a network simulator. Power grid simulators are usually time-driven, solving a set of equations in discrete small steps while network simulators are event-driven, performing action when something actually happens (packet send or receive, delays etc.) rather than in predetermined intervals. A synchronization mechanism is therefore required for the communication of the co-simulation platform components. This mechanism acts as a middleware that exchanges time and data between the simulators in different ways, depending on the priorities given. Power system output becomes input for network and vice versa, with both

simulators stopping at predefined synchronization points in order to exchange information.

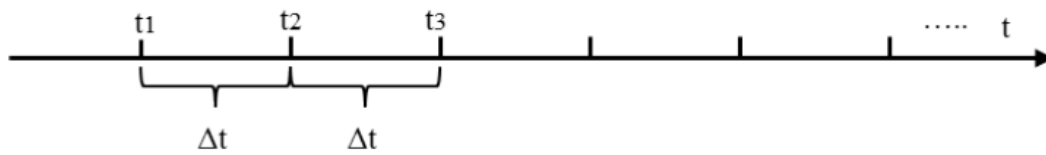


Figure 6 time continuous simulation, extracted from [24]

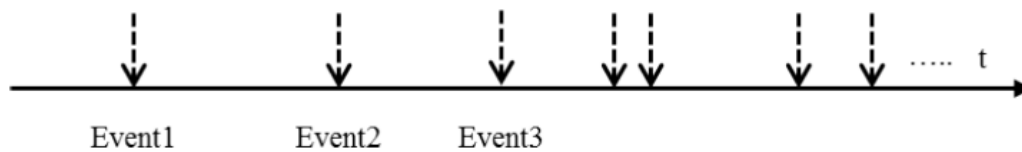


Figure 7 discrete event simulation, extracted from [24]

4.2 Power grid simulators

4.2.1 MATPOWER/PYPOWER

Initially part of the PowerWeb project, it is a package for solving power flow and optimal power flow problems. Power (load) flow analysis is a non-linear solution that estimates real and re-active power based on known voltage and magnitude angle. Optimal power flow optimizes the process using specific constraints adjusting the power system control settings accordingly. Although MATPOWER is open source it comes as a MATLAB package which is a proprietary software. PYPOWER [28] is a translation of MATPOWER using python language and SimPy library.

4.2.2 OpenDSS

An open-source power system simulator by EPRI mainly designed for distribution utilities system simulation [30]. It is a well-suited platform for smart grid scenarios as it is easily extendable and supports analysis related to distributed generation and renewable energy resources. A flexible tool that provides interfaces for users to implement their own models. DSS performs frequency domain circuit analysis such as power flow, harmonics and dynamics. Example of DSS applications are wind generation and farms, PV modules and distribution automation and control. It can be run as a standalone program or as a process from a variety of existing software platforms.

4.2.3 GridLAB-D

A smart grid distribution system simulator developed at U.S. Department of Energy that provides useful information regarding the role of distribution operators, utilities and markets [30]. It is a powerful power system simulator that can also be used to model end-user environments and distribution automation processes. GridLAB-D can simulate millions of interconnected devices efficiently and be easily extended with new modules. Modules are used to define classes that instantiate objects which are being monitored synchronized and updated during simulation process. It provides comprehensive tools for distributed energy resources and data collection for analysis, distribution level power flow and residential models, which makes it most appropriate for modernized grid scenarios.

4.3 Network simulators

4.3.1 OMNeT++

An open source, object oriented discrete event simulator for modeling wired and wireless network environments [32]. It has a basic structure and tools for simulations that make it a proper framework for generic simulation of any network and common communication protocols. Simulation models consist of reusable components called modules that can be connected together to form more complex ones and exchange information. Both Graphical and command-line user interfaces are available for most common operating systems.

OMNeT++ active modules are named simple modules and are written in C++ language and can be group together to form a complex simulation network. Modules and messages are represented by C++ classes in OMNeT++ libraries. The generic model structure is written in OMNeT++'s NED language. Information that is exchanged between modules when simulating a communication network act as frames or packets and travel through simple and complex modules. Connections are used to model the physical links of the network including user or OMNeT++ defined objects related to data rate delays and errors. Parameter values and control flow of the simulation are usually stated in the configuration file. Simulation results can be saved externally or be analyzed through OMNeT++ IDE.

4.3.2 Ns-3

Ns-3 [31] is a discrete event network simulation platform mainly developed for educational purposes and experiments on Internet based environments. Despite its extensive focus on TCP/IP networks it can be also used for more generic modeling purposes. Ns-3 has a modular structure too, with a preference for command line tools. Its' purely written in C++ language with Python bindings and supports simulation

scripts in both languages. It provides more detailed modeling features, allowing the modeling of Wi-Fi, cellular environments and many different routing protocols, it is maintained more actively than its previous version ns-2 and allows for pcap generation that can be used for further analysis.

Ns-3 uses Mercurial as a source code management system and Waf for building the source code libraries. It is mainly designed for Linux like systems giving Windows users a choice between Cygwin or virtual like environments.

4.4 Co-simulation approach

4.4.1 OpenDSS/OMNeT++ co-simulation

SGSim framework

SGSim [33] is an open source smart grid co-simulation framework that uses OpenDSS for power grid simulation and OMNeT++ for communication network simulation. It also provides an interface for the integration of phasor data concentrators allowing the modeling and analysis of lower layer parts.

OpenDSS runs a script that describes the power grid components and their interconnections. The COM interface is used to execute the script while the DSS solver is used to synchronize and connect OpenDSS with OMNeT++ components. The DSS solver is an OMNeT++ component that connects at certain intervals to the DSS process to synchronize the simulators

OMNeT++ is composed of devices like switches and batteries, distributed generation sources, sensors that are used to read data from different components and demand response applications that can be used to measure different data and perform control actions and changes through the COM interface. Simulated packets can also be forwarded to real software PDC components in a real-time simulation scenario.

Because of the difference of the two simulators a DLL library is used to access the elements through the DSS COM interface and return their values to OMNeT++. The library provides a set of functions that set or change values in different elements of the power grid network. OMNeT++'s INET framework, a model library for wired wireless and mobile networks is used for communication networks components providing models for all OSI layers.

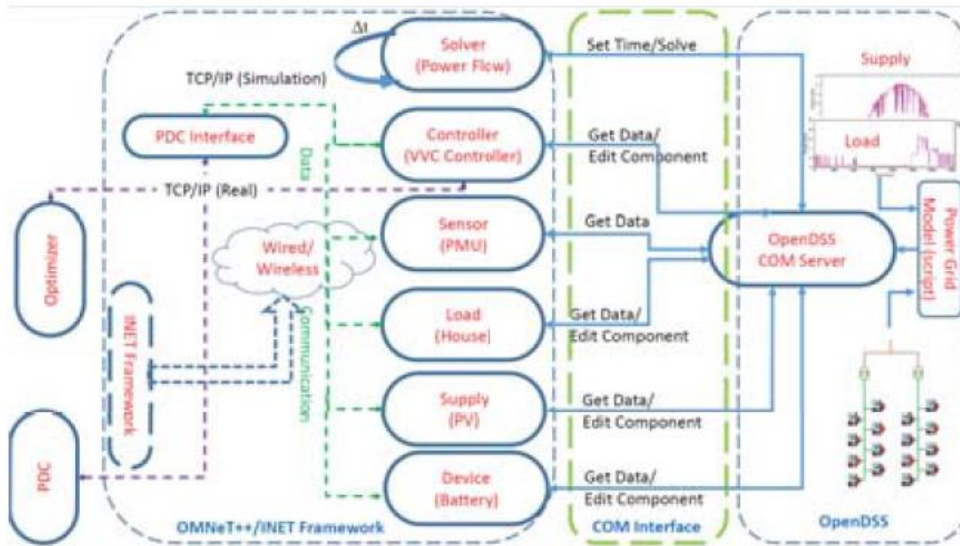


Figure 8 SGSim framework, extracted from [33]

IIT Madras framework

A generic co-simulation framework that also utilizes OpenDSS and OMNeT++ to analyze the performance of PMU based wide area monitoring with the use of smart grid applications [34]. A NASPInet based environment that includes the modeling of PMUs, PDCs, Phasor Gateways and their interconnection is being evaluated. The synchronization is based on the electric power and communication synchronizing simulator (EPOCHS) approach.

OpenDSS scripts are used to define and solve circuits, while modifying parameters at run-time and export their results to files represents the actions of PMUs. A Visual C++ library is used to provide the interface between the two simulators. OMNeT++'s network is composed of PMUs that forward data to PDCs, PGs, control centers and then through the DSS solver to OpenDSS.

EPOCHS uses a Real-time-infrastructure (RTI) package that takes control of the system at certain intervals to exchange information between the simulators. This approach has to deal with two main problems, fault detection and resolution that are being solved by the controller module of OMNeT++.

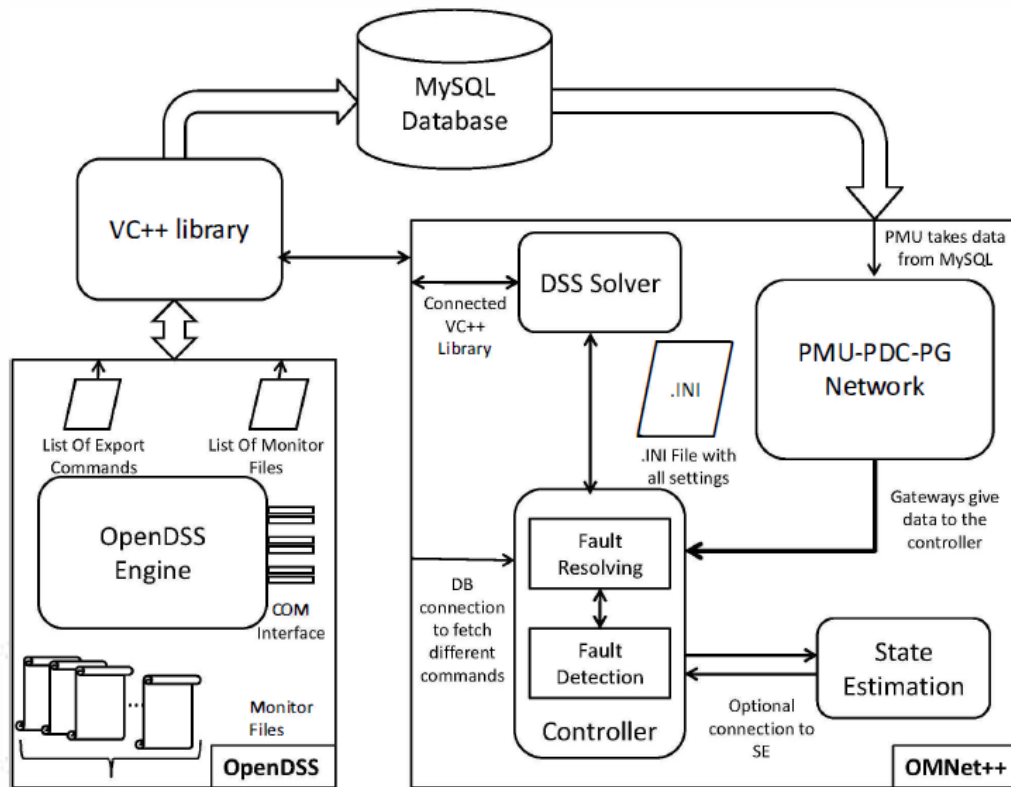


Figure 9 IIT Madras framework [34]

PEVs reactive control framework

An OpenDSS and OMNeT++ co-simulator used to evaluate a reactive control algorithm with plug-in electric vehicles (PEVs) in a distribution network environment [35]. The power network consists of a substation that steps down the transmission voltage to distribution levels, several residential networks that split and step the voltage further down to consumption levels, connected to nodes which correspond to customer households. The communication network consists of a converged fiber and wireless technology that splits to a wireless NAN and a wired or wireless broadband access network.

The co-simulator includes two main components. The process that is used to connect the power and the communication networks. It runs in OMNeT++ and uses HTTP requests and responses to forward information back and forth to OpenDSS. The packets exchanged are related to PEV applications. HTTP is used because the power simulator and the merging co-simulation process run in different OS environments.

4.4.2 FNCS framework

Framework for network co-simulation (FNCS) [36] is a multi-domain co-simulation framework for the modeling of smart grid. It can be used for many smart grid

simulation scenarios, including transmission, distribution markets and communication network modeling scenarios. Like all co-simulation frameworks it is based on the re-use of already known and validated simulators. These simulators run on their own process and FNCS acts as a synchronization and interconnection middleware between them. It mainly uses GridLAB-D and MATPOWER for the power grid and ns-3 for the communication network simulation, integrating transmission, distribution and communication domains simultaneously.

FNCS is programmed in C++ and provides an API for handling time management and intercommunication between the on-use simulators. FNCS core provides an interface (Integrator) that is responsible for handling the initialization and simulation steps between the different discrete-time and discrete-event simulation environments. It also provides an interface (ObjectCommInterface) for the simulator components that need to send messages and communicate with other simulators.

Before each simulation step FNCS initializes all required properties for simulators and other time related dependencies. Two main methods are used for time management and synchronization, one at the beginning of each step to start the simulation and one at the end to get the time of the next step. Time management is important for on time and consistent exchange of messages without adding delays. FNCS depends only on a networking library called ZeroMQ that deals with the sockets that carry all network related messages. Messages are buffered and exchanged during synchronization process.

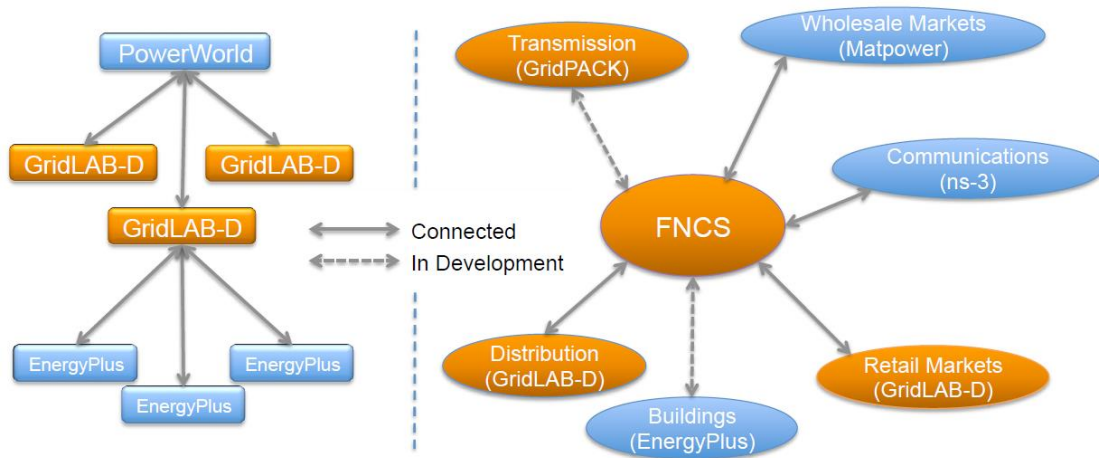


Figure 10 FNCS overview, extracted from [37]

4.4.3 Mosaik

An open source co-simulation framework that is used for the modeling of synchronized smart grid scenarios of already existing simulators and control strategies. Purely written in Python 3 mosaik [38], [39] is flexible in a way that it can be used to model a great number of different scenarios with most known simulators. It offers a language agnostic API (mosaik-core) which allows for these kind of simulation scenarios and handles synchronization and message exchange issues. As a standalone program mosaik is not useful so many free simulators and other helpful tools are provided to for simple simulation modeling.

Mosaik consists of four main components:

The mosaik *Sim API* which handles the communication between mosaik-core and the simulators used. It is divided to low and high-level API. The former uses simple networks sockets to exchange messages between the simulators. The latter is the binding of the low-level API to a specific language. By default, mosaik offers Python and Java APIs. The low-level API allows the users to implement their own work in any other language.

The *scenario API* allows the creation of simulation scenarios in Python. It includes simulators startup, modeling of the entities needed and their interconnection.

The *simulator manager* is responsible for dealing with external simulation processes and connecting with them. This gives mosaik the advantage to integrate simulators, written in any language which will run in its own process.

SimPy library, a discrete-event simulation framework written in Python. It is used for the synchronization and coordination of scenarios with simulators of different step sizes.



Figure 11 mosaik features, extracted from [38]

4.4.4 NeSSi²

NeSSi² [40] provides a security-oriented network simulator environment that can be used for IDS evaluation and efficiency. Its simulation features include common attack scenarios that can be used for research by network and security experts along with a GUI for real-time inspection and configuration.

NeSSi² supports several TCP/IP standard protocols along with an implementation for application (HTTP etc.) network (IPv4, routing etc.) and transport layer with a pcap file format traffic generation. It is built on top of three framework models for simulation environment setup with security related configuration capabilities, simulation execution and evaluation of results.

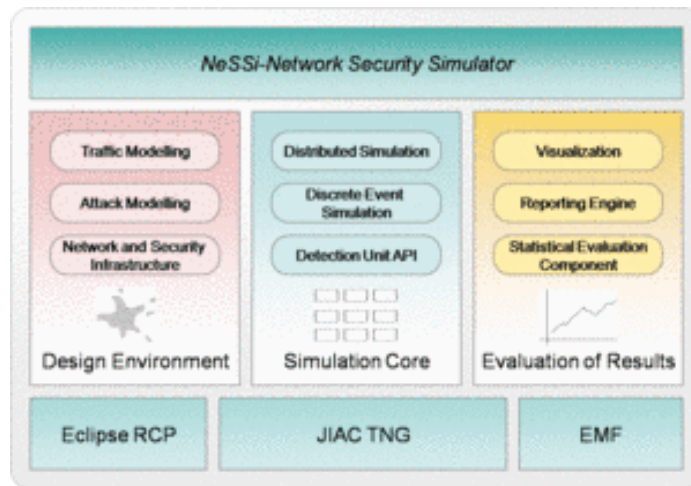


Figure 12 NeSSI2 architecture, extracted from [40]

NeSSI² consists of three main components: the GUI that allows for the creation of drag and drop scenarios and visualization of simulation results, the simulation backend that refers to the machine which creates the network entities and security algorithms, connects with the database and executes the simulation. The database that stores network topology, traffic and detection data generated during the simulation and attack related events for further analysis.

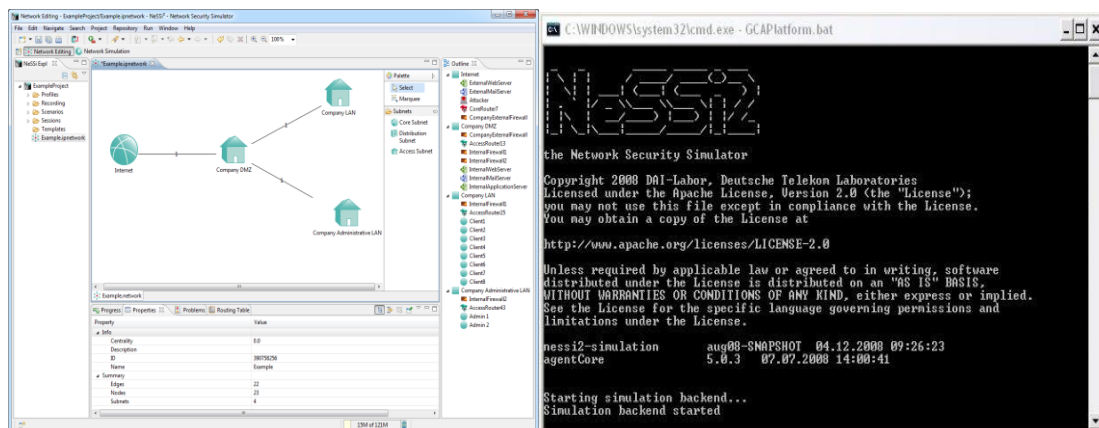


Figure 13 NeSSI2 GUI and backend

NeSSI² are composed of project folders that further include subfolders of networks, applications, profiles, scenarios and simulation parameters. Networks consist of nodes and edges and can be created and manipulated through the palette editor. Applications model behavior of nodes during the simulation and run separately both on backend and user interface in created JAR files. Applications are collected together into profiles to model the entire network behavior. Scenarios are instances of profiles that run on specific nodes in the network. The executable components in the user interface constitute the simulations. Simulation files handle the duration, recording configurations and node mappings between multiple network simulations. The GUI editor allows for visualization of recorded simulation statistics, simulation progress, properties and outline.

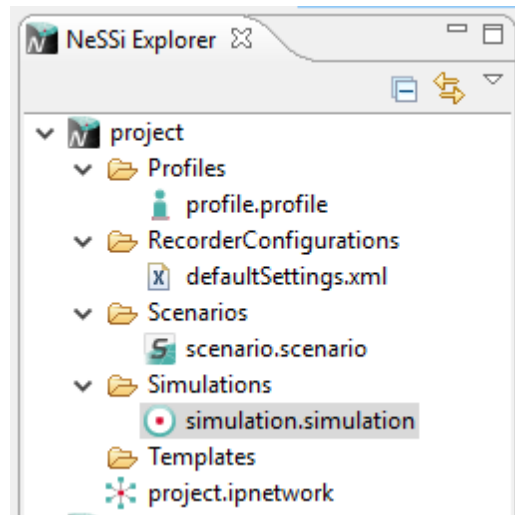


Figure 14 applications -> profiles -> scenarios -> simulations -> NeSSi2 project

4.5 Emulation approach

Besides co-simulation techniques for smart grid modeling, there are a few emulation based attempts to replicate smart grid environment subdomains and communication scenarios. Emulators mimic the original behavior of different components, usually operate at real-time, allow for external hardware and software integration and experiments. Network emulators differ from simulators in the way that they can be attached to hosts and be viewed as a real network would. This gives the potential for real-world scenario network tests using packet generation or other related software and penetration testing platforms. Two attempts to model cyber-attacks against smart grid networks have been made by a variation of the Core and the GNS3 network emulator.

4.5.1 CORE

The Common Open Research Emulator (CORE) [41] aims for the representation of virtualized computer networks that run in real time using standard applications and protocols. Its main goals are to provide an easy-to-use GUI, flexibility and scalability and modification capabilities and is commonly used for networks and security related scenario evaluation.

Core runs on Linux based systems and is based on virtual node networks which are lightweight virtual machines that are built by daemons used to manage sessions between them. Core GUI uses Tcl programming on top of the standard Tk graphical user interface and communicates with the daemons through the API. The user may directly interact with scripts, CLI tools and the GUI.

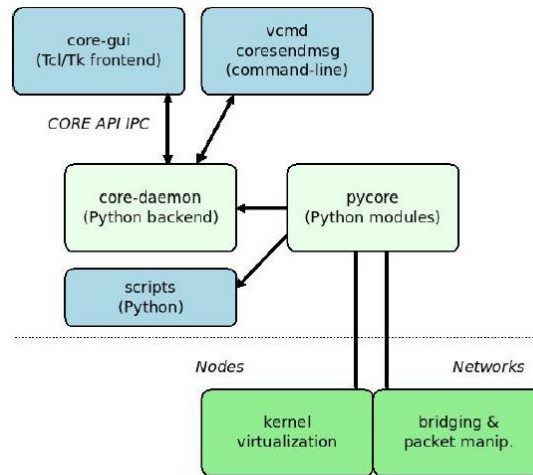


Figure 15 CORE architecture, extracted from [41]

The CORE GUI operates in two modes: Edit mode which provides drag and drop tools for network modeling and emulation instantiation and Execute mode where the tools aim mainly at interacting between the virtual nodes and in visualizing network traffic. Network configuration and traffic can be manual or automatically arranged. Both wired and wireless scenarios are supported.

Core provides interfaces for connecting with physical devices, tunnel tools to interact with other emulations or virtual machines and may also communicate with the host machine that runs the emulation. Services run on core nodes to describe the processes running.

4.5.2 GNS3

GNS3 [42] is an open source network emulator that is used to configure real networks that range from small local topologies to multiple server hosted ones. GNS3 allows for real hardware device virtualization, initially started with cisco devices, now supports many devices from different network vendors as well as Linux, Windows and many other appliances that can be easily added, configured and integrated into its projects.

GNS3 is composed of a client and server part that are made of two software components, the GNS3 all-in-one GUI, and the GNS3 VM. The GUI is the client part of GNS3 that can be used to graphically create and run network topologies. The server part of GNS3 is responsible for hosting and running the network devices and appliances created by the GUI client and can be configured to run in a few different ways. It can be run locally on the host that runs the all-in-one software or on a local/remote GNS3 VM through VMware software. For basic GNS3 topologies a local all-in-one installation that uses the Dynamips older technology is sufficient whereas VM is recommended for a more robust, advanced scenario that makes use of cisco VIRT images.

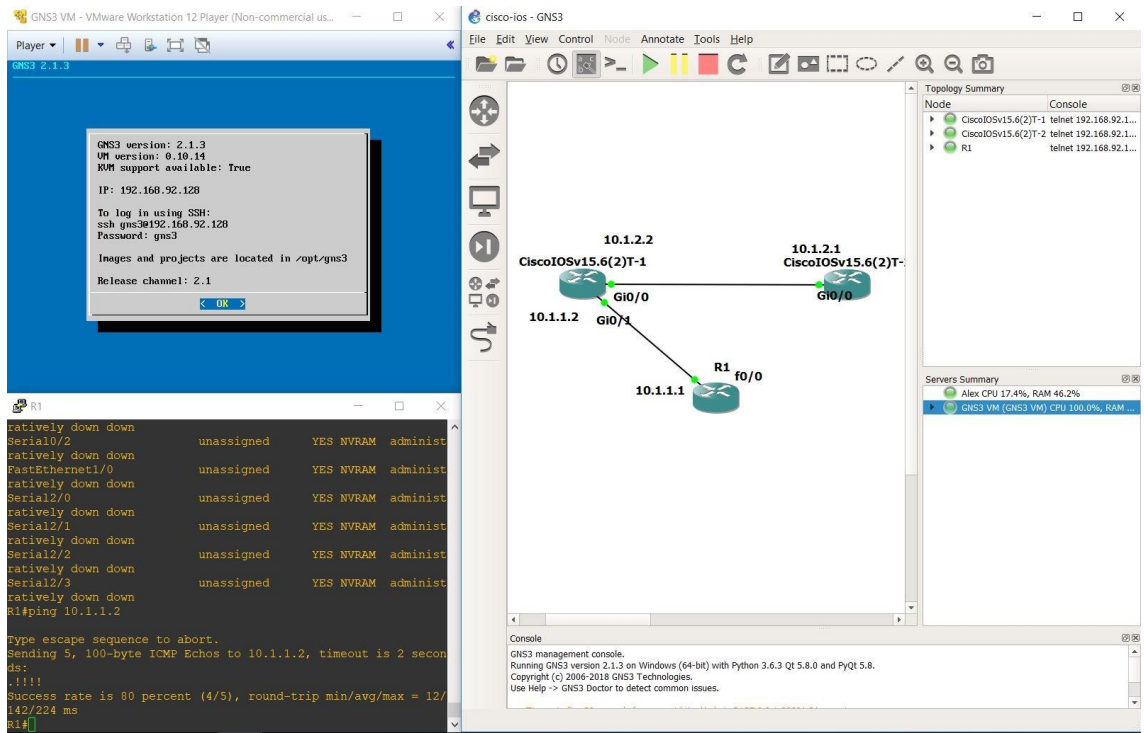


Figure 16 GNS3 GUI, VM and router console

5 Cyber-attack modeling techniques

Cyber-attacks are a delicate issue in the world of cyber security that forces governments and organizations to spend a great amount of effort to detect and handle them, with the use of different tools and techniques keeping operations and services running at normal levels. New types of malware that are constantly emerging are of significant importance to understand them in order to provide better system and network security.

Attack modeling techniques [43]-[45] have been developed and used to find most common vulnerabilities and mitigate ongoing incidents that require instant response. Many researchers have been working on cyber-attack modeling and analysis providing their point of view on how to defend against potential threats. These techniques help in a better understanding of a victim network and its vulnerabilities as well as the attackers' nature and their motives. Modeling an attack in advance helps in better handling and planning if it actually occurs, saving money and resources from potential victims.

5.1 Attack Graphs/Trees

Attack graphs [46]-[49] are tree-structured (with multilevel children and a single root) conceptual diagrams that examine how a target can be attacked. They are divided into four components: the root, which is the fundamental objective, leaf nodes that represent different paths through the attack model and logical AND and OR-nodes. They model different scenarios executed in systems that can lead to undesired and harmful states that are called failures. Failures are caused by malicious actions of adversaries that attack the system or network in order to gain unauthorized access, privileges or cause a deliberate disruption of services. Recent attack graph models are based on tools that provide a detailed representation of the system and network parameters, automated graph analysis and generation based on the modeled environment inputs, visualization and incident response strategies.

The basic information that is needed to generate an attack model is included in the network model and is divided into host and attack action elements. Hosts are objects that are discovered to have been attacked by a malefactor. Attack actions include reconnaissance, preparatory, privilege escalation and confidentiality, integrity and availability compromise actions. In order to add a potential attack in the attack graph the required conditions are that the system or network has vulnerabilities and the adversary has enough knowledge and resources to perform the attack successfully.

Attack graph methodologies that have been proposed by many researchers vary from Buchi 5-tuple automaton models that use acceptance condition and violations to generate attack graphs, hidden Markov models that explore the probabilistic nature between observation and actual states, two-layer or multi-layer graphs that combine vulnerability information and the topology of the network to model attacks, combining host and attack actions with objects of type route threat and graph to evaluate security

metrics and using logical expressions for automatic statistical analysis and countermeasures of attack paths.

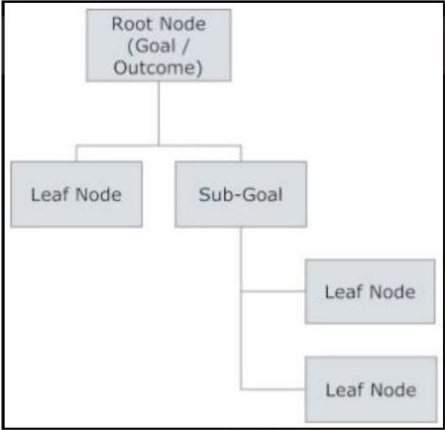


Figure 17 Attack tree conceptual model, extracted from [49]

The main tool approaches for generating attack graphs are TVA (Topological Vulnerability Analysis), NETSPA (Network Security Planning Architecture) and MULVAL (Multihost, multistage Vulnerability Analysis). TVA uses graphs based on system and network condition before and after exploit are used combining their interdependencies to identify all possible attacks paths and remediation actions. TVA models network configuration, including scan reports and firewall rules, vulnerabilities and services, matching them against a modeled database of possible exploits indicating how attackers can penetrate a network. NETSPA aims to generate attack graphs searching through a possible attack space with a use of a simple description language. It also accepts network configuration, software, firewall and IDS information. An efficient tool that can be combined with other security components for network security analysis. MULVAL is a logic programming based network security analyzer. Information is also based on common vulnerability databases and network configurations and graphs are being generated based on the interaction between them.

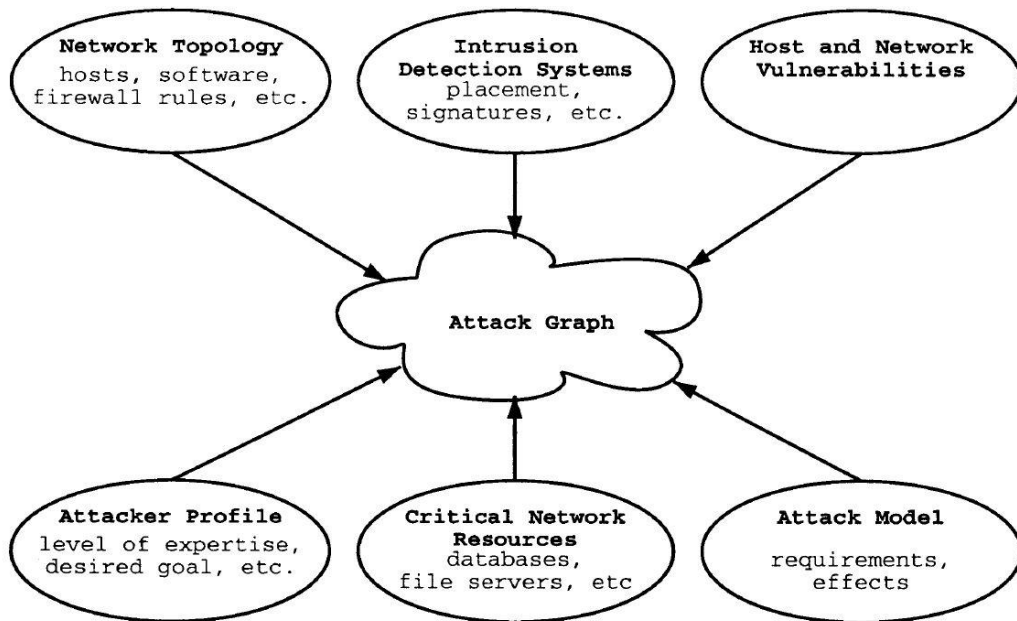


Figure 18 Necessary information to create a graph model, extracted from [48]

5.2 Attack Surface/Vector

An attack vector [50] is a path or a set of tools, commands and events that a user can gain unauthorized access to a computer system or network. Examples include malicious email attachments and pop-up windows, chat rooms, vulnerable web pages or web pages with malicious content, and the human factor. Vectors refer more to the paths that allow attackers to land their attacks rather than the malware itself like viruses or other malware executables.

Attack surface is the sum of the attack vectors that an adversary may use to attack a modeled environment. To visualize an attack surface model the first step is to map the topology of the targeted system or network. This includes servers, endpoints, network and security devices such as firewalls, VPNs and IDSs. The second step is to identify and map all the Indicators of Exposure (IOEs). IOEs can be found in all different entry and exit points such as UIs, APIs, databases, files from outside the network, emails, authentication, authorization, operational and monitoring interfaces. These include software vulnerabilities, and lack of proper input validation poor security mechanisms in control systems and security policy violations. An additional step that can be taken is discovering the Indicators of Compromise (IOCs), forensic artifacts that indicate a computer intrusion, data collected from security logs and scanning tools like malware hashes and blacklisted IP addresses. The last step includes using the topology, the IOCs and the IOEs together to attain valuable security related information.

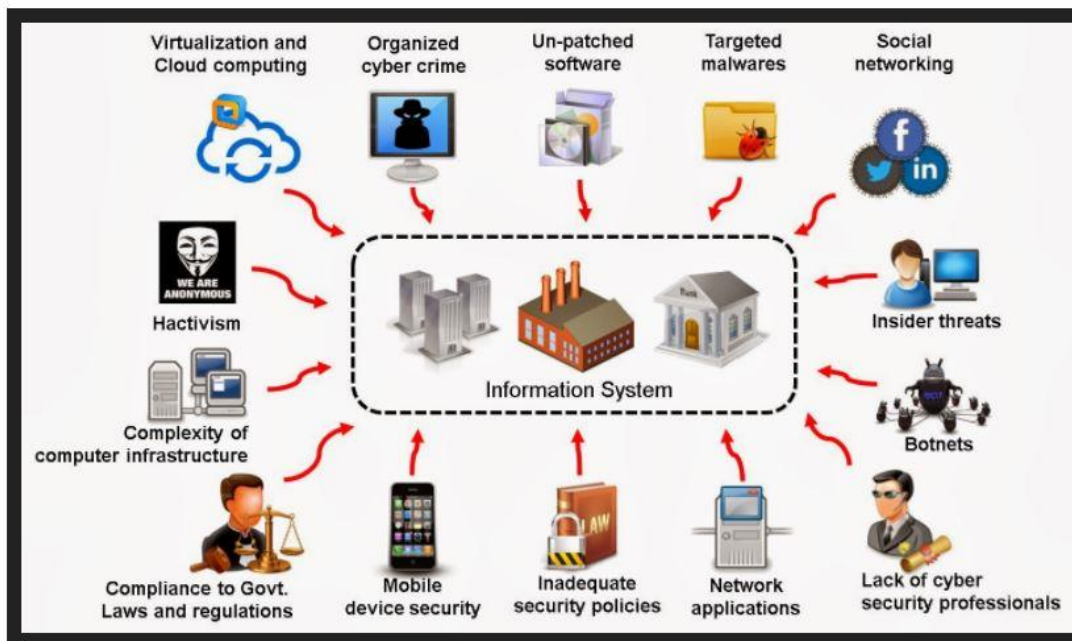


Figure 19 A sum of information security attack vectors, extracted from [51]

5.3 OWASP and Threat Modeling Techniques

OWASP an open community that focuses on application security, defines threat modeling as a procedure for optimizing security by identifying vulnerabilities and threats and providing the corresponding countermeasures and mitigations. It is a process that starts during the planning phase of an application development and runs throughout its lifecycle as new features are added. It is based on a generic process that, rather than searching for every possible threat and vulnerability, focuses on factors that have greater probability or impact if exploited.

OWASP divides threat modeling [53] to a multi-step process that starts with the assessment scope which aims at identifying all tangible and intangible assets. The process goes on with the modeling of the system and with identifying all possible threat agents that can harm the system based on the data given and all the possible ways they can achieve it. The next few steps include a study of the vulnerabilities that can indeed be exploited and the already existing countermeasures. The process ends with an evaluated prioritization to reduce the risks to satisfactory levels.

There are different organized techniques for threat analysis such as Threat Modeling Attack Paths (T-MAP), STRIDE and Petri-nets. Most of these require a thorough study of assets, entry points and data flow diagrams (DFDs) for a better understanding of the system. A DFD is a graphical representation of data in the system. Its main elements are the external entity, data flow, data store and processes. Flow diagrams are created in as many steps required. An alternative to DFDs are activity diagrams which focus on the workflow of the system. STRIDE is a reminder of Spoofing, Tampering, Repudiation, Denial of Service, Elevation of privilege and is used to identify these types of threats.

STRIDE assumes that at least a DFD is present and is often used together with other threat modeling techniques. T-MAP is a quantitative threat modeling technique used for calculating the weights of possible attack paths. Attack paths include firewalls, commercial off the shelf systems and IT infrastructure. By calculating all weights, the overall threat is known. Petri-nets also known as p/t nets or places and transition nets propose an alternative technique. As the term implies, they consist of places or states, transitions and arcs that suggest changes in state of the modeled system.

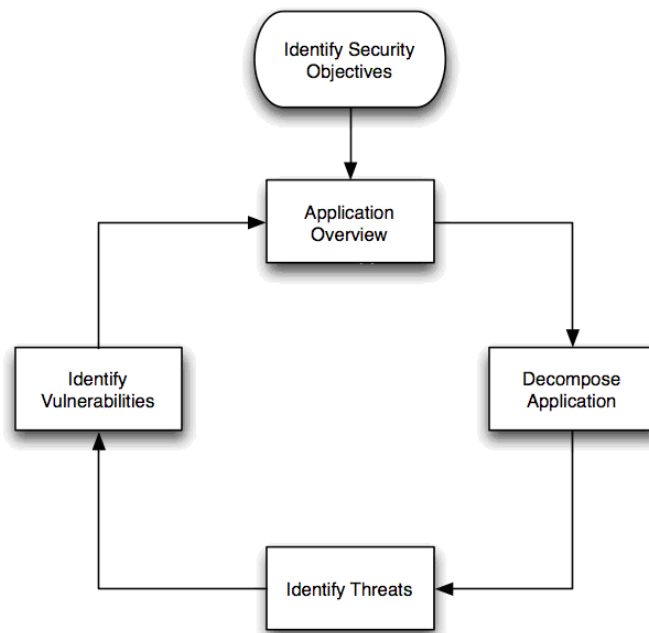


Figure 20 OWASP threat modeling, extracted from [53]

OWASP provides both web and cloud-based threat modeling projects that aim at finding possible attack scenarios on systems and what can be done in order to detect them or mitigate their impact. Threat Dragon is an online web application that uses data flow diagrams to model system behavior and a rule engine to autogenerate threats and mitigations following the STRIDE approach. OWASP cloud security project focuses on cloud-based products and services security with the use of human-friendly threat model templates in simple readme files and control actions in the form of Behavioral Driven Development stories.

5.4 Diamond model

The Diamond model [53] is one of the main models for cyber intrusion analysis that focuses on the intrusion event element rather than using a stepped based approach. It consists of four basic elements: adversary, victim, capability and infrastructure which are represented in a diamond shape that implies their underlying dependencies. There are also some meta-features included that are used to link events together into activity

groups forming an extensive model of suspicious activity. It is a flexible model that captures adversary actions, able to be extended with new ideas.

Core definitions: An event is a discrete time activity requiring external resources, that an adversary executes against a victim using a capability over some infrastructure. Each event has core and meta features as described earlier and a confidence value, a function which may vary in different model scenarios. An adversary is the one responsible for using a capability against a victim. Adversaries include insiders, outsiders and organizations that launch attacks for personal gain. Capability describes the tools of the adversary that are used during an event. Capacity is the set of vulnerabilities that can be exploited by an adversary capability. The sum of the tools used by an attacker, from the simplest to the most sophisticated one constitute the adversary’s arsenal. Infrastructure refers to the physical and logical communication between the victim and the adversary during the capability usage. Infrastructure is divided to type 1 that is fully controlled by the attacker, type 2 that is controlled by an intermediary and can be useful for obfuscation purposes and service providers for the communication of type 1 and 2 infrastructures. Victim is the target of the attacker and can be described either as a person, organization, industry or as an asset like a network system and an IP address. Vulnerabilities that concern a specific target form the susceptibilities of the victim and are expressed in plain text or in a formal common vulnerabilities and exposures (CVE) form.

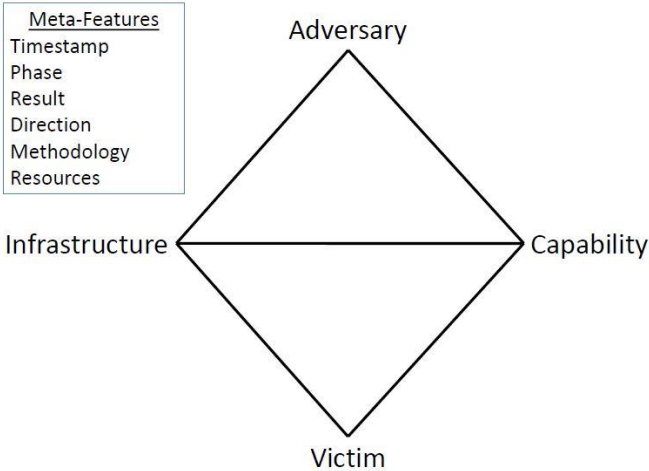


Figure 21 Diamond model overview, extracted from [53]

Meta-features: These are non-critical but important events that can help in a better description of the model. They can be dynamically defined or extended by the user needs. Typical examples are: Timestamps, which refer to the time or date of the incident and can be moments or durations of malicious activity. Phases or chain of events that a malicious activity can be broken down into. Phases can be actions like host scanning discovery, reconnaissance and exploitation. Results report the condition after the attack and are useful for determining its success rate. There are several ways that they can be defined such as the success, failure and unknown triplet or confidentiality, integrity and availability compromise. Direction is a feature that can be primarily used for the flow of

network based events and is essential for the decision of countermeasures and detection actions. Methodology is used to describe the kind of malicious activity like a port scan or a spear phishing e-mail. There are many related formal studies like the Snort class types. Resources are all the supporting elements that core and meta-features really onto. Examples include software, hardware, knowledge and information. Meta-features are not limited to the aforementioned and can be adapted to each scenario for a better grasp of the model.

5.5 Kill Chain model

Kill chain [55] is a phase-based systematic process that describes the stages of an attack. As a military term it was defined as a multi-step process of find, fix, track, target, engage and assess (F2T2EA). The main concept behind the model term is that the closer to the beginning of the chain an attack can be stopped, the better in terms of cost and time it will be. It is also applied in the cyber security sector accordingly also providing a countermeasure framework. The cyber intrusion kill chain is a seven-step process which will be described below:

Reconnaissance: this is the information gathering process that an attacker executes before an attack. Common sources include web crawlers, mailing lists, social media. Many passive reconnaissance tools exist on the internet that hold information useful for attackers such as search engines for connected devices on the internet, IP addresses, domains and website data. Active tools can be used to extract more information on identified targets.

Weaponization: the selection of appropriate tools and the creation of the malicious payload that will be sent to the victim. Data found from reconnaissance step such OS flavor, server types and services are essential for the malware type selection. It could be a trojan behind an executable file that gives remote access to the attacker or a spear-phishing attack to gain access to restricted information.

Delivery: The act of transmission of the weapons to the target environment through some means of communication. The most popular ways to deliver payloads are email attachments, websites and USB removable.

Exploitation: This phase starts when the victim has downloaded the payload in his computer. The payload targets a service or application that has a certain vulnerability and executes code. This is a crucial phase because the chain can be killed if the payload is not downloaded.

Installation: The execution of the payload either manually by the victim or automatically. This is also an important step for the adversary to maintain persistence in the target environment. Manual execution means that the chain can be broken if the payload is not executed.

Command and control: Compromised hosts connect to a controller server and wait for instructions. In this phase the attacker has successfully gained access in the target environment.

Actions on objectives: The final step, when the adversary has already access on the targeted system, is to take actions to achieve the objective of the attack. This includes collecting, encrypting or extracting information from the victim. Other objectives include attacks against integrity and availability. The target could also be used as an intermediary point to compromise additional systems.

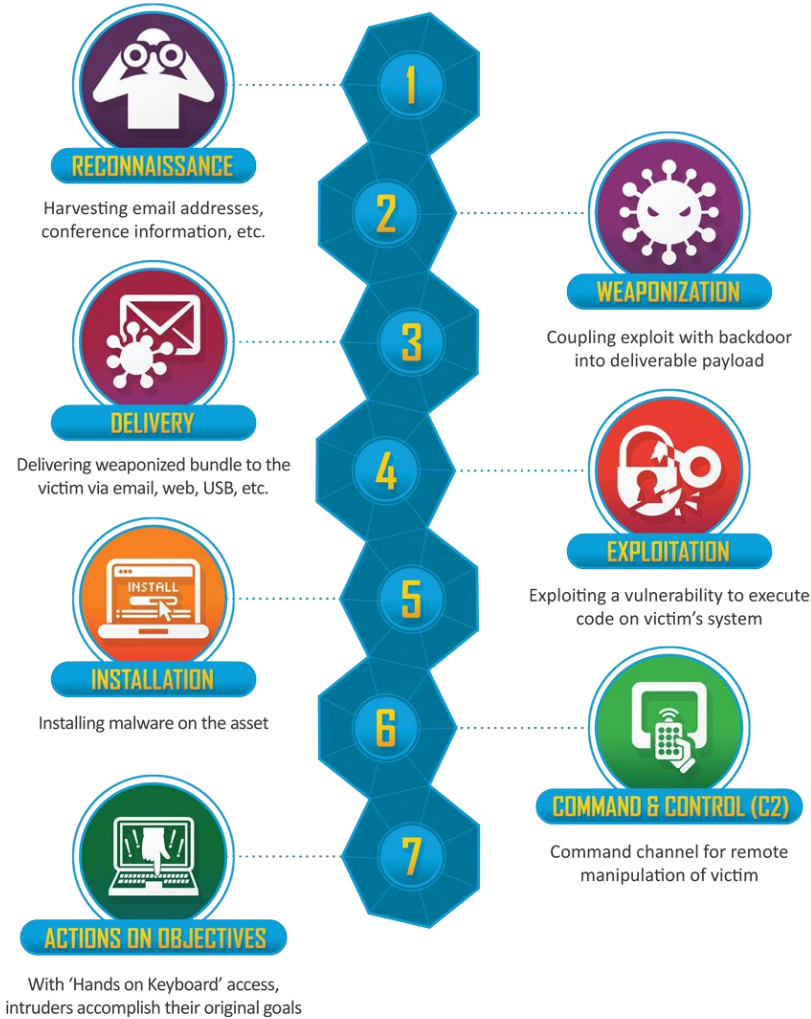


Figure 22 the 7-step process of killchain model, extracted from [54]

Based on the previous, kill chain is divided into two main phases that correspond to the left and right part of the exploit. The left part includes the reconnaissance, weaponization and delivery phases and understanding its' patterns is of great importance for cyber defense. The adversary then waits for the victims' response, whose decision will be crucial for the right part phases and the delivery of the exploit. Some attacks follow their own rules by skipping or adding extra steps. Zero-day

exploits and newly discovered vulnerabilities often mean that we move to the right part of the chain which requires instant incident response and patch deployment.

5.6 Combining models

Modeling methods can be combined to give a better view both from the adversary's and the target system's perspective. ARENA [56] discrete-event simulation model, creates a virtual network that generates representative cyber-attack and intrusion alert data, creating scenarios based on a combination modeling approach. The first concept is a kill switch variation that models hackers' typical actions into 10 stages, as a sequence of events. The first five stages (Reconnaissance, intrusion user, escalation, intrusion root, DoS,) represent actions on external machines while the five last (enumeration, intrusion user, escalation, intrusion root, goal pilfering) on internal. The second concept includes graphs that show both the progression of cyber-attacks and the representation of their structure. Parameters included are goals, targets and other attack related terms. A logic diagram helps with attack generation by choosing legitimate attackers, victims and the possible ways they can penetrate the network.

5.7 Attack models on CPS

Critical infrastructures such as the power grid have become an attractive target for attackers that can may put integrity of operations and human safety at risk. Therefore, cyber security is now an integral part of CPS and many modeling scenarios have been proposed for a better analysis of their consequences.

[57] Provides a categorization on CPS based attacks to conventional and cross-domain ones. Stuxnet like attacks that crossed the cyber-physical domain boundary played a big part in the proposed model structure which provides a qualitative and quantitative analysis of the attacks. It is based on the existing classifications of cyber-attacks with single or multi-dimension properties giving an emphasis to cross-domain features.

The main terms are: The target group which includes both influenced elements directly affected and the influence that describes what is changed on the target element, the effects group that includes the victim element and the impact on victim (terms equivalent to the target group subparts that may belong to different domains) and the attack group which consists of the means that made it successful and the preconditions required. The subtle distinction between influenced and victim elements allows for a division of the attacks to four categories based on whether they belong to the cyber or physical domain.

The proposed method can be used in several application areas such as documentation and analysis, by providing a formal representation of attack on CPS, vulnerability assessment by utilizing attack means, influenced elements and preconditions for an

attack to be successful and better attack propagation and impact assessment with the help of the cyber and physical categorization scheme.

[58] uses the attack vector model to identify all possible security holes on SCADA, discusses the system protection infrastructure in CPS, examining a SCADA and communication network scenario under a malware injection, a DoS and a MITM attack. Netlogo, a multi-agent modeling environment is used to help with the malware injection occurrence in the corporate network and ns2 to compute the results of the DoS and MITM attack.

5.8 Attack models on Smart Grid

[59] aware of the complicated, large-scale nature of Critical infrastructure systems such as the smart grid, combines system theory and the cyber world to follow a cross-domain like taxonomy of attacks. Attacks and consequences are therefore classified into four categories depending on whether they affect cyber or physical components of the system. Based on the main three security principles, confidentiality of power related data, integrity of software and control commands and attacks against availability of services are the main requirements for a secure smart grid environment. Attack stages include the exploit of one or more entry points (network holes, backdoors, external devices, insiders) and the attackers actions and consequences after the successful compromise of an entry point (false data injection, database access, denial of service). Consequences included belong both to the cyber and physical domains.

		Consequence		
		Cyber	Physical	
Attack	Cyber	Eavesdropping of private information	Stuxnet	
	Physical	Meter bypassing	Instability due to physical destructions	

	Price information	Control command	Meter data	Software
Confidentiality	Leakage of price info.	Exposure of control structure	Unauthorized access to meter data	Theft of proprietary software
Integrity	Incorrect price info.	Changes of control commands	Incorrect meter data	Malicious software
Availability	Unavailability of price info.	Inability to control grid	Unavailability of billing info.	N/A

Figure 23 Attack taxonomy and threat classification, extracted from [59]

[60] comprehends a study about the attack surface of the AMI with an implied aim to provide effective cybersecurity countermeasures. AMI provides sensitive information about power grid parameters, billing prices and can be utilized by consumers to perform power control actions. According to the paper the main attack domains include denial of power service to a consumer, theft of power from utilities and disruption of the grid. The study separates the AMI attack surface into three main categories:

Smart meters are the main component of the AMI which gather power and billing related data and can connect or disconnect an end user to the power grid. Potential attack vectors are wireless network, attacks to the communication radios and attacks on serial links through physical access to the meters. Smart meters are based on low-cost components which can lead to more security concerns.

The ICT network which connects smart meters and data concentrators together, with the HAN and with the utilities WAN is another entry point. Common vulnerabilities in ZigBee or WiFi protocols can give network access to an attacker. HAN side compromise could further allow an attacker to remotely control smart appliances due to the shared interface with the home computers.

Vulnerabilities against commonly used protocols (TCP, UDP, IPv6) and software also provide a sum of possible attack vectors.

[61] provides a framework in order to estimate the level of susceptibility of the smart grid infrastructure to attacks. An integration with NIST risk framework is used to provide more accurate results. The model is based on terms like privileges, which is access to specific parts of the system, information objects which constitute primary target points, attackers and security common mechanisms. A stride-based data flow diagram is used to give a graphical overview of the model.

The exposure diagram is an arc connected relationship between the main terms of the model through the possible attack paths. The graph development begins by suspicious information flow attacker identification. Each attacker requires an amount of effort, represented by a number, to bypass a security mechanism and gain a set of privileges that can finally give him access to targeted information objects.

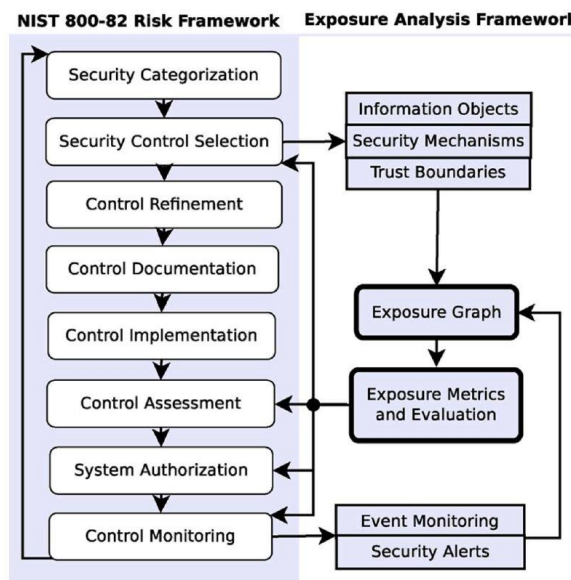


Figure 24 proposed framework, extracted from [61]

6 Analysis of cyber-attack modeling in smart grid related frameworks

This section comprehends the most considerable up to date efforts of cyber-attack modeling in smart grid infrastructure sub-domains. Albeit smart grid modeling has been achieved with many different combinations of the tools mentioned in the previous sections, the attempts for cyber-attack related models are minimal and currently undergoing research. In those efforts however, the proposed frameworks have followed both co-simulation approaches combining both power and communication networks for impact evaluation, and emulation approaches using virtualized software and/or real hardware to interact and experiment with the system in real time.

6.1 ASTORIA

ASTORIA [62] (Attack Simulation TOolset for smart gRid InfrAstructures) is a co-simulation framework that uses mosaik and its integrated simulators to model the power grid elements combined with the ns3 simulator for the ICT equipment. It enables simulation of cyber-attacks by providing a set of built in profiles that correspond to different attack scenarios.

6.1.1 Mosaik overview

Mosaik framework provides a demo which is a good starting point and with simple code changes can be adapted to the user needs. The main script includes the configuration parameters for the simulators used, the scenario and the actual run of the co-simulation. Json or xls files are used to define the grid parameters that will be passed to PyPower and csv files for the production and consumption profiles.

We modify the json script that corresponds to the power grid topology to create a scenario with 3 households, 1 photovoltaic panel, 1 windmill generator, 1 electric vehicle profiles, all connected to a transformer and a reference to the upper power grid. The script includes the required transformer, buses, branches and their interconnection that PyPower will use to solve the flow of the model.

```

{
  "base_mva": 10,
  "bus": [
    ["tr_pri", "REF", 20.0],
    ["tr_sec", "PQ", 0.23],
    ["node_a1", "PQ", 0.23],
    ["node_a2", "PQ", 0.23],
    ["node_b1", "PQ", 0.23],
    ["node_b2", "PQ", 0.23]
  ],
  "trafo": [
    ["transformer", "tr_pri", "tr_sec", 0.25, 4.2, 0.00275, 6.9, 360.8]
  ],
  "branch": [
    ["branch_1", "tr_sec", "node_a1", 0.100, 0.2542, 0.080425, 0.0, 240.0],
    ["branch_2", "node_a1", "node_a2", 0.375, 0.2542, 0.080425, 0.0, 240.0],
    ["branch_3", "tr_sec", "node_b1", 0.021, 0.2542, 0.080425, 0.0, 240.0],
    ["branch_4", "node_b1", "node_b2", 0.021, 0.2542, 0.080425, 0.0, 240.0]
  ]
}

```

Figure 25 Grid Topology

In the main script, after importing mosaik and other required modules we define the simulators we want to include, with the required parameters. They are used by default from mosaik and include PyPower, a HouseholdSim used to model residual load consumption profiles, a CSV simulator that models the profiles of renewable energy resources and the electric vehicle, the database used by mosaik to store simulation data and the mosaik web visualization simulator.

```

sim_config = {
  'CSV': {
    'python': 'mosaik_csv:CSV',
  },
  'DB': {
    'cmd': 'mosaik-hdf5 %(addr)s',
  },
  'HouseholdSim': {
    'python': 'householdsim.mosaik:HouseholdSim',
    # 'cmd': 'mosaik-householdsim %(addr)s',
  },
  'PyPower': {
    'python': 'mosaik_pypower.mosaik:PyPower',
    # 'cmd': 'mosaik-pypower %(addr)s',
  },
  'WebVis': {
    'cmd': 'mosaik-web -s 0.0.0.0:8000 %(addr)s',
  },
}

# Start simulators
pypower = world.start('PyPower', step_size=15*60)
hhsim = world.start('HouseholdSim')
pvsim = world.start('CSV', sim_start=START, datafile=PV_DATA)
weccsim = world.start('CSV', sim_start=START, datafile=WECS_DATA)
evsim = world.start('CSV', sim_start = START, datafile=EV_DATA)

# Instantiate models
grid = pypower.Grid(gridfile=GRID_FILE).children
houses = hhsim.ResidentialLoads(sim_start=START,
                                profile_file=PROFILE_FILE,
                                grid_name=GRID_NAME).children

pvs = pvsim.PV.create(1)
weccs = weccsim.WECS.create(1)
evs = evsim.EV.create(1)

# Connect entities
connect_buildings_to_grid(world, houses, grid)
connect_randomly(world, pvs, [e for e in grid if 'a1' in e.eid], 'P')
connect_randomly(world, weccs, [e for e in grid if 'a2' in e.eid], 'P')
connect_randomly(world, evs, [e for e in grid if 'b1' in e.eid], 'P')

#connect_randomly(world, pvs, [e for e in grid if 'node' in e.eid], 'P')

```

Figure 26 Simulation configuration and scenario definition

Useful simulation parameters and profiles are stored into local variables and will be used to build the scenario. The main function creates a mosaik world using the configuration we defined earlier, creates the scenario which will be displayed in the web browser until we tell it to stop.


```

def main():
    random.seed(23)
    world = mosaik.World(sim_config)
    create_scenario(world)
    webbrowser.open('http://localhost:8000')
    world.run(until=END) # As fast as possible
    #world.run(until=END, rt_factor=1/60) # Real-time 1min -> 1sec

START = '2014-01-01 00:00:00'
#END = 31 * 24 * 3600 # 1 day
END = 3600*24*5 #5days
PV_DATA = 'data/pv_10kw.csv'
WECS_DATA = 'data/wecs_10kw_small.csv'
EV_DATA = 'data/ev_small.csv'
PROFILE_FILE = 'data/profiles.data'
GRID_NAME = 'demo_lv_grid'
GRID_FILE = 'data/%s.json' % GRID_NAME

```

Figure 27 main function and profile parameters

The rest of the code includes the connection between simulation entities and the web visualization parameters. Executing the script starts the simulation and opens up a browser window where simulation entities and their parameters can be explored. We changed a bit the CSS code in the web visualization simulator to separate between the entities.

```

Starting "PyPower" as "PyPower-0" ...
Starting "HouseholdSim" as "HouseholdSim-0" ...
Starting "CSV" as "CSV-0" ...
Starting "CSV" as "CSV-1" ...
Starting "CSV" as "CSV-2" ...
Starting "DB" as "DB-0" ...
INFO:mosaik_api:Starting MosaikHdf5 ...
Starting "WebVis" as "WebVis-0" ...
INFO:mosaik_api:Starting MosaikWeb ...
Starting simulation.
INFO:mosaik_web.mosaik:Creating topology ...
INFO:mosaik_web.mosaik:Topology created
Progress: 48.81%

```

- EV
- House
- PQBus
- PV
- RefBus
- WECS

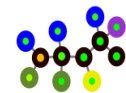


Figure 28 Simulation execution and web browser visualization

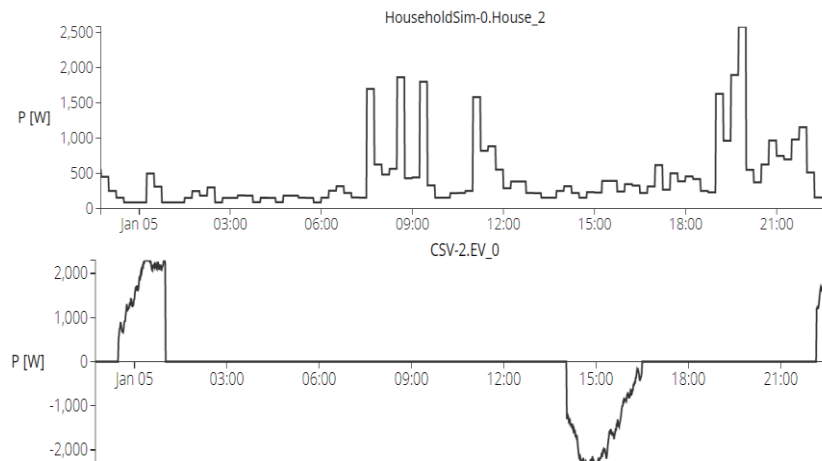


Figure 29 Load profiles of a household and an electric vehicle

Mosaik also provides an easy to use GUI called Maverig where we can easily drag and drop the previous scenario. Many simulators are already predefined. It also supports the integration of new ones and provides built in consumption and production profiles.

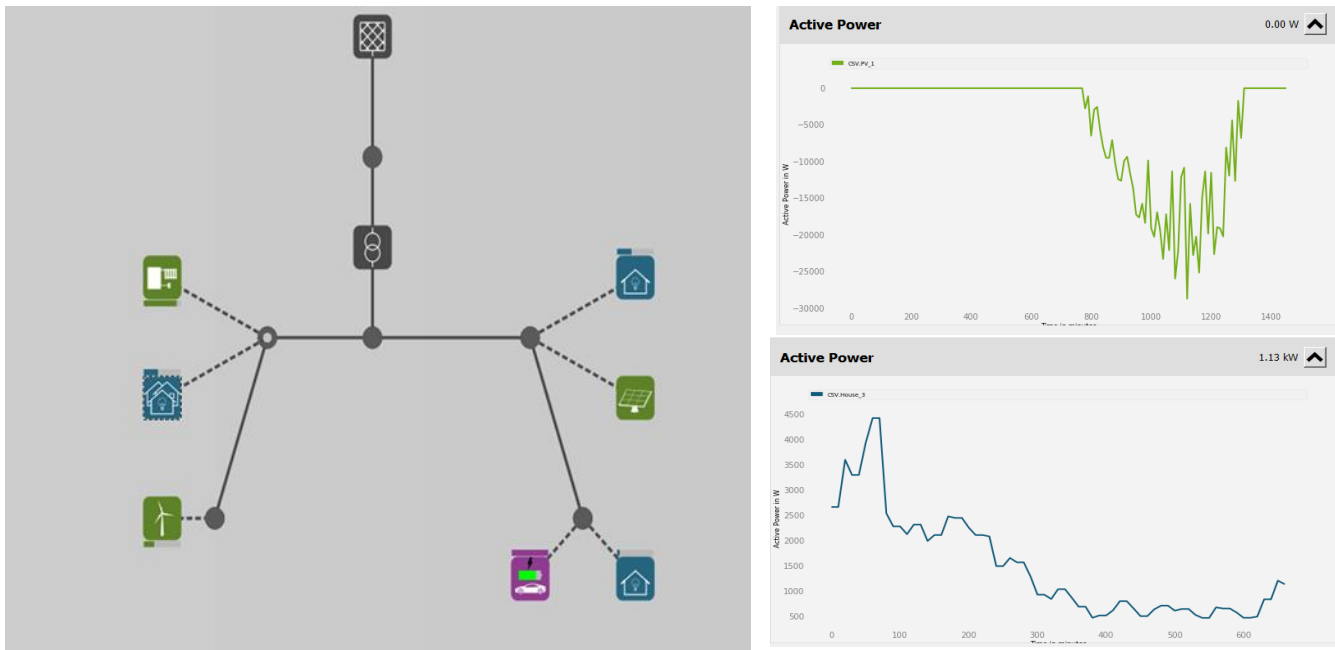


Figure 30 Maverig simulation and load profile diagrams

6.1.2 NS-3

To model a communication network, Ns-3 uses built in classes that represent internet core concepts. Node class refers to computing devices, application class is used to generate simulated activity, channel class is equivalent to the media that information flows through, net device class enables the communication between nodes (NICs) and topology helpers coordinate the common connection tasks required.

As smart grid power grid systems are network connected we can use ns-3 to simulate the behavior of the communication counterpart. Consumers and renewable energy resources be represented by sensors nodes that send power grid related data to RTUs located in system buses. RTUs normally send gathered data to an MTU, the only part of the network simulation that doesn't have a power grid pair.

After importing the required ns3 modules we create the ns-3 node objects that figuratively correspond to the RTUs (client), MTU (server) and the consumers (sensors). The internetstackhelper helps in installing TCP, UDP, IP and other internet related parameters to the nodes. Based on the topology two sensors will be placed in each client and all clients will be star connected to the server node.

```

// Network topology, 5 nodes in a star
/*
      s-n3-s
      |   |   |
      s   n1---n0---n2
      |   |   |
      s   s   s
      |   |   |
      s-n4-s
*/
*/
#include <iostream>
#include <fstream>
#include <string>
#include <cassert>
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("TcpServer");

int
main (int argc, char *argv[])
{
    // Set up some default values for the simulation.
    Config::SetDefault ("ns3::OnOffApplication::PacketSize", UintegerValue (250));
    Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("5kb/s"));
    uint32_t N = 5; //number of client-server nodes

    NS_LOG_INFO ("Create nodes.");
    NodeContainer serverNode;
    NodeContainer clientNodes;
    NodeContainer sensorNodes;
    sensorNodes.Create(2*N-2);
    serverNode.Create (1);
    clientNodes.Create (N-1);
    NodeContainer allNodes = NodeContainer (serverNode, clientNodes, sensorNodes);

    // Install network stacks on the nodes
    InternetStackHelper internet;
    internet.Install (allNodes);
}

```

Figure 31 Network topology and node creation

Using the corresponding point to point topology helpers with create the channels enable the communication with the net device and assign IP addresses to both sensor-client and client-server channels. The rest of the code generates application layer simulated activity and runs the simulation.

```

// sensor-client channel and IP addressing

PointToPointHelper myptp;
myptp.SetDeviceAttribute("DataRate", StringValue ("5Mbps"));
myptp.SetChannelAttribute ("Delay", StringValue ("2ms"));
NetDeviceContainer mydev;
NetDeviceContainer mydev2;
for(uint32_t i=0; i<N-1; i++)
{
    mydev = myptp.Install (sensorNodes.Get(i),clientNodes.Get(i));
    Ipv4AddressHelper myadd;
    std::ostringstream subnet;
    subnet<<"10.1."<<i+1<<".0";
    myadd.SetBase (subnet.str().c_str(), "255.255.255.0");
    Ipv4InterfaceContainer interfaces = myadd.Assign (mydev);
    mydev2 = myptp.Install (sensorNodes.Get(N+i-1),clientNodes.Get(i));
    Ipv4AddressHelper myadd2;
    std::ostringstream subnet2;
    subnet2<<"10.1."<<i+2<<".0";
    myadd2.SetBase (subnet2.str().c_str(), "255.255.255.0");
    Ipv4InterfaceContainer interfaces2 = myadd2.Assign (mydev2);
}

```

Figure 32 Channel creation and IP addressing

6.1.3 ASTORIA Co-simulation

ASTORIA is based on the integration of mosaik’s power related simulators to ns-3. The previous scenario is based on the exchange of simple TCP packets between the nodes. ASTORIA took advantage of the extensibility of ns-3 to develop SCADA related communication protocols like Modbus and DNP3 and the behavior of MTU, RTU and field devices, features not previously available within ns-3 framework.

Ns3 topology is passed to mosaik through an xml file using the networkx package for its creation and configuration. Mosaik handles simulation execution, the connection between them, it matches simulation components and forwards power grid related data to the communication network. Using the simulator manager, the mosaik-api connects to a running ns-3 instance to a predefined host:port pair in simulation configuration description. Scenario parameters for ns-3 are configured in the scenario creation function in the main mosaik script.

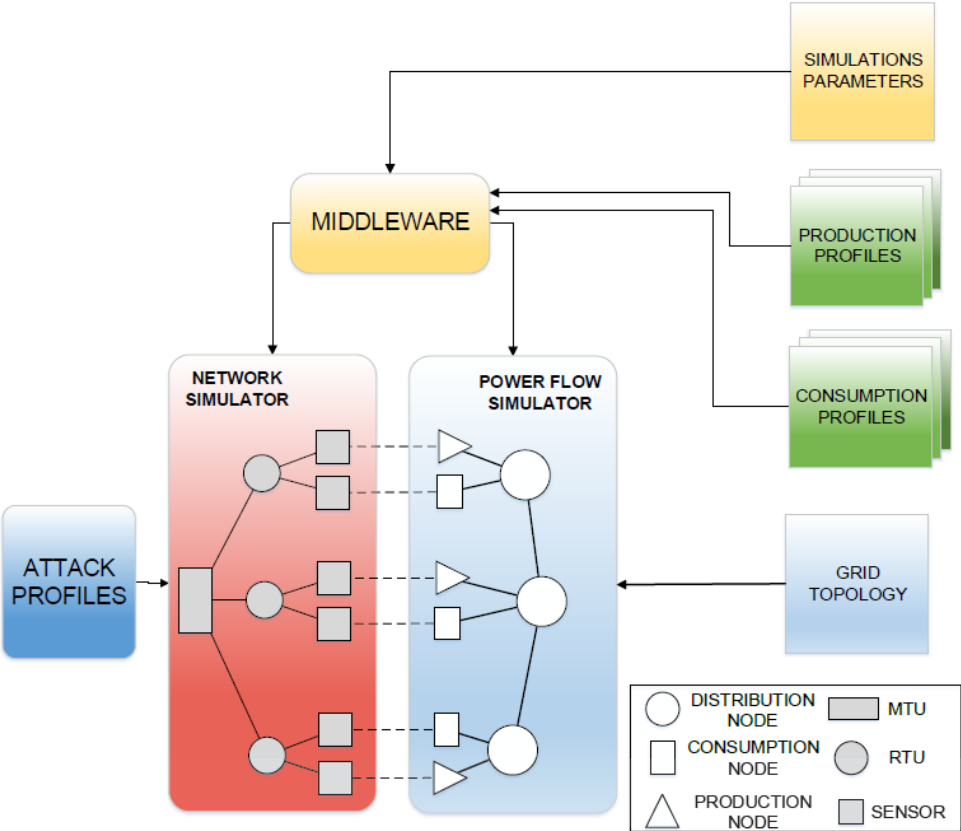


Figure 33 Astoria framework architecture, extracted from [62]

6.1.4 Attacks

ASTORIA makes use of an attack surface like technique as a cyber-attack modeling guide. The topology of the system is mapped by SCADA systems, field devices and the communication network, which constitute a modern smart grid environment. Power grid legacy components, their connection to the internet and the common protocols adopted and their vulnerabilities are enough to give an overview of the possible paths that an adversary can take to compromise or gain access to a smart grid network device. Main attacks supported are DoS attacks against SCADA devices, man in the middle attacks and false data injection by modifying sensor measurements

	Confidentiality	Integrity	Availability
Control Center	Malicious Software, Phishing, Ping Sweeps, Port Scanning, Spyware	Malicious Software, SQL Injection, Unauthorized Access, Spoofing, Replay	Denial of Service, Malicious Software
RTU	Malicious Software, Ping Sweeps, Port Scanning	Malicious Software, Replay, Spoofing, Unauthorized Access,	Denial of Service, Malicious Software
Field Devices	Denial of Service	Denial of Service, Replay	Denial of Service, Buffer Overflow
Communication Protocols	Eavesdropping, Sniffing	Man-in-the-Middle, Sniffing	Denial of Service, Replay

Figure 34 Attacks supported by Astoria, extracted from [62]

Two attacks evaluated by ASTORIA team with an outer purpose of evaluating the impact on operational and financial losses. The first is a malware injection simulation by adding a component that changes the actual sensor measurement data sent to the MTU to a 10% of the actual consumption. The second is a buffer overflow due to a DoS attack that stops RTUs from sampling sensor related data. A small input queue was implemented to RTU simulated components based on an actual buffer overflow vulnerability in SCADA systems reported in 2014.

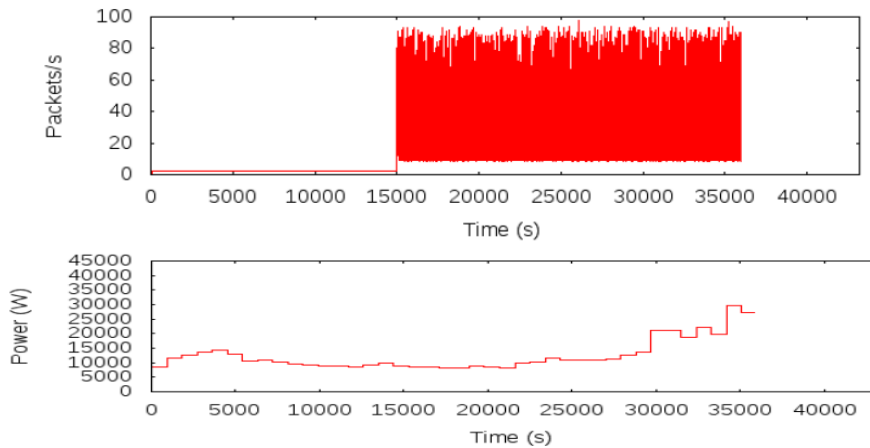


Figure 35 RTUs stop their operation after DoS attack, extracted from [62]

6.2 NeSSi²

[63] represents a distributed Denial-of-Service (DDoS) attack against a real like simulated smart grid topology consisting of smart meters, power plants and utility servers. The paper focuses on the AMI nature of the smart grid and specifically on smart meters and the vulnerabilities introduced due to the power and ICT integration. An attack surface like technique that focuses on AMI and IT is followed here, rendering hardware (smart meters, RTUs), software (protocols), networks and the possible exploitation of their vulnerabilities as the possible attack vectors. NeSSi2 is used to evaluate the results of a large-scale DoS attack of multiple coordinated attack sources that exploits common Internet vulnerabilities.

Before diving into the actual simulation process the paper provides an overview of the real aspects of a DDoS attack. The main categories of are briefly mentioned. These are SYN, ICMP and UDP flooding that exploit the three-way-handshake, configuration errors in network devices and the simplicity of UDP packets correspondingly to waste network resources and attacks that exploit vulnerabilities in network protocols or software to disrupt service normality. Finally, widely used tools in real DDoS examples like Low Orbit Ion Cannon (LOIC) and its upgraded version HOIC are briefly explained.

The NeSSi2 scenario simulates a UDP flooding DDOS attack by sending a large number of UDP packets to random ports on the victim host. The victim is forced to send ICMP replies for those packets rendering it unreachable to legitimate requests. Spoofing the source IP address provides anonymity and ensures that ICMP replies do not flood the attacker.

The energy network consists of a real life adopted scenario that includes five low voltage sub networks of 1kV. Each network follows an open ring topology that can be isolated in fault cases and consists of 10 households running a smart home application. Respectively, the IP network consists of five sub-networks with 10 hosts that represent the ICT part of the smart grid.

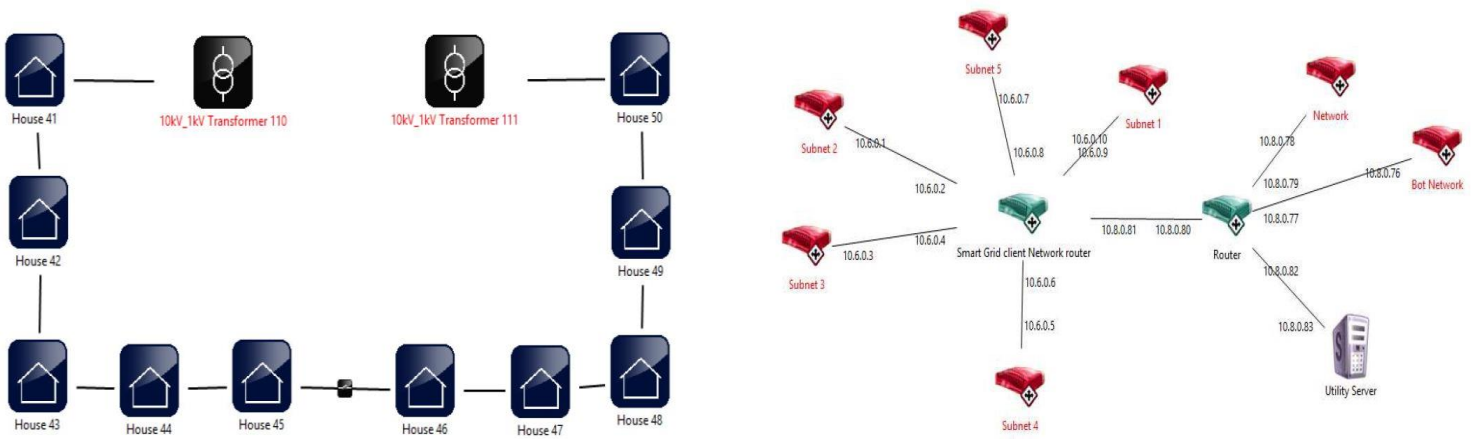


Figure 36 Energy subnetwork and IP network in NeSSi2, extracted from [63]

In order to perform a DDoS attack on the utility server, it is connected to a bot network that consists of zombie hosts controlled by the attacker that will be used to anonymously send large volumes of data to the server victim. The NeSSi2 profiles used are client and server which run standard UDP and energy failure applications and a bot profile that runs the DDoS application.

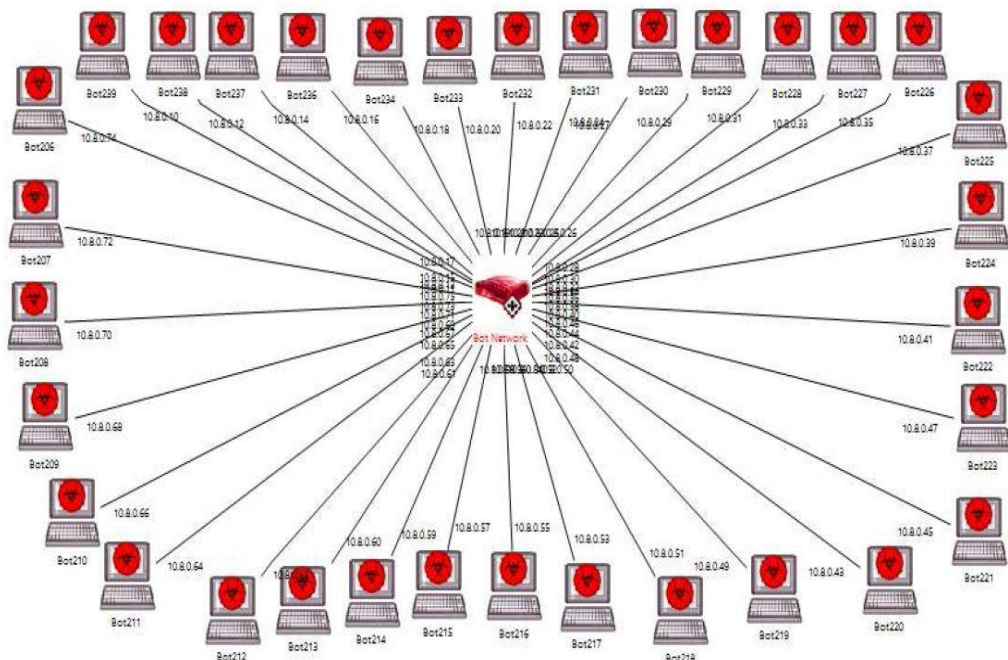


Figure 37 The bot network, extracted from [63]

The simulation runs for the default NeSSi2 duration of 1000 discrete time units. The DDoS attack is set to start at tick 300 and at tick 500 the server starts dropping all incoming packets. The energy produced is equal to the requirements sent by the utility server and the lack of data results in a stop of electricity production by the power plant.

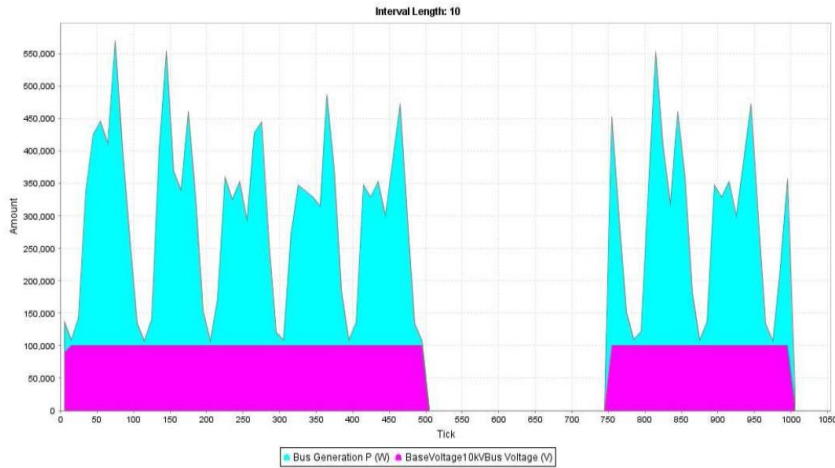


Figure 38 power plant halts after DDoS attack, extracted from [63]

6.3 SCORE

6.3.1 Overview

Smart-Grid Common Open Research Emulator (SCORE) [64], based on CORE, is the first attempt to emulate an integrated power and communication network. Its main purpose is to provide a middle ground between a real testbed and a co-simulator that can be easily ported to real smart grid devices. Its main features are: fidelity with the use of both power and communication modules, portability of the virtual nodes that run the smart grid applications to real systems and scalability due to distributed emulation capabilities.

Score is based on CORE's structure providing additional modules and configurations for the power grid. The session related daemon, related services, virtual nodes and network configuration parameters are started using the GUI. Emulated nodes are individual virtual or real device instances that either way represent smart grid applications that one can interact with through a Linux like shell.

The communication network emulation inherits all CORE characteristics allowing virtualized nodes to be connected to physical devices. Power flows, loads, renewable energy sources are also implemented through SCORE power module, which also handles grid topology, configurations and is updated through interactions with the communication network.

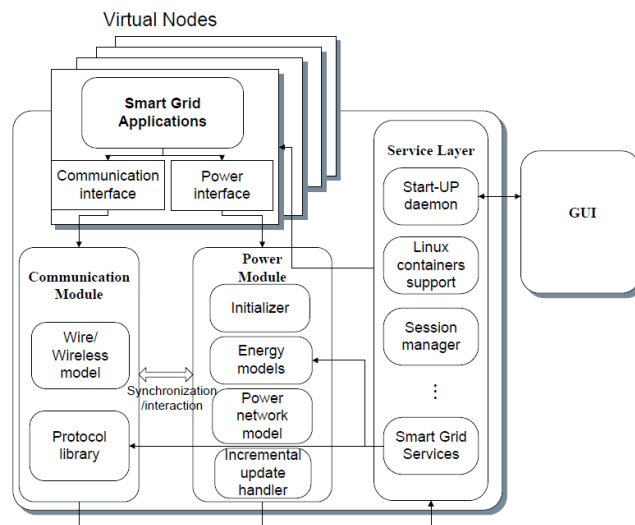


Figure 39 SCORE architecture, extracted from [64]

6.3.2 Attacks on a SCORE emulator

In [65] SCORE is used in a testbed capable of being extended with cyber-attack execution capabilities. The testbed consists of two Linux virtual machines, an Ubuntu that serves as the target machine that runs the smart grid emulation and a Kali Linux penetration testing platform that corresponds to the attacker.

A typical SCORE smart grid scenario consists of a power plant and a number of houses represented by an intelligent switch that different nodes are attached to. These can be loads, storage devices or renewable energy resources.

After downloading and installing CORE and SCORE, we start the score-daemon, navigate to the folder where score is stored and start the GUI.

```
ubuntu@ubuntu-VirtualBox:~/Desktop/scoreplus$ sudo /etc/init.d
/scoreplus start
[sudo] password for ubuntu:
* Starting SCOREPLUS service
ubuntu@ubuntu-VirtualBox:~/Desktop/scoreplus$ ./scoreplus
Connecting to "cored" (127.0.0.1:4038)...connected.
```

Figure 40 Starting score

Scoreplus comes up with some predefined inn topologies. One of them is a smart grid scenario with six houses with an intelligent switch in each one, all connected to a solar panel, a wind turbine, two loads and a storage device. Nodes can be connected through power (red) or link (green) ethernet lines depending on their type and the scenario

needs. Main configurations and services are automatically handled by SCORE. We add to the scenario a power supply and a GRE tunnel connection to one of the nodes that will allow us to connect the attacker with the target system. 192.168.1.152 is the IP of the CORE host, 192.168.1.153 the IP of Kali Linux and 10.0.2.2 the IP it gets when we tunnel it to the emulation network.

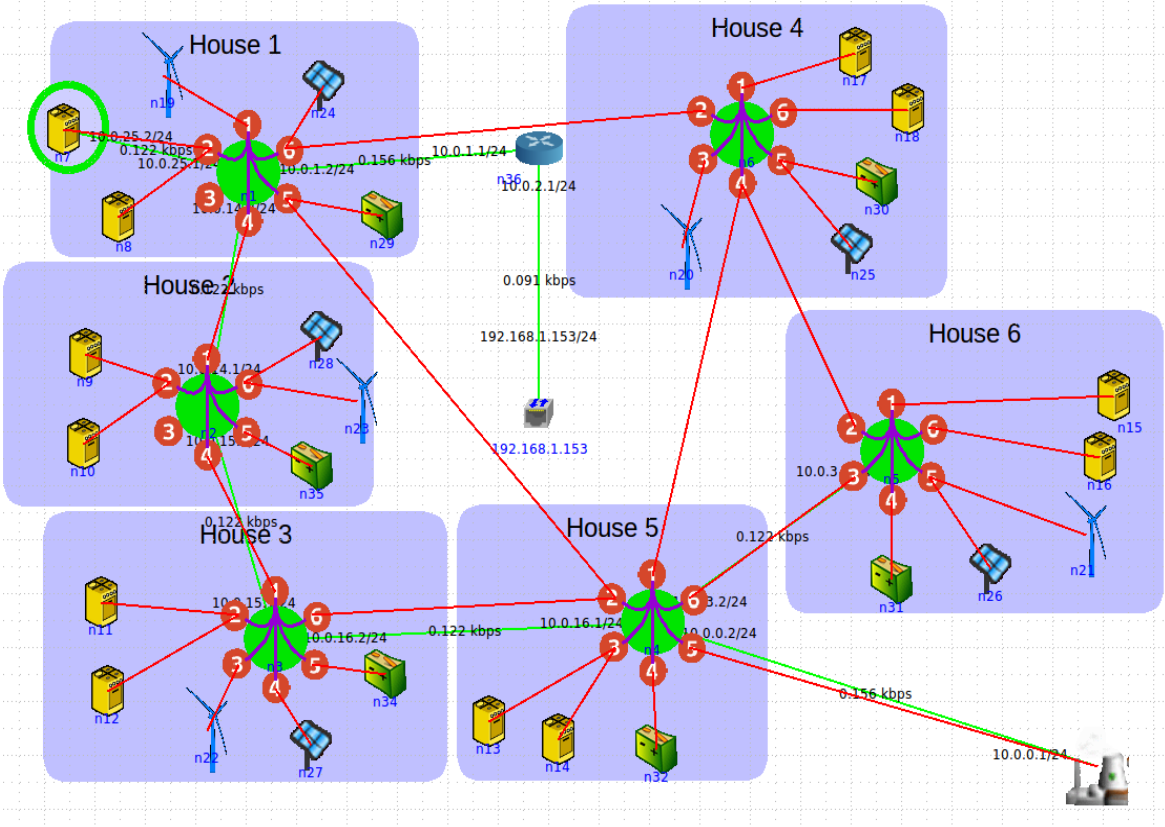


Figure 41 SCORE scenario

```

root@kali:~# ip link add gt0 type gre tap remote 192.168.1.152 local 192.168.1.153
root@kali:~# ip addr add 10.0.2.2/24 dev gt0
root@kali:~# ip link set dev gt0 up
root@kali:~#

```

Figure 42 GRE tunnel setup

DoS attack

Using Kali Linux as the attack platform means that a kill chain model approach is followed. Nmap GUI is used as a reconnaissance tool to gather information about the smart grid network. We limit the IP range to those needed in order to dramatically reduce execution time. Scan results include an IP address list of live hosts and a short description of them, open ports, services and a topology related graph of the network.

```
nmap -T4 -A -v 10.0.0-20.0-10
```

Starting Nmap 7.60 (<https://nmap.org>) at 2018-01-06 16:12 EET
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Initiating Ping Scan at 16:12
Scanning 220 hosts [4 ports/host]

	Port	Protocol	State	Service	Version
✓	2601	tcp	open	zebra	Quagga routing software
✓	2604	tcp	open	zebra	Quagga routing software

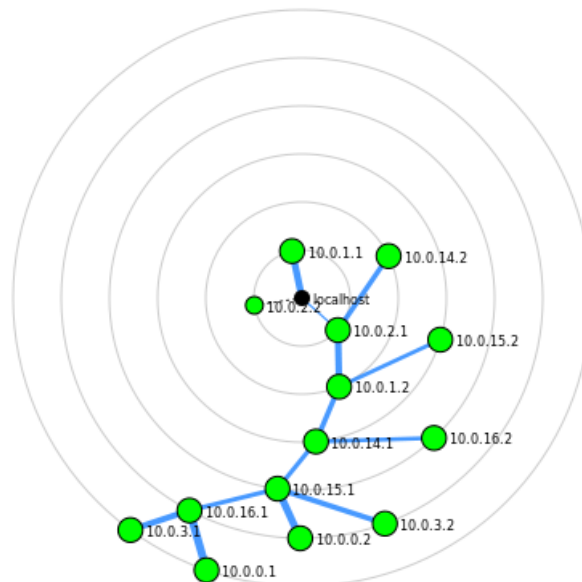


Figure 43 Zenmap results

For the weaponization process, the tool used was a free packet generator and analyzer for TCP/IP protocol called hping3. The following command sends to the power supply 100000 SYN packets with a window size of 64 and a size of 120 to port 2601 discovered by Nmap as fast as possible without waiting for replies in flood mode. We also use rand source to hide the source IP address.

```
root@kali:~# hping3 -c 100000 -d 120 -S -w 64 -p 2601 --flood --rand-source 10.0.0.1
HPING 10.0.0.1 (gt0 10.0.0.1): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.0.1 hping statistic ---
9926217 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~#
```

Figure 44 DoS with hping3

During the SYN flood attack the throughput jumps to extremely high values as the attacker sends packets repeatedly. As the victim tries to respond and the attacker doesn't

close down the connection an overflow of half-open three-way handshakes is created that can deny the service to other nodes due to network congestion. We can see that when we try to ping the target during the DoS attack a high percentage of the packet is lost or even the target host is unreachable.

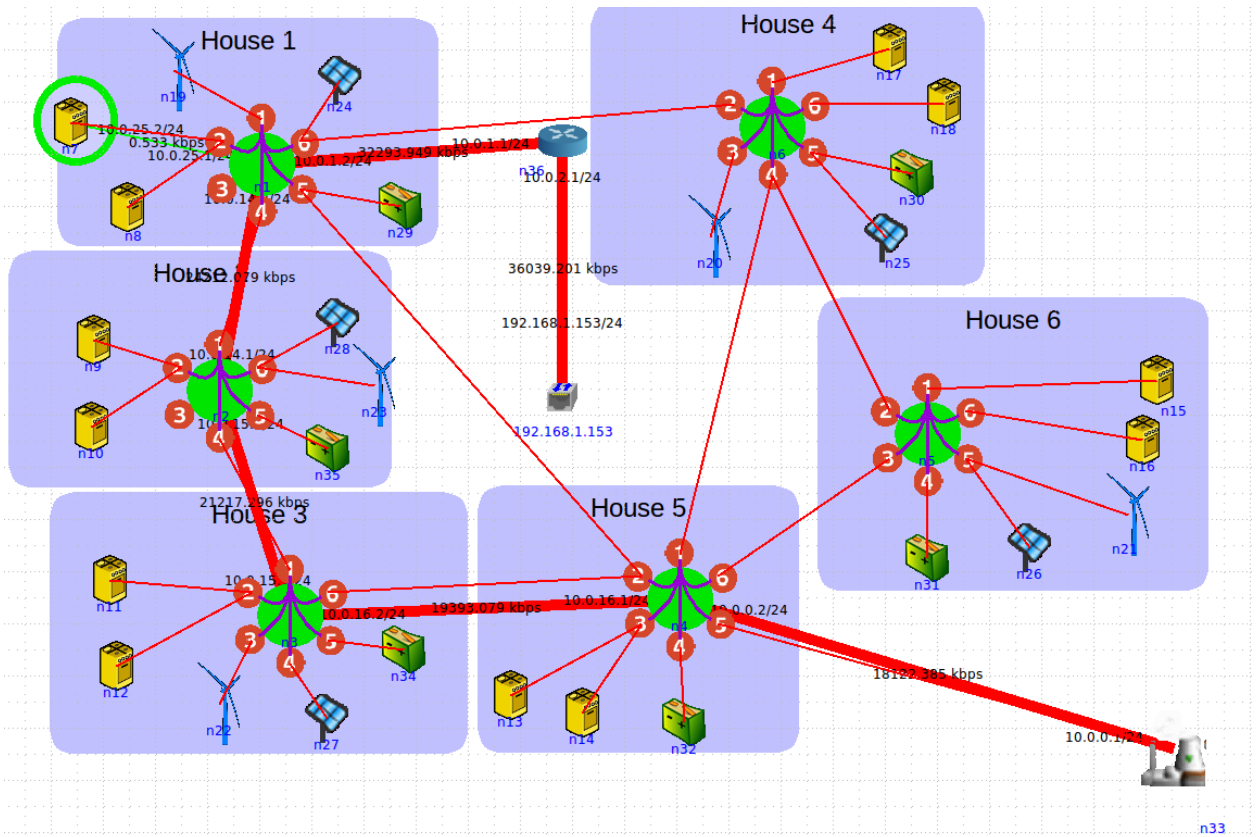


Figure 45 High throughput during SYN flood

```

root@n5: /tmp/pycore.38612/n5.conf# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=9 ttl=63 time=125 ms
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
ping: sendmsg: Network is unreachable
^C
--- 10.0.0.1 ping statistics ---
32 packets transmitted, 1 received, 96% packet loss, time 31711ms
rtt min/avg/max/mdev = 125.396/125.396/125.396/0.000 ms
root@n5: /tmp/pycore.38612/n5.conf#

```

Figure 46 Ping during attack

MITM suggestion

A man in the middle attack is also a network-based attack that could be performed in the previous smart grid scenario. The purpose of this attack is often to fool the victim into associating the MAC address of the default gateway with the attacker's MAC address in order to intercept and modify the traffic meant for it. A common example in smart grid is the capture and modification of meter data from RTU gateways to control centers. As SCORE doesn't provide any SCADA emulation software and traffic here is a simple analogy between a gateway and a host victim.

We enable packet forwarding in the network system processes for our Kali Linux machine to let it act as a router. We set up arpspoof to capture traffic received and sent traffic and send it to the attacker machine and display the sniffed ICMP echo packets with tcpdump.

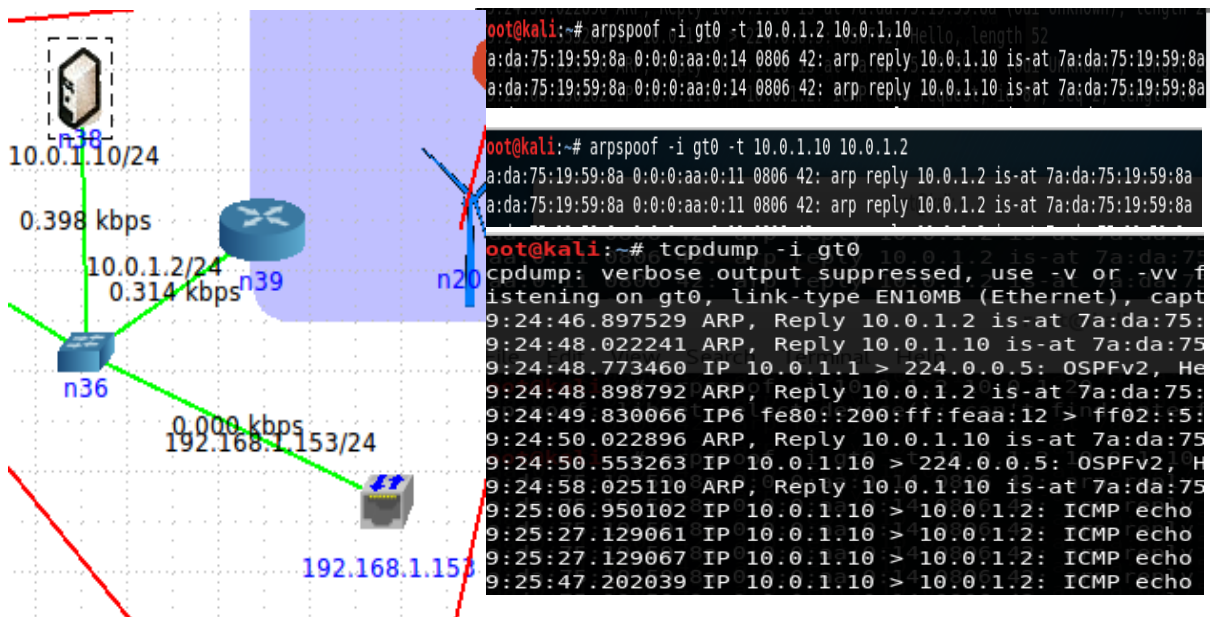


Figure 47 ARP spoofing attack

6.4 GNS3

6.4.1 Modeling approach

In [66] a DoS attack against a smart grid communication network modeled environment is studied. The paper focuses on PMUs and the measurement data they continuously send to PDCs for synchronization, monitor and control of the power grid in real time. A combination of an attack surface and a kill chain model is followed here. The former identifies all the possible attack vectors on PMUs and generally on wide area monitoring systems and the latter helps in appropriate selection of attack tools. Synchrophasor networks vulnerabilities vary from physical, denial of service to malicious data injection, data spoofing and man in the middle attacks. The tools used

are GNS3 network emulator, openPDC, an open source packet data concentrator along with the PMU connection tester to send PMU sample data between hosts and Low Orbit Ion Cannon (LOIC), a tool that is commonly used for Denial of Service attacks.

6.4.2 DoS with GNS3

The following model is a simplified variation of the aforementioned paper, that is used to evaluate a DoS attack against a smart grid communication network environment. 2 VMWare virtual hosts with 2 separate virtual NICs are used and connected to a simplified network topology through the GNS3 emulator. The hosts are 2 windows machines that will use the OpenPDC and the PMU connection packet tester. The network includes typical bandwidth and delays in ethernet and serial links. Switches and hosts are default GNS3 appliances and routers are a c3725 cisco router image.

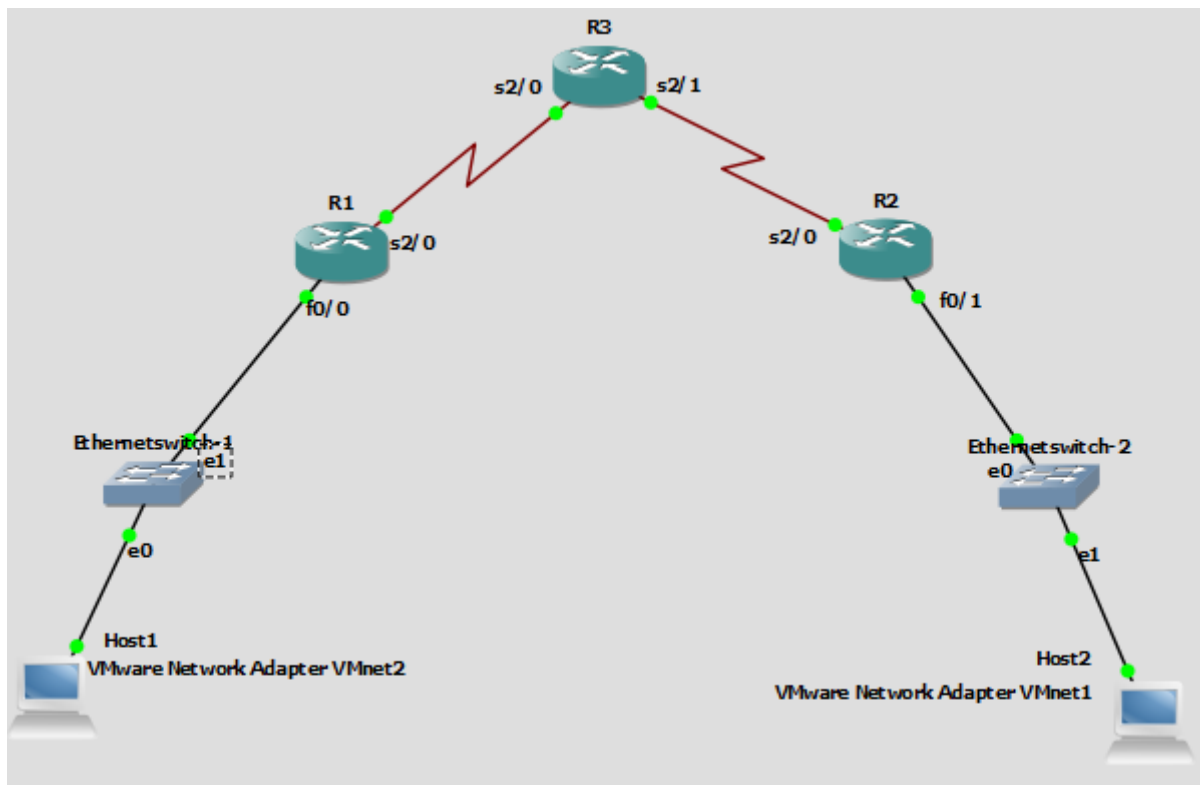


Figure 48 Network topology

All router interfaces are configured, both dynamically and manually where needed and the hosts default gateways are changed to match the corresponding router in the GNS3 network. A simple routing protocol is also configured to announce all the possible IPs in the network. Host 1 has an IP of 192.168.164.129 and host 192.168.52.129. The main differences between the paper and this model is that the paper used a more complex ISP routing topology and GNS3 marketplace appliances instead of the default cloud appliance that was used here for the hosts. The logic however remains the same.

```

R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface f0/0
R1(config-if)#ip address dhcp
R1(config-if)#no shutdown
R1(config-if)#end

*Mar 1 00:02:17.911: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:02:18.911: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1#
*Mar 1 00:02:20.143: %SYS-5-CONFIG_I: Configured from console by console
R1#
*Mar 1 00:02:28.383: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 192.168.164.135,
mask 255.255.255.0, hostname R1
R1#

```

```

R2(config)#int f0/0
R2(config-if)#ip add dhcp
R2(config-if)#no shut
R2(config-if)#end
R2#
*Mar 1 00:03:27.923: %SYS-5-CONFIG_I: Configured from console by console
R2#
*Mar 1 00:03:29.395: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:03:30.395: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2#
*Mar 1 00:03:40.451: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 192.168.52.132,
mask 255.255.255.0, hostname R2

```

Figure 49 Configuration of router interfaces connected to hosts

```

R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface s2/0
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#clock rate 64000
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Mar 1 00:05:28.815: %LINK-3-UPDOWN: Interface Serial2/0, changed state to up
R1(config)#
*Mar 1 00:05:29.819: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
R1(config)#

```

```

R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int s2/0
R2(config-if)#
*Mar 1 00:04:23.563: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
R2(config-if)#ip add 10.1.2.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#end
R2#
*Mar 1 00:04:44.291: %SYS-5-CONFIG_I: Configured from console by console

```

Figure 50 Configuration of ISP related serial interfaces

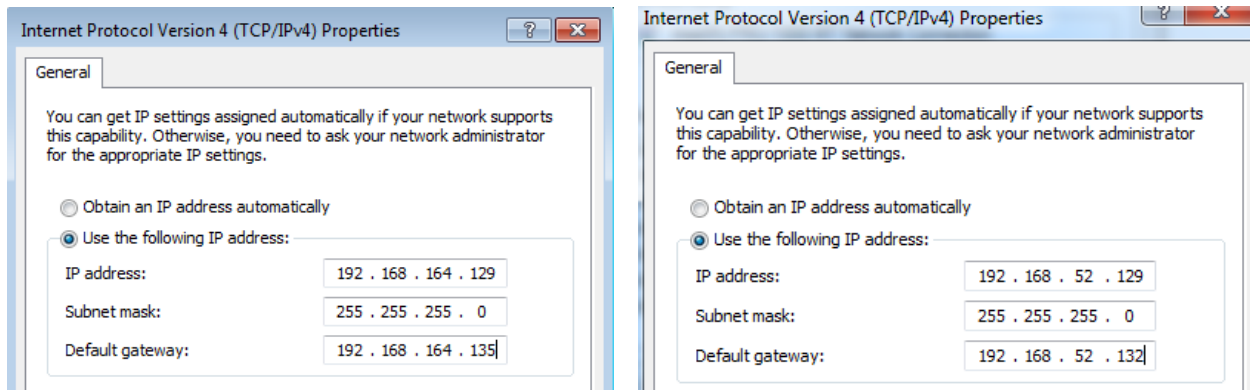


Figure 51 Host configuration

Low Orbit Ion cannon is used to execute a Denial of Service attack to Host 2. Its use is pretty simple by providing a target IP address and a TCP flood as the method of the attack. The flow of the TCP packets which can be observed below, causes huge latencies to the PMU data transmission and can even render the victim unreachable by the network, which is unacceptable for the demand and response and the continuous availability of the power grid services. Sample data used by PMU connection tester where both old IEEE 1344-1995 and their replacement C37.118-2005 standards for synchrophasors for power systems.

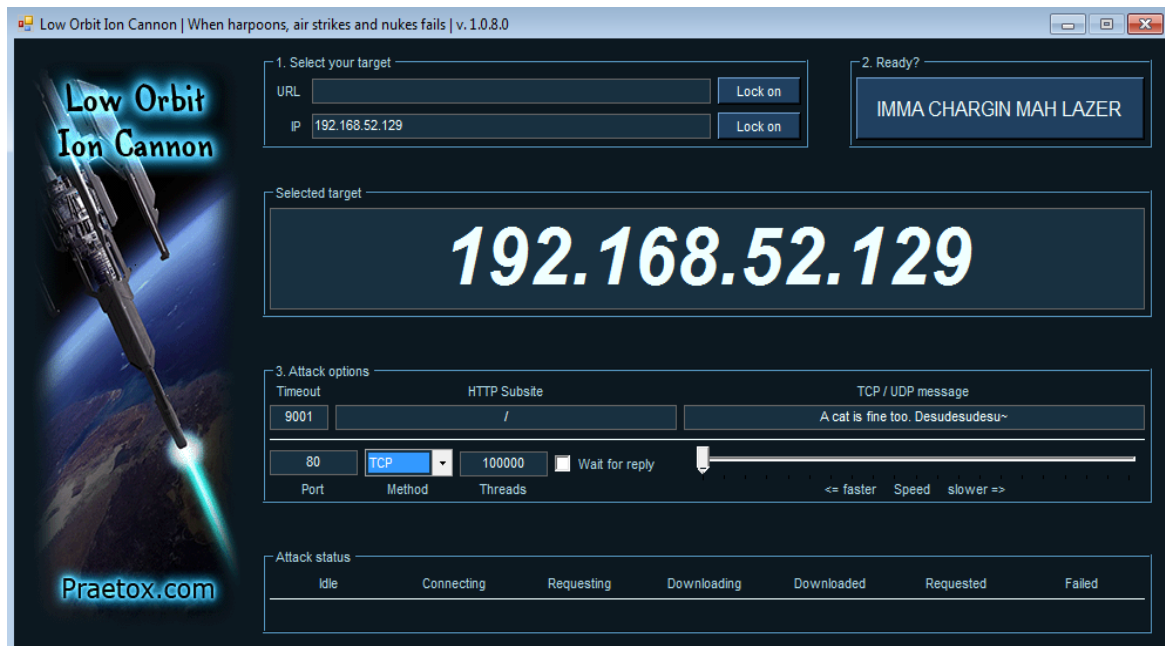


Figure 52 LOIC

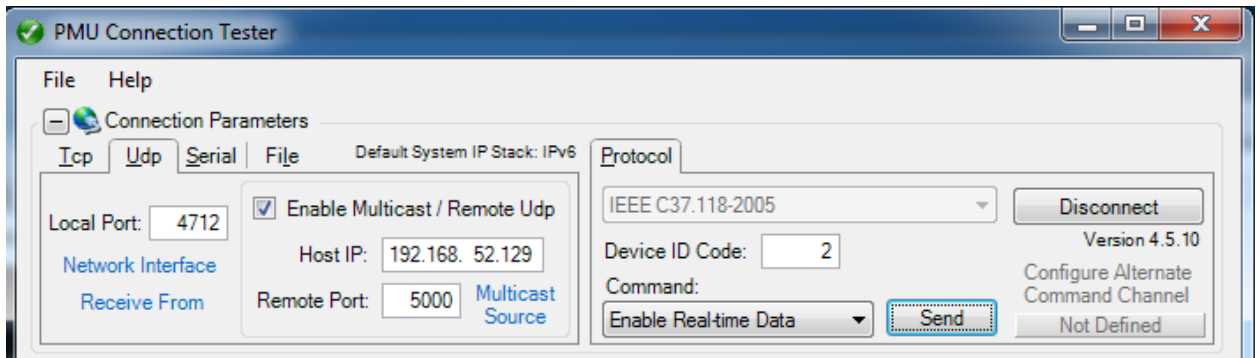


Figure 53 Data sent with the PMU connection tester

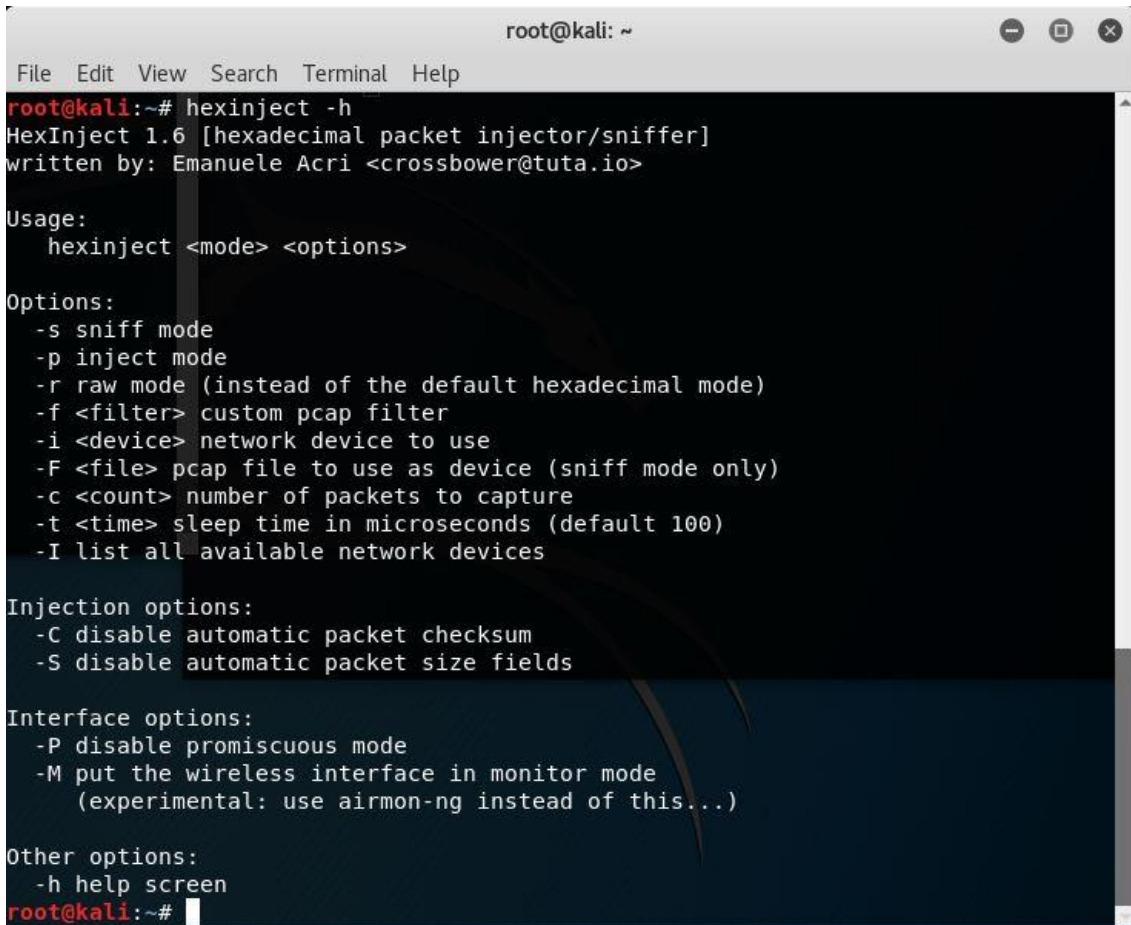
No.	Time	Source	Destination	Protocol	Length	Info
22505	-234.677734	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22506	-234.677683	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22507	-234.677388	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22508	-234.677352	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22509	-234.676681	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22510	-234.676614	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22511	-234.676306	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22512	-234.676241	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22513	-234.676163	192.168.52.129	192.168.164.135	TCP	54	80 → 5...
22514	-234.675791	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22515	-234.675753	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22516	-234.675422	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22517	-234.675383	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22518	-234.674956	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22519	-234.674909	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22520	-234.674647	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22521	-234.674610	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
22522	-234.674359	192.168.164.135	192.168.52.129	TCP	66	[TCP P...
17	24.593618	192.168.164.129	192.168.164.129	ICMP	70	Destination unreachable (Host unreachable)
18	24.604531	Vmware_16:0d:01	c2:01:23:54:00:00	ARP	42	Who has 192.168.164.135? Tell 192.168.164.129
19	24.637378	c2:01:23:54:00:00	Vmware_16:0d:01	ARP	60	192.168.164.135 is at c2:01:23:54:00:00
20	28.259401	fe80::31b9:3974:4a6...	ff02::1:2	DHCPv6	157	Solicit XID: 0x81b5ff CID: 000100012213d9c1000c29e0fb48
21	29.031110	192.168.164.129	192.168.52.129	UDP	58	4712 → 5000 Len=16
22	29.056821	192.168.164.135	192.168.164.129	ICMP	70	Destination unreachable (Host unreachable)
23	29.564894	192.168.52.129	192.168.164.129	ICMP	74	Echo (ping) request id=0x0001, seq=102/26112, ttl=126 (reply i...
24	29.565582	192.168.164.129	192.168.52.129	ICMP	74	Echo (ping) reply id=0x0001, seq=102/26112, ttl=128 (request...
25	29.596861	192.168.164.135	192.168.164.129	ICMP	70	Destination unreachable (Host unreachable)
26	30.896739	192.168.164.129	192.168.52.129	UDP	58	4712 → 5000 Len=16
27	30.920170	192.168.164.135	192.168.164.129	ICMP	70	Destination unreachable (Host unreachable)
28	34.556830	192.168.52.129	192.168.164.129	ICMP	74	Echo (ping) request id=0x0001, seq=103/26368, ttl=126 (reply i...
29	34.557208	192.168.164.129	192.168.52.129	ICMP	74	Echo (ping) reply id=0x0001, seq=103/26368, ttl=128 (request...
30	34.588927	192.168.164.135	192.168.164.129	ICMP	70	Destination unreachable (Host unreachable)

Figure 54 TCP flood results

6.4.3 Hexinject

Hexinject is a packet sniffer and injector that comes built in with Kali Linux. It can be used to sniff and display packets in raw or hexadecimal format on the specified

interfaces. It can be also used to inject and modify network traffic by replacing hexadecimal values of intercepted packets. The basic usage of the tool is shown in the figure below.



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# hexinject -h
HexInject 1.6 [hexadecimal packet injector/sniffer]
written by: Emanuele Acri <crossbower@tuta.io>

Usage:
  hexinject <mode> <options>

Options:
  -s sniff mode
  -p inject mode
  -r raw mode (instead of the default hexadecimal mode)
  -f <filter> custom pcap filter
  -i <device> network device to use
  -F <file> pcap file to use as device (sniff mode only)
  -c <count> number of packets to capture
  -t <time> sleep time in microseconds (default 100)
  -I list all available network devices

Injection options:
  -C disable automatic packet checksum
  -S disable automatic packet size fields

Interface options:
  -P disable promiscuous mode
  -M put the wireless interface in monitor mode
    (experimental: use airmon-ng instead of this...)

Other options:
  -h help screen
root@kali:~#
```

Figure 55 Hexinject usage

A compromised node or a weakness in network encryption mechanisms that has often been observed in many known attacks, renders network traffic susceptible to sniffing and modification. A simple Hexinject command is used to change the legitimate destination IP address of the PMU data preventing them from reaching the PDC. Wireshark usage confirms that the packets have indeed been modified.



```
root@kali:~# hexinject -s -i eth0 -f 'udp' | replace "C0 A8 34 81" "08 08 08 08"
| hexinject -p -i eth0
```

Figure 56 data spoofing with hexinject

2422	2220.247498	192.168.164.129	192.168.52.129	UDP	60 4712 → 5000 Len=18
2426	2226.902387	192.168.164.129	192.168.52.129	UDP	60 4712 → 5000 Len=18
2427	2227.293605	192.168.164.129	192.168.52.129	UDP	60 4712 → 5000 Len=18
2429	2230.365609	192.168.164.129	8.8.8.8	UDP	60 4712 → 5000 Len=18
2430	2230.365609	192.168.164.129	8.8.8.8	UDP	60 4712 → 5000 Len=18
2431	2230.366557	192.168.164.129	8.8.8.8	UDP	60 4712 → 5000 Len=18
2434	2230.366557	192.168.164.129	8.8.8.8	UDP	60 4712 → 5000 Len=18
2435	2230.366557	192.168.164.129	8.8.8.8	UDP	60 4712 → 5000 Len=18
2436	2230.409512	192.168.164.135	192.168.164.129	ICMP	70 Destination unreachable (Host unreachable)
2437	2230.420437	192.168.164.135	192.168.164.129	ICMP	70 Destination unreachable (Host unreachable)
2438	2230.430470	192.168.164.135	192.168.164.129	ICMP	70 Destination unreachable (Host unreachable)
2439	2230.463406	192.168.164.135	192.168.164.129	ICMP	70 Destination unreachable (Host unreachable)
2440	2230.474368	192.168.164.135	192.168.164.129	ICMP	70 Destination unreachable (Host unreachable)

Figure 57 data spoofing with hexinject

6.5 Summary

This section included a brief look on the most considerable modeling techniques on smart grid networks. The proposed frameworks included both co-simulation (ASTORIA, NeSSi2) and emulation (SCORE, GNS3) approaches. ASTORIA and SCORE were tested in an Ubuntu environment while NeSSi2 and GNS3 on a Windows 10 host. GNS3 is the most versatile tool and can be used in most common operating systems. The co-simulation approaches provide both power and communication network components within their frameworks. ASTORIA uses mosaik and its integrated simulators to simulate power grid features and to synchronize them with the NS3 simulator, used for the ICT components. NeSSi2 has its own built in power and network simulators, implemented through a structured method on the frontend GUI and running as backend processes. Emulators follow an alternate approach. SCORE uses built in power modules for the power grid components and CORE virtualized network architecture for the ICT part. GNS3 scenario used external software to send sample PMU data between hosts through networks modeled with pre-included or installed appliances. ASTORIA tested a variety of attacks including malicious software, DoS, eavesdropping, sniffing while the other tools focused on DoS. Man-in-the-middle and data spoofing was also shown to be a possible scenario. Simulators used built-in (NeSSi2) or manually defined (ASTORIA) components as attack tools while emulators focused on tools of Kali Linux penetration testing platform. GNS3 scenario also included windows hosts that ran LOIC and PMU data software. Simulators test area was the distribution power grid, SCORE attempted an attack on a wind generator and GNS3 on PMUs that are located in both transmission and distribution power grid. Simulators evaluate the impact of the attack by running the scenario while emulators allow for real time interaction with the modeled network. Attack surface prevailed as the modeling technique along with the kill chain model approach in the emulation scenarios that was necessary for the appropriate attack tool selection. GNS3 project seems to be having the best support among the platforms and SCORE the lowest as it has been abandoned by its founders. The table below provides an overview of this section highlights.

Simulator /Attribute	ASTORIA	NeSSi²	SCORE	GNS3
Type	Co-simulation	Co-simulation	Emulation	Emulation
OS	Ubuntu Linux	Windows	Ubuntu Linux	Windows/Linux/iOS
Power components	Mosaik simulators	Built in discrete simulators	SCORE power modules	PMU/PDC software
Network components	NS3	Built in discrete simulators	CORE virtual machines	GNS3 appliances
Attacks tested	Malicious software/DoS/ Eavesdropping	DDoS	DoS/ARP spoofing	DoS/data spoofing
Tools used	Attack profiles	DoS appliance	Hping/arpspoof	LOIC/hexinject/ OpenPDC/PMU connection tester
External/virtual components	*	*	Kali Linux/GRE tunnel	Kali Linux/windows VMWare hosts
License	Open source	Open source	Open source	Open source
Grid test area	Distribution	Distribution	Generation	Transmission/ Distribution
Scalability	High	Medium/High	Medium	Medium
Evaluation	By impact	By impact	Real time interaction	Real time interaction
Modeling techniques	Attack surface	Attack surface	Kill chain	Kill chain/attack surface
Support	Medium	Low	None	High
Language	Python/C++	Java	Python	Python

7 Conclusions

The smart grid is an integrated power and communication network infrastructure with an ideal of efficient use of renewable energy resources, real time monitor and automation, demand response strategies and billing participation of the consumers. The use of standard communication protocols however, albeit the conveniences it may offer, poses also a great threat by introducing the common vulnerabilities of the ICT infrastructure. This thesis included an overview of the smart grid power and network architecture indicating its main security issues. The complex nature of the smart grid ecosystem sparked interest for simulation attempts. An overview of both co-simulation and emulation platforms was consequently described with an eye towards security related modeling attempts. To model, understand and mitigate a cyber-attack in an efficient way, the main cyber-attack modeling techniques were subsequently mentioned. The last section included an analysis of the most considerable efforts up to date, to model a cyberattack against smart grid networks, both on co-simulation and emulation, an overview of their tools and some extra possible attacks scenarios. Being one of the critical infrastructures smart grid cybersecurity is an important factor that both simulation and emulation approaches can provide useful information on past incident analysis and future attack avoidance and mitigation strategies.

Bibliography

- [1] Momoh, J. A. (2012). *Smart grid: fundamentals of design and analysis*. Hoboken (NJ): Wiley.
- [2] Mets, K., Ojea, J. and Devellder, C. (2014). Combining Power and Communication Network Simulation for Cost-Effective Smart Grid Analysis. *IEEE Communications Surveys & Tutorials*, 16(3), pp.1771-1796.
- [3] SCE-Cisco-IBM SGRA Team. (2012). Smart Grid Reference Architecture Volume 1. Using Information and Communication Services to Support a Smarter Grid
- [4] Richter, A., Laan, E. V., Ketter, W., & Valogianni, K. (2012). Transitioning from the traditional to the smart grid: Lessons learned from closed-loop supply chains. *2012 International Conference on Smart Grid Technology, Economics and Policies (SG-TEP)*. doi:10.1109/sg-tep.2012.6642382
- [5] Bayindir, R., Hossain, E., & Vadi, S. (2016). The path of the smart grid -the new and improved power grid. *2016 International Smart Grid Workshop and Certificate Program (IS-GWCP)*. doi:10.1109/isgwcp.2016.7548270
- [6] Suyan, L., Yanmin, G., & Jun, X. (2014). Research and application of smart grid cross regional trading control technology. *2014 International Conference on Information Science, Electronics and Electrical Engineering*. doi:10.1109/infosee.2014.6947798
- [7] Ali, A. B. (2013). *Smart Grids Opportunities, Developments, and Trends*. London: Springer.
- [8] Knapp, E. D., & Samani, R. (2013). What is the Smart Grid? Applied Cyber Security and the Smart Grid, 1-15. doi:10.1016/b978-1-59749-998-9.00001-3
- [9] H. Farhangi, "The path of the smart grid," in *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18-28, January-February 2010. doi: 10.1109/MPE.2009.934876
- [10] S. McLaughlin et al., "The Cybersecurity Landscape in Industrial Control Systems," in *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039-1057, May 2016. doi: 10.1109/JPROC.2015.2512235
- [11] Tsang, Rose. (2010). *Cyberthreats, Vulnerabilities and Attacks on SCADA Networks*.
- [12] Siemens AG. Smart grid division. (2011). *Communications network solutions for smart grids*
- [13] V. C. Gungor et al., "Smart Grid Technologies: Communication Technologies and Standards," in *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529-539, Nov. 2011. doi:10.1109/TII.2011.2166794
- [14] European Union Agency for Network and Information Security. (2016). *Communication network interdependencies in smart grids*. doi:10.2824/949547

- [15] Ho, Q., & Le-Ngoc, T. (2013). Smart Grid Communications Networks: Wireless Technologies, Protocols, Issues, and Standards. Handbook of Green Information and Communication Systems, 115-146. doi:10.1016/b978-0-12-415844-3.00005-x
- [16] Boyer, W. F., & Mcbride, S. A. (2009). Study of Security Attributes of Smart Grid Systems-Current Cyber Security Issues. doi:10.2172/957512
- [17] Aouini, Imen & Azzouz, Lamia. (2015). Smart Grids cyber security issues and challenges. International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, vol 9
- [18] Dong Wei, Yan Lu, M. Jafari, P. Skare and K. Rohde, "An integrated security system of protecting Smart Grid against cyber attacks," 2010 Innovative Smart Grid Technologies (ISGT), Gaithersburg, MD, 2010, pp. 1-7. doi: 10.1109/ISGT.2010.5434767
- [19] Aloul, F., Al-Ali, A. R., Al-Dalky, R., Al-Mardini, M., & El-Hajj, W. (2012). Smart Grid Security: Threats, Vulnerabilities and Solutions. International Journal of Smart Grid and Clean Energy, 1-6. doi:10.12720/sgce.1.1.1-6
- [20] Saed, M. Smart Grid Security Concepts and Issues.
- [21] European Union Agency for Network and Information Security. (2013) Smart Grid Threat Landscape and Good Practice Guide
- [22] European Union Agency for Network and Information Security. (2012). Smart Grid Security. Annex II. Security aspects of the smart grid
- [23] Ghansah I., Smart grid cyber security potential threats, vulnerabilities and risks, PIER Energy-Related Environmental Research Program, CEC-500-2012-047, California Energy Commission, Sacramento, CA, pp. 1-93, 2012
- [24] W. Li, M. Ferdowsi, M. Stevic, A. Monti and F. Ponci, "Cosimulation for Smart Grid Communications," in IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2374-2384, Nov. 2014. doi: 10.1109/TII.2014.2338740
- [25] Garofalakis, J.D., Leligou, H., Sarakis, L., Tsampasis, E., & Zahariadis, T.B. (2016). Novel Simulation Approaches for Smart Grids. J. Sensor and Actuator Networks, 5, 11.
- [26] K. Mets, J. A. Ojea and C. Davelder, "Combining Power and Communication Network Simulation for Cost-Effective Smart Grid Analysis," in IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1771-1796, Third Quarter 2014. doi: 10.1109/SURV.2014.021414.00116
- [27] Hua Lin, S. Sambamoorthy, S. Shukla, J. Thorp and L. Mili, "Power system and communication network co-simulation for smart grid applications," ISGT 2011, Hilton Anaheim, CA, 2011, pp. 1-6. doi: 10.1109/ISGT.2011.5759166
- [28] R. (2017, December 03). Rwl/PYPOWER. Retrieved February 19, 2018, from <https://github.com/rwl/PYPOWER>
- [29] OpenDSS. Retrieved February 19, 2018, from <https://sourceforge.net/p/electricdss/wiki/Home/>
- [30] GridLAB-D. Retrieved February 19, 2018, from <https://sourceforge.net/projects/gridlab-d/>
- [31] Ns-3. Retrieved February 19, 2018, from <https://www.nsnam.org/>

- [32] OMNet 5.2.1 Released. Retrieved February 19, 2018, from <https://www.omnetpp.org/>
- [33] Abdalkarim Awad, Peter Bazan, and Reinhard German. 2017. A Short Tutorial On Using SGsim Framework For Smart Grid Applications. In proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools on 10th EAI International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'16), Antonio Puliafito, Kishor S. Trivedi, Bruno Tuffin, Marco Scarpa, Fumio Machida, and Javier Alonso (Eds.). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 143-148. DOI: <https://doi.org/10.4108/eai.25-10-2016.2267055>
- [34] D. Bhor, K. Angappan and K. M. Sivalingam, "A co-simulation framework for Smart Grid wide-area monitoring networks," 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS), Bangalore, 2014, pp. 1-8.doi: 10.1109/COMSNETS.2014.6734880
- [35] Martin Lévesque, Da Qian Xu, Géza Joós, and Martin Maier. 2012. Communications and power distribution network co-simulation for multidisciplinary smart grid experimentations. In Proceedings of the 45th Annual Simulation Symposium (ANSS '12). Society for Computer Simulation International, San Diego, CA, USA
- [36] F. (2017, April 20). FNCS/fncs. Retrieved February 19, 2018, from <https://github.com/FNCS/fncs>
- [37] Christopher Hannon, Jiaqi Yan, and Dong Jin. 2016. DSSnet: A Smart Grid Modeling Platform Combining Electrical Power Distribution System Simulation and Software Defined Networking Emulation. In Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS '16). ACM, New York, NY, USA, 131-142. DOI: <https://doi.org/10.1145/2901378.2901383>
- [38] Mosaik is a flexible Smart Grid co-simulation framework. Retrieved February 19, 2018, from <https://mosaik.offis.de/>
- [39] Schütte, S., Scherfke, S., & Sonnenschein, M. (2012). Mosaik - Smart Grid Simulation API - Toward a Semantic based Standard for Interchanging Smart Grid Simulations. SMARTGREENS.
- [40] Nessi2, Latest News. Retrieved February 19, 2018, from <http://www.nessi2.de/>
- [41] C. (2017, October 05). Coreemu/core. Retrieved February 19, 2018, from <https://github.com/coreemu/core>
- [42] GNS3 | The software that empowers network professionals. Retrieved February 19, 2018, from <https://www.gns3.com/>
- [43] H. Al-Mohannadi, Q. Mirza, A. Namanya, I. Awan, A. Cullen and J. Disso, "Cyber-Attack Modeling Analysis Techniques: An Overview," 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, 2016, pp. 69-76.doi: 10.1109/W-FiCloud.2016.29
- [44] Chun Yu (CY) Cheung, (2016). Threat Modeling Techniques
- [45] Jajodia, Sushil & Noel, Steven. (2010). Advanced Cyber Attack Modeling Analysis and Visualization. 113.

- [46] Shandilya, Vivek & B. Simmons, Chris & Shiva, S. (2014). Use of Attack Graphs in Security Systems. *Journal of Computer Networks and Communications*. 2014. . 10.1155/2014/818957.
- [47] Ou, X., & Singhal, A. (2012). *Quantitative security risk assessment of enterprise networks*. New York: Springer.
- [48] Michael Lyle,, Artz,. (2006). NetSPA : a Network Security Planning Architecture.
- [49] SANS Institute InfoSec Reading Room (2009), Investigative Tree Models
- [50] Duc, H. N. (2014, December 11). Attack Vector. Retrieved February 19, 2018, from <https://eforensicsmag.com/attack-vector/>
- [51] OH, S. Y. Penetration tester diary. Retrieved February 19, 2018, from <http://nerv0.blogspot.gr/2014/03/top-information-security-attack-vectors.html>
- [52] Category:Threat Modeling. Retrieved February 19, 2018, from https://www.owasp.org/index.php/Category:Threat_Modeling
- [53] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz, "Diamond Model of Intrusion Analysis," Center for Cyber Threat Intelligence and Threat Research, Hanover, MD, Technical Report ADA586960, 05 July 2013
- [54] Myers, M. K. (2017, November 07). What is the cyber kill chain? Why it's not always the right approach to cyber attacks. Retrieved February 19, 2018, from <https://www.csoonline.com/article/2134037/cyber-attacks-espionage/strategic-planning-erm-the-practicality-of-the-cyber-kill-chain-approach-to-security.html>
- [55] M Hutchins, Eric & J Cloppert, Michael & M Amin, Rohan. (2011). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Leading Issues in Information Warfare & Security Research*.
- [56] M. E. Kuhl, M. Sudit, J. Kistner and K. Costantini, "Cyber attack modeling and simulation for network security analysis," 2007 Winter Simulation Conference, Washington, DC, 2007, pp. 1180-1188.doi: 10.1109/WSC.2007.4419720
- [57] Yampolskiy, Mark & Horvath, Peter & Koutsoukos, Xenofon & Xue, Yuan & Sztipanovits, Janos. (2013). Taxonomy for description of cross-domain attacks on CPS. 135-142. 10.1145/2461446.2461465.
- [58] E. Ciancamerla, M. Minichino and S. Palmieri, "Modeling cyber attacks on a critical infrastructure scenario," IISA 2013, Piraeus, 2013, pp. 1-6.doi: 10.1109/IISA.2013.6623699
- [59] Y. Mo et al., "Cyber-Physical Security of a Smart Grid Infrastructure," in *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195-209, Jan. 2012.doi: 10.1109/JPROC.2011.2161428
- [60] Foreman, James & Gurugubelli, Dheeraj. (2016). *Cyber Attack Surface Analysis of Advanced Metering Infrastructure*.
- [61] A. Hahn and M. Govindarasu, "Cyber Attack Exposure Evaluation Framework for the Smart Grid," in *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 835-843, Dec. 2011.doi: 10.1109/TSG.2011.2163829
- [62] A. G. Wermann, M. C. Bortolozzo, E. Germano da Silva, A. Schaeffer-Filho, L. P. Gasparly and M. Barcellos, "ASTORIA: A framework for attack simulation and evaluation in smart grids,"

NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, 2016, pp. 273-280.doi: 10.1109/NOMS.2016.7502822

[63] Satin Asri and Bernardi Pranggono. 2015. Impact of Distributed Denial-of-Service Attack on Advanced Metering Infrastructure. *Wirel. Pers. Commun.* 83, 3 (August 2015), 2211-2223. DOI=<http://dx.doi.org/10.1007/s11277-015-2510-3>

[64] S. Tan, W. Z. Song, Qifen Dong and L. Tong, "SCORE: Smart-Grid common open research emulator," 2012 IEEE Third International Conference on Smart Grid Communications (Smart-GridComm), Tainan, 2012, pp. 282-287.doi: 10.1109/SmartGridComm.2012.6485997

[65] Olayokun, Olaoluwa (2016). Executing cyber security attacks on a smart grid testbed

[66] S, P. (2017). Impact of Denial of Service (DoS) attack in Smart Distribution Grid Communication Network.

Appendix