# Система за симулиране на кибератаки и събиране на данни за мрежовия трафик, натоварването на процесора и паметта

## System for simulating cyberattacks and acquisition data for network traffic and usage of processor and memory of hosts

## Система имитации кибератак и сбора данных о сетевом трафике и использовании процессора и памяти хостов

Александър Христов, Румен Трифонов
Technical University Sofia, Bulgaria

***Abstract:*** *Целта на настоящата работа е създаването на прототип на система за симулиране на кибератаки с възможности за събиране и съхраняване на данни от крайните устройства(хостове) за мрежовия трафик, натоварването на процесор и памет в централно хранилище за последващо обработване и класифициране на същите като компрометирани или некомпрометирани, използване на централизиран модел за управление, комуникация и съхранение на данни в системата, с удобен и лесен за употреба уеб интерфейс, който служи за основен панел за управление и наблюдение на мрежата и хостовете, и визуализиране и обобщаване на цялата събрана информация при наличие или отсъствие на кибератаки*
***Keywords****: DENIAL-OF-SERVICE ATTACK, VMware ESXi HYPERVISOR, CACTI, HPING3*

## *1. Въведение*

В днешните мрежи откриването и реакцията на атаки и пробиви е задача от първостепенна важност. Откриването на пробиви се базира на методи, средства и технологии за откриване, реагиране на компютърни атаки. Тези системи включват функции като предпазване, възпиране, реагиране, оценка, очакване на атаки, поддържане на информация за пробиви. Системите за откриване на проникване (IDS) наблюдават мрежовия трафик за подозрителна активност и извеждат предупреждения, когато такава дейност бъде открита [1,2]. Те използват софтуерно приложение, което сканира дадена мрежа или система за зловредни дейности (нарушения на зададените правила). Обикновено, всяко едно нарушение се докладва или на администратор, или се съхранява на централно място, използвайки система за информация за защита и управление на събития (SIEM). Системата SIEM интегрира[10] изходи от множество източници и използва техники за филтриране на известия, за да разграничи злонамерената дейност. Въпреки че системите за откриване на проникване наблюдават мрежите за потенциално злонамерена дейност, те също са склонни да извеждат фалшиви съобщения за проблем при липса на такъв. Следователно организациите, които ги използват, трябва да настроят своите IDS приложения при пускането им в продуктивна среда. Това означава системите за откриване на проникване така да се настроят, че да разпознават [4,5] как изглежда нормалният трафик в мрежата в сравнение със злонамерения такъв. В [3] е предложен подход за идентифициране на компрометирани устройства в резултат на кибератака, който се базира на мониторинг на използването на процесорите, паметта и мрежата.

Целта на настоящата работа е създаването на прототип на система за симулиране на кибератаки и събиране на данни за мрежовия трафик, натоварването на процесора и паметта със следната функционалност:

• събиране и съхраняване на данни от крайните устройства(хостове) за мрежовия трафик, натоварването на процесор и памет в централно хранилище за последващо обработване и класифициране на същите като компрометирани или некомпрометирани вследствие на кибератаки;

• симулиране на кибератаки и визуализиране и обобщаване на цялата събрана информация при наличие или отсъствие на кибератаки;

• използване на централизиран модел за управление, комуникация и съхранение на данни в системата;

• удобен и лесен за употреба уеб интерфейс, който ще служи за основен панел за управление и наблюдение на мрежата и хостовете.

Разработената система да позволява събиране на данни във времето за натоварването на мрежовия интерфейс, паметта и процесора на крайните устройства (хостове), както при отсъствие, така и симулиране на (състояща се) кибератака.

В първата част на работата е направен анализ на някои конкретни видове атаки. Във втора част са разгледани архитектурата на системата и основните й компоненти и инструменти, които са използвани за реализирането й. В трета част са дадени симулирането на някои конкретни атаки и първите получени резултати от работата на системата.

## *2. Анализ на някои конкретни атаки*

Атака за отказ на услуга[6] (Denial-of-service attack, DoS attack) е опит даден ресурс, предоставян от компютър (наричан жертва), да бъде направен недостъпен за потребителите му. Атаката може да бъде чрез изтощаване на ресурси или чрез експлоатиране на грешка в софтуера на жертвата. Най-често биват атакувани популярни уеб сървъри, като целта е те да станат недостъпни от интернет. Атаката се състои в претоварване на дадена машина (изчерпване на системния й ресурс) чрез заливане със заявки като целта е да се предотврати изпълнението на легитимните нови или на всички потребителски заявки. Когато атаката е осъществена от повече от един източници на трафик, тя се нарича дистрибутирана атака за отказ на услуга (англ. Distributed denial-of-service attack, DDoS attack “). DDoS атаките често се извършват срещу уеб сайтове от ботове, заразени машини с malware, действащи по инструкции от команден сървър от престъпни групировки, с цели на кражба, изнудване, и т.н. Пример за подобни действия са атаките срещу сайтовете на VISA, Mastercard и Paypal от страна на Anonymous през 2013г.

SYN flood е тип protocol-based DoS атака. За нейната реализация се използват пропуски в имплементацията на TCP three-way handshake. SYN flood атаката [11] създава наполовина отворени TCP канали, които никога не се отварят напълно. Злонамереното лице играе ролята на клиент, който инициира комуникацията по договаряне със сървъра. Проблемът възниква, когато сървърът изпрати SYN/ACK пакетът и остава в изчакване за обратен отговор с ACK пакет от страна на клиента. Хакерската машина никога не изпълнява завършващия етап от TCP three-way handshake. Инициирането на множество опити за създаване на TCP канал, без да бъде напълно завършен, постепенно изразходва системните ресурси на сървъра-мишена (за всяка наполовина отворена TCP връзка се заделя буфер). Тези действия могат да доведат до възпрепятстване на потребителите, които искат да се свържат легитимно. SYN flood атаката може да бъде разпозната, ако се следи мрежовият трафик на сървъра. Основен фактор, който определя потенциална атака, е броят на стартираните процеси

по договаряне в секунда. Основен параметър, който трябва да се наблюдава, е броят инициирани TCP пакети с SYN флаг в тях, които клиентите изпращат към сървъра, за секунда.

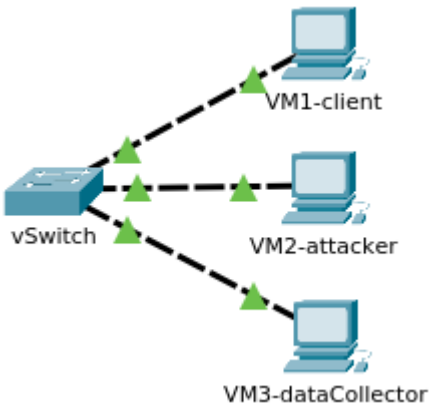
ICMP flood е тип volume-based DoS атака, която се базира на принципа на работата на ICMP протокола [11]. Съобщенията, които се използват за проверка на свързаността между две устройства са ICMP Echo Request и ICMP Echo Reply. ICMP flood атаката [2,3] използва тази комуникация, за да създаде недоброкачествен трафик. В рамките на малък период от време злонамереното лице генерира огромно количество Echo Request пакети, които пренасищат атакувания сървър. Мишената е длъжна да генерира огромно количество ICMP пакети от тип Echo Reply и да ги изпрати към отсрещната машина. В резултат целият този трафик задръства канала за комуникация и отнема голяма част от ресурсите на сървъра. В повечето случаи това води до прекъсване на достъпа до предоставената услуга. ICMP flood атаката може да бъде открита, ако се следят броят на ICMP пакетите от тип Echo Request за секунда, отправени към сървърната машина. В случай, че количеството значително превиши нормалното за сървъра, трафикът се счита за злонамерен.

UDP flood е тип volume-based DoS атака, която се осъществява в резултат на начина на функциониране и имплементация на UDP протокола. За разлика от TCP протокола, при който се създава TCP канал за комуникация, UDP функционира без да се създава сесия между клиента и сървъра. В нормални условия, когато клиентът иска да достъпи конкретна услуга на сървъра, той изпраща UDP пакет, с който проверява дали на даден порт се предлага търсената услуга. В случай, че на конкретния порт не работи никое приложение, сървърът генерира и връща към клиентската машина ICMP пакет от типа Destination Unreachable. UDP flood атаката [2,4] представлява генериране на огромно количество UDP пакети, които са насочени към различни портове на жертвата. В резултат сървърът е длъжен да изпрати обратен отговор на всеки пристигнал UDP пакет. Огромното количество трафик, което трябва да приема и предава сървърът обикновено води до заемане на всички системни ресурси и невъзможност за отговор на легитимните клиенти, които искат да използват дадена услуга. UDP flood атаката може да бъде разпозната като се следи за възникнали аномалии в трафика на UDP пакети. Можем да считаме, че е засечена потенциална DoS атака, ако броят на UDP пакетите изпратени към сървъра е значително по-голям от обикновено.

## *3. Архитектура на предлаганата система.*

Изградената система се състои от 3 виртуални машини: клиентска машина, атакуваща машина и машина за събиране на данни. Трите виртуални машини са свързани към един виртуален комутатор. Топологията на предлаганата система е показана на фиг. 8. За създаването на трите виртуални машини и виртуалния комутатор чрез който те са свързани в локална мрежа се използва VMware ESXi hypervisor [7, 8, 9]. На базата на анализ на различни инструменти[4, 5, 10] за наблюдение на мрежата е избран инструментът Cacti. Cacti е безплатен широко използван инструмент, с удобен графичен GUI интерфейс с богати възможности и позволява събиране на данни за мрежовия трафик, натоварването на процесора и паметта на клиентските машини в мрежата. Cacti използва стандартния протокол SNMP- Simple Network Management Protocol. При типично използване на Cacti, устройствата в мрежата следва да са с активиран SNMP [8, 9] и да има отделен сървър за наблюдение, където Cacti събира системните данни от тези устройства. Трите виртуални машини са с операционна система lubuntu 16.04 и параметри: 1 процесор; 1ГБ RAM памет; 10ГБ диск; виртуален мрежови интерфейс 1Gbps.

Настоящата работа е разработена при (в рамките на) някои ограничения поставени от HyperVisor-а. Ограничени са изчислителната мощност на използваните виртуални машини, както и събраните системни данни (от системния монитор). В процес на доразвиване и усъвършенстване на работата в реална локална мрежа ограниченията следва да бъдат премахнати.




***Фиг. 1*** *Топология на предлаганата система.*

Първоначално е създадена клиентската виртуална машина, като е инсталирана операционната й система. Клиентската виртуална машина е копирана 2 пъти- първо за атакуваща машина и след това за машина за събиране на данни. В системата се използва инструментът Cacti за събиране на данни за мрежовия трафик, натоварването на процесора и паметта на клиентските машини. Cacti използва протоколът SNMP, за да обменя с клиентската машина „мишена“ тези данни. Ето защо на машината „жертва“ е необходимо да е активиран SNMP [8]. В [6] е даден конфигурационният файл, в който основно са добавени редове, свързани с поведението на агента, работещ върху SNMP протокола.

Както бе споменато по- горе, атакуваща машина е копие на клиентската виртуална машина, като за копирането е използван инструментът VMware vCenter Converter Standalone [7, 8 ,9]. Тук за симулиране на различни видове атаки, се използва терминалният инструмент hping3 [11], тъй като е широко разпространен и лесен за употреба. Ето защо допълнително върху атакуваща машина е нсталиран hping3 пакета, което става чрез командата:

$ sudo apt-get update && apt-get install hping3

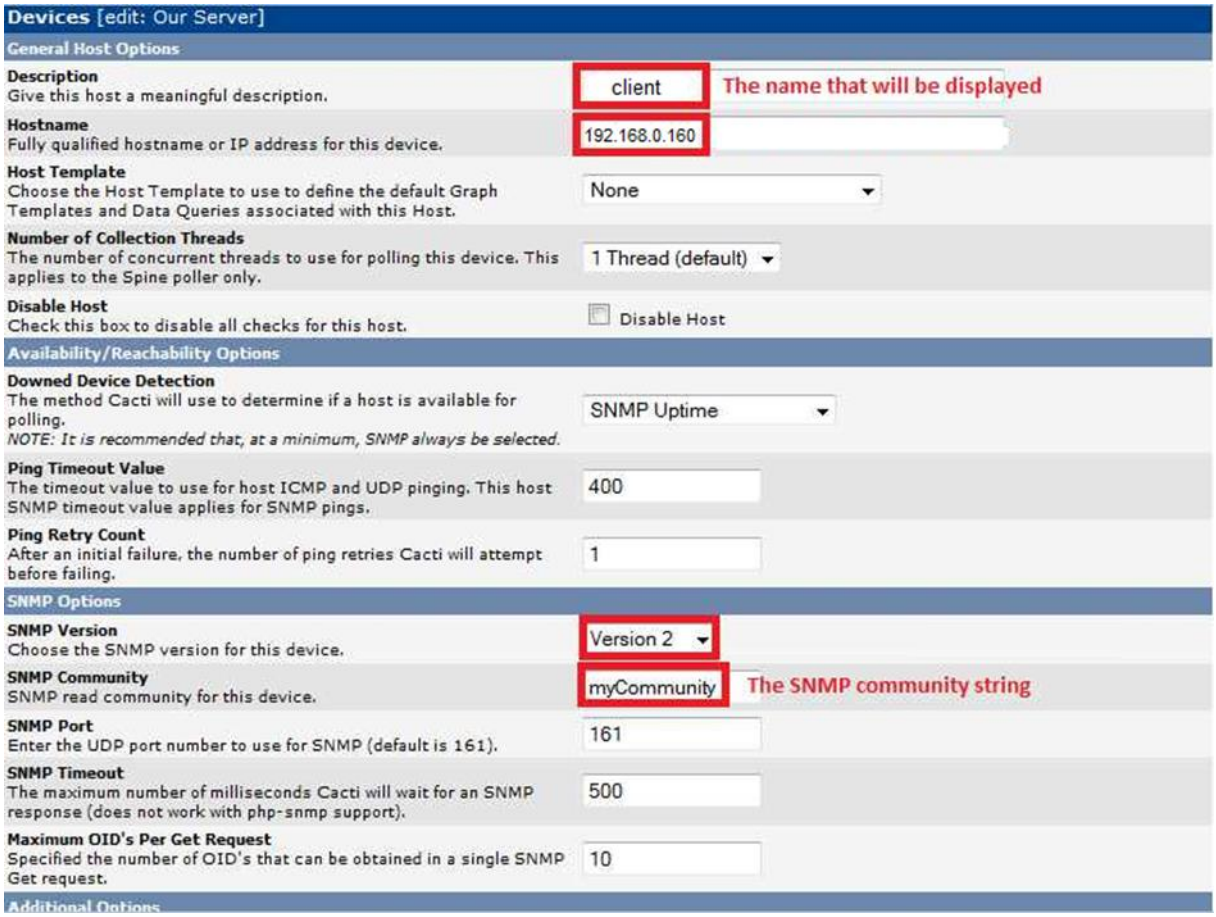
На машината за събиране на данни се инсталира cacti и необходимите за работата му пакети (MySQL, PHP, RRDTool, net-snmp и уебсървър, който поддържа PHP, например Apache) става чрез командата:

$ sudo apt-get update && apt-get install cacti

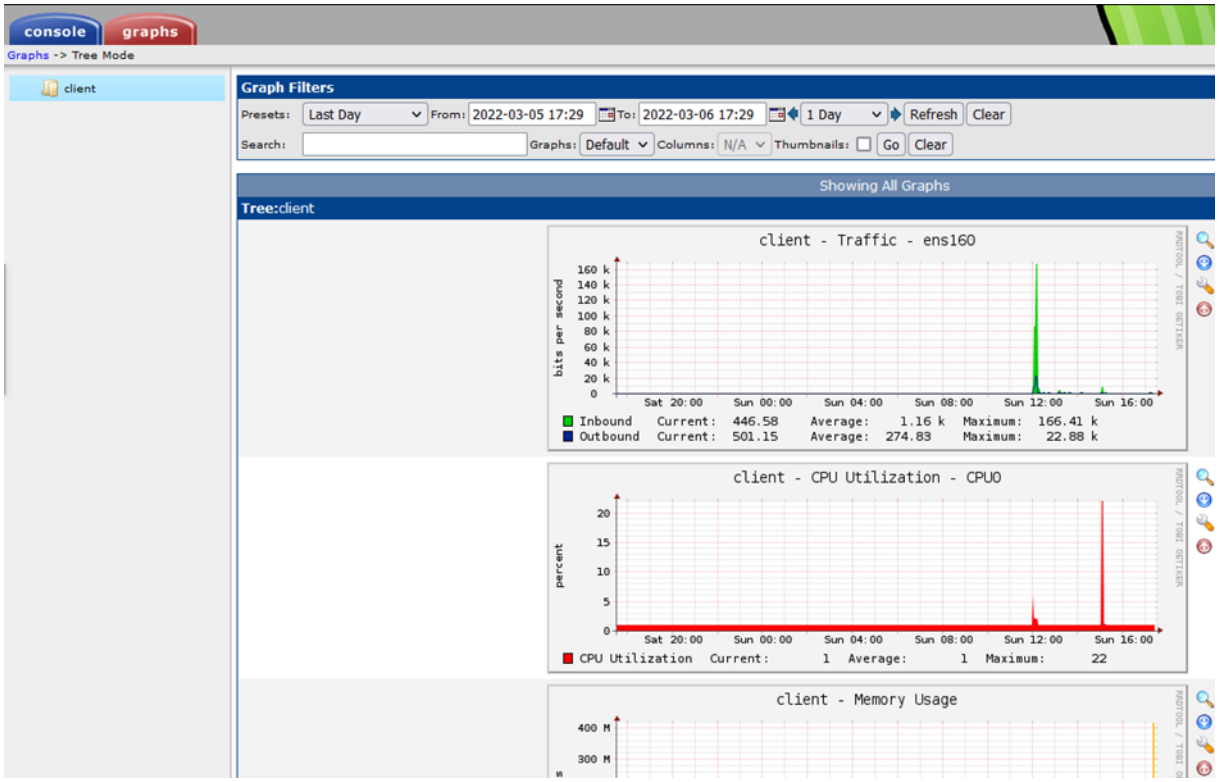
В появилите се прозорци за конфигурация на MySQL, Apache уебсървъра, базата данни на Cacti се задават съответните им параметри и пароли според [8].

След инсталирането, уеб интерфейсът на Cacti се достъпва чрез IP адреса на машината за събиране на данни (http://192.168.0.160/cacti). При първото отваряне на уеб интерфейса се преминава последователно през екран за приемане на лицензa, екран на който се избира “Нова инсталация” като тип на инсталацията и екран за настройка на пътищата до директориите на необходимите за работата на Cacti пакети (MySQL, PHP, RRDTool, net-snmp и Apache).

Клиентската машина се добавя към устройствата, за които Cacti събира данни и се задават параметрите й както е показано на фиг. 2




***фиг. 2*** *Добавяне на клиентска машина в Cacti*




***фиг. 3*** *Работен екран от Cacti с резултати*

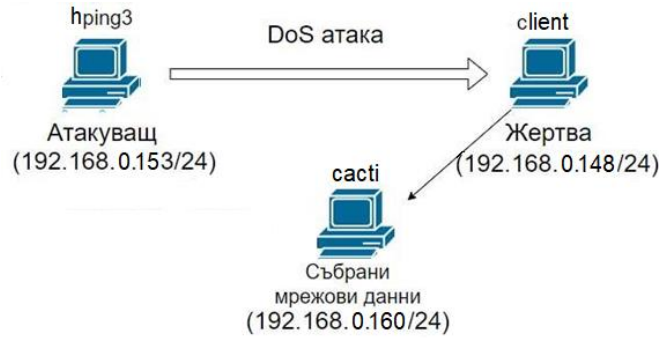
Създава се шаблон за графиките, които ще бъдат генерирани от извлечените данни. След това с натискането на бутона “Create graphs for this host” се създава графиката и се избира мрежовата карта, на която ще се следи трафика. Следва създаване на дърво от графики и задаване на параметрите на това дърво.

Накрая, към дървото с графики се добавят графики за мрежовата карта, на която ще се следи трафика, натоварването на процесора и RAM паметта. Поради ограничения обем на работата всички стъпки и работни екрани, както и файлове със събрани данни са дадени в [6].

Графиките от събраните по SNMP данни могат да бъдат видяни чрез натискане на бутона “Graphs” в горния ляв ъгъл на уеб интерфейса (фиг. 3). Също така от тази страница данните могат да бъдат свалени в CSV формат за по-нататъшна обработка. Е

## *4. Симулиране на атаки и получени резултати от работата на системата*

За целите на настоящата работа е генериран зловреден мрежов трафик като са извършени симулации на три вида DoS атаки в контролирана среда. Трите типа DoS атаки са SYN Flood, ICMP Flood и UDP Flood. Симулираните атаки са извършени само и единствено в рамките на настоящата работа с цел извличане на необходимата системна информация от тях. използвани са двете машини: едната, от които играе ролята на атакуващ, а другата – ролята на мишена. Атакуващата машина използва hping3 инструмент за симулиране на атаката. По време на атаката върху машината жертва е стартирана програмата за събиране на мрежови данни. На фигура 4 е изобразена схема на извършената симулация. Атакуващата машина има IP адрес 192.168.0.153 и 24-битова маска, а машината жертва – 192.168.0.148 и 24-битова маска. Събирането на мрежови данни (фиг. 11) се осъществява от третата машина, на която е инсталиран и конфигуриран CACTI пакета. Нейният IP адрес е 192.168.0.160 и 24-битова маска.




***фиг. 4.*** *Топология за симулиране на атаки и събиране на мрежови данни*

За стартиране на отделните атаки се използва терминалният прозорец, в който са въведени три различни команди:

-За генериране на SYN flood атаката е използвана следната команда:

hping3 -c 5000 -d 120 -p 80 —flood –rand-source 192.168.0.148

-За генериране на ICMP flood атаката е използваната команда е:

hping3 -c 5000 -d 120 –icmp –flood –rand-source 192.168.0.148

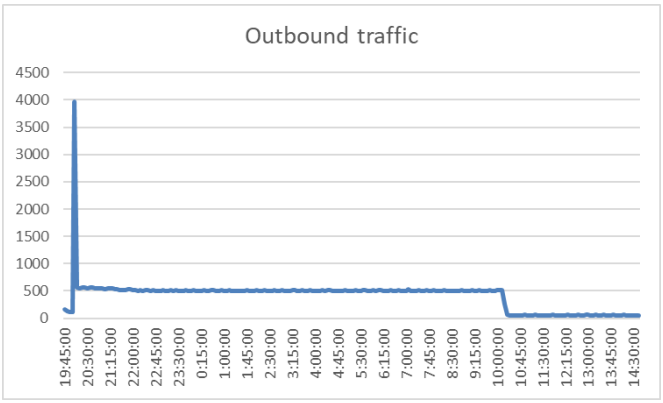
- За генериране на UDP flood атаката е въведена командата:

hping3 -c 5000 d 120 –udp –flood –rand-source 192.168.0.148

където --rand-source задава случаен IP адрес за изпращача на всеки пакет.

Командите се изпълняват многократно с различни настройки като се променят параметрите „брой генерирани пакети“ и „големина на генерирани пакети“, за да се създаде разнообразно множество от събрани данни.

На фиг. 5 са показани първите получени данни за натоварването на мрежовия интерфейс на машината «мишена» в резултат на ICMP flood атака, стартирана в 20:05 часа. Видна е разликата в натоварването на мрежовия интерфейс на машината «мишена», при нормален и зловреден мрежови трафик, т. е. по време на реализиране на атаката (между 20:05 и 10:10) и преди и след това (преди 20:05 и след 10:10 часа). Поради ограничения обем останалите получени резултати от работата на системата са дадени в [6].




***фиг. 5*** *Нормален и мрежов трафик при реализиране на атака*

## *5. Заключение*

Разработена е система за събиране на данни със зададената по- горе функционалност. Едно от многото приложения на предложената система е събраните системни данни да бъдат използвани за последваща обработка чрез достатъчно точен и бърз алгоритъм [3], който да разпознава „нормалния“ трафик и да класифицира разнообразните видове DoS и DDoS атаки.

## *6. Литература*


1. Alomari E., Gupta B. B., Karuppayah S., et al. Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art. International Journal of Computer Applications July 2012, Vol. 49, No. 7(2012)
2. Aunraj N. S., Heigl M., Hable R., et al. Comparison of Supervised, Semi-supervised and Unsupervised Learning Methods in Network Intrusion Detection System (NIDS) Application. ResearchGate, (2017)
3. Hristov A., Trifonov R. A Model for Identification of Compromised Devices as a Result of Cyberattack on IoT Devices, Proceedings of the 2021 International Conference on Information Technologies (InfoTech-2021), IEEE Conference, Rec # 52438, 16-17 September 2021, St. St. Constantine and Elena, Bulgaria (2021)
4. Othman S. M., Alsohybe N. T., Ba-Alwi F.M., et al. Survey on Intrusion Detection System Types. ResearchGate (2018)
5. Samrin R., Vasumathi D. Review on Anomaly based Network Intrusion Detection System. International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (2017)
6. https://github.com/sashkinaaa/cacti
7. https://www.xmodulo.com/install-configure-cacti-linux.html
8. https://www.cacti.net/info/downloads
9. https://docs.vmware.com/en/vCenter-Converter-Standalone/6.2/rn/conv_sa_62_rel_notes.html
10. http://www.vce-download.net/study-guide/comptia-securityplus-2.4.1-intrusiondetection-systems.html
11. https://study-ccna.com/icmp-internet-control-message-protocol/