International Journal of Safety and Security Engineering
Vol. 11, No. 3, June, 2021, pp. 239-246
Journal homepage: http://iieta.org/journals/ijsse

Mitigating Sodinokibi Ransomware Attack on Cloud Network Using Software-Defined Networking (SDN)

Rusydi Umar[1], Imam Riadi[2], Ridho Surya Kusuma[1*]

[1] Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta 55164, Indonesia
[2] Department of Information System, Universitas Ahmad Dahlan, Yogyakarta 55164, Indonesia

Corresponding Author Email: ridho2007048010@webmail.uad.ac.id

https://doi.org/10.18280/ijsse.110304

Received: 27 March 2021
Accepted: 12 June 2021

Keywords:
open virtual switch, ransomware, Ryu controller, SDN, Sodinokibi

ABSTRACT

Sodinokibi Ransomware virus becomes a severe threat by targeting data encryption on a server, and this virus infection continues to spread to encrypt data on other computers. This study aims to mitigate by experiment with building a prevention system through computer network management. The mitigation process is carried out through static, dynamic, and Software-Defined Networking (SDN) analysis to prevent the impact of attacks through programmatic network management. SDN consists of two main components in its implementation, the Ryu controller and Open Virtual Switch (OVS). Result testing mitigation system on infected networks by crippling TCP internet protocol access can reduce virus spread by 17.13% and suppress Sodinokibi traffic logs by up to 73.97%. Based on the percentage data, SDN-based mitigation in this study is per the objectives to make it possible to mitigate Ransomware attacks on computer network traffic.

1. INTRODUCTION

A computer network is a valuable asset that is business continuity in this increasingly modern digital era. Cybercrime targets attacks on computer networks by encrypting data on companies for profit [1, 2]. In the first quarter of 2020, ransom payments for attackers with ransomware viruses experienced an average increase of 33% from the fourth quarter of 2019. One of the top three types of Ransomware viruses is Revil/ Sodinokibi [3]. Based on the Global Report by BlackFog Enterprise, throughout 2020, Sodinokibi virus species have become the most widely used cyber weapons [4], which is about 15.6% of 80% [5], in May 2020 in San Diego, a Sodinokibi virus attack hijacked Harvest Food Distributors data. It demanded US $7.5 million [6].

Sodinokibi Virus Attack is a combined attack with ransomware model as a Service (RaaS) and spread through affiliates when performing its actions. Here are four stages of viruses in the episode: The first step is phishing campaigns through email and website links, then manipulating users to execute malicious files and gain access rights on the user's computer. The second stage is accessing computer resources, deleting backup data, encrypting data with curve25519 or AES-256-CTR encryption [7-9]. The third stage displays a ransom message to the user's monitor screen. Finally, the fourth stage spreads advanced infections to his other computers. Based on this, the importance of mitigating because the excess of this virus can apply widespread attack expansion through the same network traffic and can operate without having to connect to the command and control server [9-11].

Therefore, this study experimented with mitigation using SDN to prevent the spread of Sodinokibi Ransomware attacks through computer network traffic. SDN can make network configuration changes in real-time and compatible with the evolution of ransomware attack techniques that allow changes in a certain period.

The detection system in this study used the Intrusion Detection System and mitigation of SDN, which played an essential role in managing network traffic using Python programs. The implementation of the mitigation uses a malware analysis stage which consists of static and dynamic analysis [12]. This stage aims to obtain information such as Indicator of Compromise (IOCs), Code Hash, file activity system, and network behaviour that become characteristics or signature viruses [13, 14]. Thus, the signature on the virus can help as initial capital in building a detection and mitigation system.

2. LITERATURE REVIEW

Here is an analysis of previous research on ransomware. Adamov and Carlson conducted mitigation research with spam filters to quarantine suspicious emails, use execution prevention modules, and track Ransomware traffic using the Tor network [12]. Mohammad, in his research, explained two independent classifiers in reducing Crypto-Ransomware attacks, namely: first, user training, security policy, and data backup. Secondly, the detection system uses artificial intelligence [11]. Chakkaravarthy et al. use the Intrusion Detection Honeypot (IDH) and Social Leopard Algorithm systems to detect Ransomware attacks through an early warning on the user's system against suspicious file activity [15]. Xia researched by implementing a Network Assisted Approach (NAA) method to detect ransomware at the local and network level. This system makes it possible to determine which machines the ransomware is on [16]. Agrawal et al., in

their research, by adopting deep learning methods to detect Ransomware from emulation sequences, this study used neural networks with extended short-term memory models to record local event patterns in Ransomware [17]. Ganfure et al., researching with the use of the concept of DeepGuard. This concept detects Ransomware based on user behaviour so that DeepGuard can distinguish ransomware activities from user activity with false-positive assessment output [18]. Ayub et al., Perform Ransomware detection on artificial neural networks (ANN) through monitoring of I/O Request Packet (IRP) logs by obtaining the lowest level data [19]. These logs can further describe Ransomware activity [20].

Akbanov et al. conduct mitigation with the WannaCry ransomware virus case using SDN technology. SDN allows being able to detect and reduce the spread of viral infections as the technology changes the flow of the network through OpenFlow, thus breaking the chain of Ransomware attacks [21]. Implementation of SDN in Ransomware mitigation, based on DNS traffic, Pox controller, and WannaCry virus [12, 21]. This mitigation research offers a mitigation approach to the Sodinokibi Ransomware virus using SDN technology, Ryu Controller, and Cloud Network Environment.

2.1 Sodinokibi ransomware

The Sodinokibi virus is known as the REvil virus, which has a type of RaaS attack operation. RaaS means the attack process through cooperation between the ransomware developer and the affiliate spreader. Sodinokibi adopted a high-level encryption technique with two public keys to encrypt the victim's data encryption key, one belonging to develop RaaS and the other belonging to an affiliate. Different from this virus with regular ransomware is using a particular variable, namely Identity Afilisasi embedded. These variables can be configured, run automatically, and send victim data to the C2 server [22, 23].

2.2 System models

This study uses Cloud Network Environment for virtualization to run mitigations and SDN-based network systems as mitigation.

2.2.1 Cloud network environment

Cloud Network Environment is a technology that can share hosts, access distributed environments, and virtualization [24]. The process of creating a virtual environment using VirtualBox and GNS3 software to perform valid and realistic ransomware experiments. In this virtual environment, already available viruses, data, files, infected computers and clean computers [25], making it possible to carry out mitigation to prevent the spread of viruses from cleaning computers. The environmental structure design in this study is as shown in Figure 1.

Figure 1 provides information that the structure of the virtual environment has a computer network with different communication segmentation using black and orange lines. The line's colour difference indicates that the black line is a communication network access point programmed using SDN technology and managed using the Ryu controller. The following is the network configuration data for each device, as shown in Table 1.

Table 1 shows that this environment consists of five devices: two servers with windows and a Ubuntu server; two client computers with Windows 7; and an OpenFlow switch. Each machine uses a configuration with the network address 192.168.10.0/24.

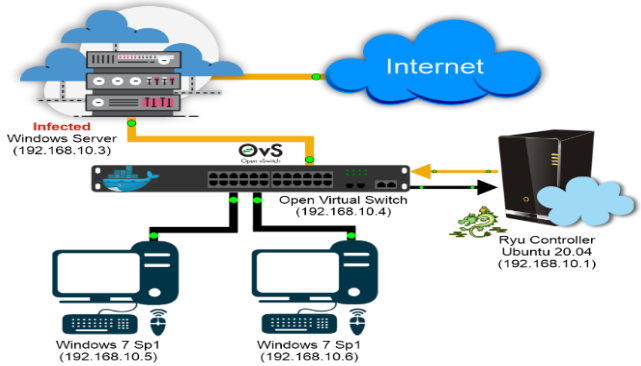


Figure 1. Environment structure

Table 1. Configuration network

Device	Os	IP Address	Port
Server1	Win. Server	192.168.10.3/24	Ether2
Server2	Ubuntu 20. 04	192.168.10.1/24	Enp0s3
Switch	OpenvSwitch	192.168.10.4/24	Eth1-15
Host1	Win. 7	192.168.10.5/24	NIC
Host2	Win. 7	192.168.10.6/24	NIC

2.2.2 SDN

SDN is a method or paradigm that separates the field of control and the field of forwarding. The control field serves to set up or control network devices, while the forwarding field sends information packets. The advantage of SDN is that it allows the configuration of programmable computer networks. SDN consists of Application, Controller, and OpenFlow switch [26, 27]. The following is an explanation of the image of the SDN structure, as shown in Figure 2.

Figure 2 shows that SDN has a role in managing network traffic based on commands on the application [28]. Generally, the functions of the three SDN layers are the Application layer as a network configuration input such as Rest API commands and python programs. The controller serves as a framework that runs configurations according to application inputs such as Ryu, Pox, Onos and others. The design of the cloud environment network structure, as shown in Figure 3.

The cloud environment network structure built will be checked for feasibility for the mitigation process by testing each device's functionality and connection; This test result serves as the initial setting. On the network structure, note the dotted red line indicating the flow between the host and host to server1 that must be connected, the dotted orange line means communication between the switch and the Ryu controller, the following is the initial contact dataset between devices before the program enters as shown in Table 2.

The entire device cannot connect except server2, which acts as a network controller and has particular communication, run the python program switch_l3.py to upload the communication stream in the dataset so that the network of computers between server1 to the host and host to the host can communicate with each other as in Table 3.

Table 3 shows all devices have successfully communicated after the dataset stream is added via python program from the controller to switch devices. Thus, setting the flow of data packets and network using python programming.

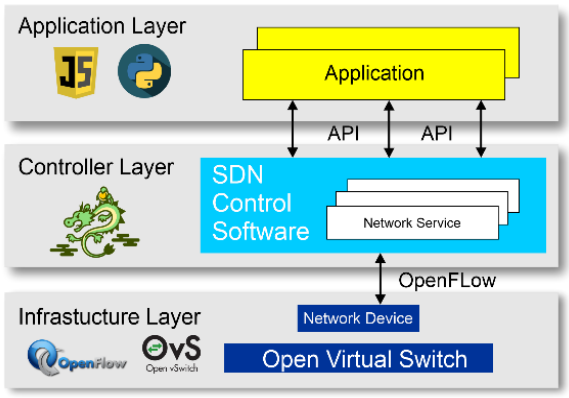


Figure 2. The SDN structure

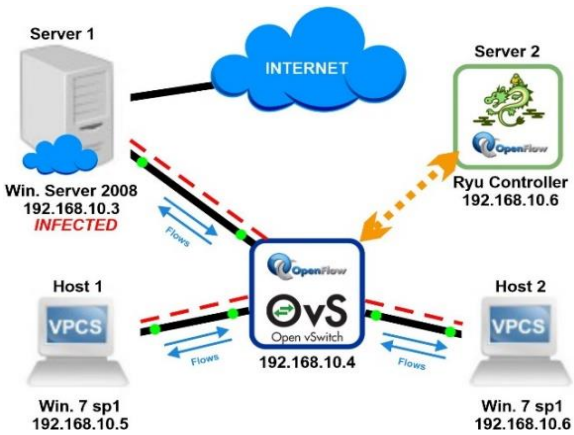


Figure 3. Network structure

Table 2. Connectivity of SDN inactive

Device	Src. IP	Etc. IP	Status
Server1	192.168.10.3	192,168.10.5	Packet Loss
Server2	192,168.10.1	192.168.10.4	Connected
Switch	192.168.10.4	192.168.10.3	Packet Loss
Host1	192.168.10.5	192.168.10.6	Packet Loss
Host2	192.168.10.6	192,168.10.3	Packet Loss

Table 3. Connectivity of SDN active

Device	Src. IP	Etc. IP	Status
Server1	192.168.10.3	192.168.10.5	Connected
Server2	192.168.10.1	192.168.10.4	Connected
Switch	192.168.10.4	192.168.10.3	Connected
Host1	192.168.10.5	192.168.10.6	Connected
Host2	192.168.10.6	192.168.10.3	Connected

3. THE PROPOSED APPROACH

This research aims to experiment with SDN with cases of ransomware virus attacks on computer networks.

3.1 Research subject

The subjects used in this study were SDN Ryu Controller-based networks. Ryu Controller is a part of SDN; Ryu's strengths control and manage the network with python programming, which allows it to mitigate virus attacks, especially viruses that can spread and threaten assets on a network.

3.2 Experiment research stages

This experiment research consists of three scopes to start the implementation of an SDN-based mitigation system [29], namely: The first scope prevention phase involves looking for potential threats, static analysis, dynamic analysis, updating the detection system, and strengthening security system to prevent any attack so as not to interfere [30] successfully.

The second scope is the scanning phase to vulnerability assessment, security reinforcement, and risk management. Finally, the third scope is the mitigation phase to reduce the spread of infections from ransomware attacks. The flowchart of the experimental stages in this study refers to Figure 4.

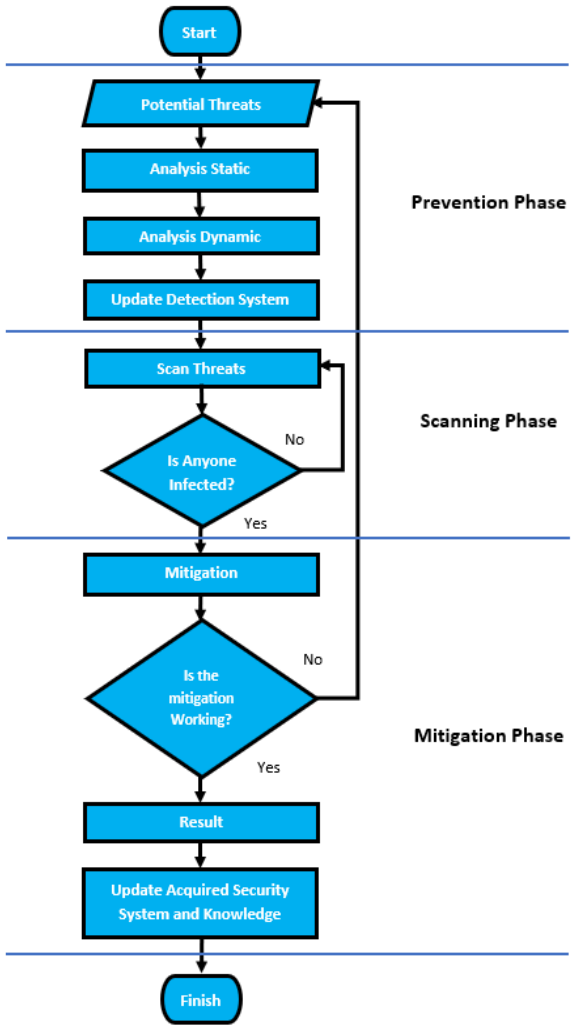


Figure 4. Flowchart of experiment research stages

Figure 4 provides information on how the experimental workflow in this research is for the implementation of mitigation. There are six stages to begin the process with; the first stage is finding potential threats and preparing tools forensics used in the analysis and mitigation process. The second stage of analysis static is the process of identifying suspicious file characteristics so that they can be alerted and detected with IDS based on data on the database signature. The third phase of the analysis dynamic is the process of executing malicious files in a cloud network environment to obtain data, behaviour patterns, and activity logs based on network behaviour [31]. The fourth stage builds IDS early detection on computer networks [32, 33]. The detection system set to find

threats on cloud network environment is to implement SDN and network calibration before the mitigation process. The fifth phase of Implementation Mitigation is the process of testing mitigation against Ransomware attacks on computer networks. Mitigation process by sending additional programs to block the network protocol most used by viruses so that the machine cannot send data packets through a specific network protocol. This mitigation does not affect connections on other devices, and infected machines remain connected through limited network services.

Based on the mitigation process, the infected machine allows for data repair and recovery. The sixth stage of the result obtained in mitigation is the percentage value of relief applied, updating system detection, ransomware, and network security knowledge [11].

4. RESULT AND DISCUSSION

Here is the analysis of the results of there is research mitigation of Sodinokibi virus attack based SDN that allows the prevention of spread or risk on a computer network.

4.1 Potential threats

Mitigation preparation by preparing a testing environment, providing samples of potential threat Sodinokibi Ransomware viruses that serve as objects (https://app.any.run/submissions), and implemented into a phishing website www. skincare shop.42web.io, influences its victims to become infected by downloading and executing malicious files [34]. Here are forensic tools in SDN-based mitigation testing against Sodinokibi Ransomware attacks, as shown in Table 4.

Table 4. Forensic tools

No.	Tools	Version	Function
1.	Wireshark	3.2.4.	Capture Network
2.	Ryu Controller	4.34	SDN *Controller*
3.	IDS	4.0.5	*Detection*
4.	Pestudio	9.09	*Analysis Static*
5.	Noriben	v1.8.4	*Analysis Dinamic*
6.	Proc. Monitor	V3.61	*Analysis Dynamic*

Table 4 describes the type of forensic tools as a support in the mitigation process, namely Wireshark for capture log traffic network, Ryu for network transmission controller, IDS for Ransomware detection, Pestudio for static analysis process, Noriben and Proc. Mon serves for dynamic analysis [35].

4.2 Static analysis

Static analysis results to find out the Indicator of Comprimes (IOCs) as characteristics Sodinokibi virus files and investigate it using Pestudio tools can be seen in Table 5.

Static analysis file in Table 5. provides signature information in the form of md5 hash code, sha1, sha256, first-byte-hex starting with code "4D 5A", first-byte-text beginning with the letter "M Z", file-type "dynamic-link-library", cpu 32bit, and virus signature using Microsoft Visual C++. Based on data files can be retrieved IOCs SHA-256 FBA8297590359DEA 91DB09ACBB4674237D8DBC57EC8B76A3EBE227DA9A E965353.

Table 5. IOCs Sodinokibi

Property	Value
MD5	E9075F6BB4802B4CB56EEC65F289...
SHA1	4C7DA5878B1A233B46F5E80F748...
SHA256	FBA8297590359DEA91DB09A...
First-hex	4D 5A 90 00 03 00 00 00 04 00 00 ...
First-text	M Z.. .. ..
Impash	CF2F44068A4D324522248D26669...
Signature	Microsoft Visual C++ 8
Entry-point	E8 B6 00 00 E9 89 FE FF FF 8B FF ...
File type	Executable
Cpu	32-bit
Subsystem	Gui
Timestamp	0x5B92DA38 (Sep 08 03:06:16 2018)

4.3 Dynamic analysis

In the Dynamic analysis process, ransomware must be run in a controlled environment to determine the characteristics of the virus based on Network Behavior and impact, as shown in Figure 5.

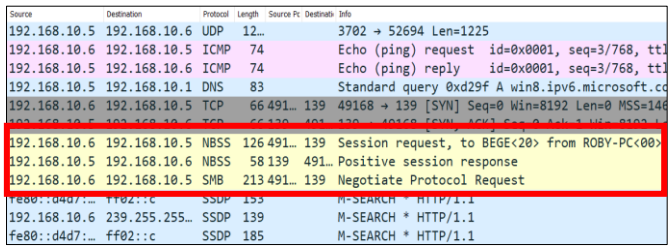


Figure 5. Network traffic log

The results of the network traffic log in Figure 5, showing the spread of Sodin, begins by creating a request session on the NBSS protocol to Roby-PC client computer from an infected computer that is BEGE-PC, then obtained a message "Positive session response", which means that it allows the virus can move from one place to another. The use of the target port is port 139 that is a computer with the host-name Roby-PC. The following details of the use of network protocols by viruses when spreading expansion to other computers as shown in Table 6.

Table 6. Network protocols

No	Protocol	Load (ms)	Percentage (%)
1	SMB	28	0.1
2	SMB2	1112	4.04
3	TCP	25087	91.1
4	UDP	16	0.06
5	ICMP	32	0.12
6	NBSS	12	0.04
7	NBNS	32	0.12
8	DNS	1220	4.43

Based on the information in Table 6 shows that there are eight network protocols used by the virus when running its attack, namely SMB, SMB2, TCP, UDP, ICMP, NBSS, NBNS, and DNS. In this research, the NBSS protocol has a role in finding the vulnerability and looking for ports that are free to use to open access to other computers, the SMB/SMB2 protocol has a role in opening sessions to connect and open paths between computers, and the TCP protocol has a role to perform data transfer illegally from an infected computer to

another. Here the protocol used by the virus during the following deployment process can be seen in Figure 6.

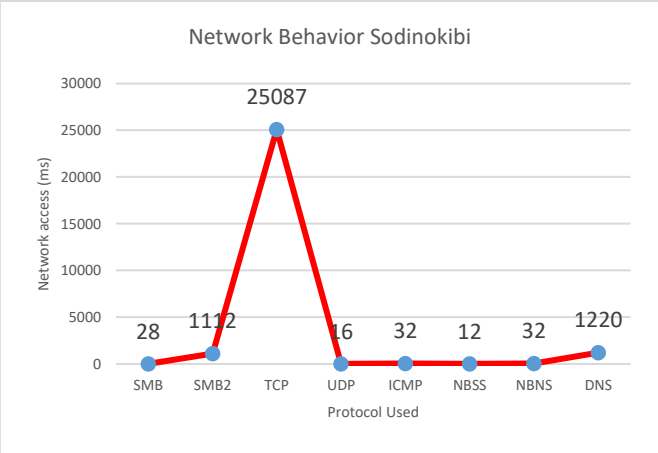


Figure 6. Network behaviour Sodin/REvil

Figure 6 shows a graph of the spread of Sodinokibi infection to client computers connected to the same network. Based on the network behaviour graph, most protocols used as infection media are NBSS, SMB, SMB2, and TCP. The following is the suspicious traffic found, as shown in Table 7.

Table 7. Suspicious traffic network

IP Source	Prot.	IP Dest.	Info
185.27.134.218	HTTP	192.168.10.6	Download file
192.168.10.6	DNS	35.204.114.36	Virus to C2 server
192.168.10.6	TLSv	35.204.114.36	Client Hello
35.204.114.36	TCP	192.168.10.6	Sending Data
35.204.114.36	TLSv	192.168.10.6	Handshake Failure
192.168.10.6	NBSS	192.168.10.5	Positive Session
192.168.10.6	SMB2	192.168.10.5	Spreading Virus
192.168.10.6	TCP	192.168.10.5	Sending Data

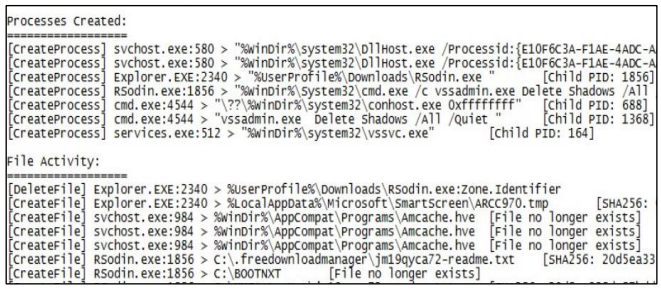
The following is a list of ports used by viruses in infection and spread on computer networks, as shown in Table 8.

Table 8. The ports used Sodinokibi

No.	Src. Address	Src. Port	Dst. Port	Protocol
1.	192.168.10.6	49168	139	TCP
2.	192.168.10.6	49171	445	TCP
3.	192.168.10.6	49168	139	SMB2
4.	192.168.10.6	49171	445	SMB2
5.	192.168.10.6	49168	139	NBSS
6.	192.168.10.6	49171	445	NBSS

Table 7 shows the initial infection and spread of viruses in computer networks, and Table 8 shows the list of ports. The process infection starts when the host computer downloads a malicious file on a phishing website with IP Address 185.27.134.218; after the file runs, the file immediately contacts the C2 server with IP Address 35.204.114.36 and spread to other computers with IP Address 192.168.10.5. The following explains the log processes created and virus activity files captured using Noriben tools in Figure 7.

The result log processes created describe the process that Sodin built with PID 1856, accessing %WinDir%\System32 \cmd.exe/c vssadmin.exe for escalation credentials to the administration level, and the File Activity log describes the steps it performs by dropping the payload on %LocalAppData %\ Microsoft \SmartScreen\ARCC970.temp then encrypting the file with sha256 code. Finally, based on dynamic analysis data, the SDN-based mitigation model is designed by preparing a Python script to block the TCP connection of an infected computer, as in Figure 8.



```
Processes Created:
==================
[CreateProcess] svchost.exe:580 > "%WinDir%\system32\DllHost.exe /Processid:{E10F6C3A-F1AE-4ADC-A
[CreateProcess] svchost.exe:580 > "%WinDir%\system32\DllHost.exe /Processid:{E10F6C3A-F1AE-4ADC-A
[CreateProcess] Explorer.EXE:2340 > "%UserProfile%\Downloads\RSodin.exe "       [Child PID: 1856]
[CreateProcess] RSodin.exe:1856 > "%WinDir%\System32\cmd.exe /c vssadmin.exe Delete Shadows /All
[CreateProcess] cmd.exe:4544 > "\??\%WinDir%\system32\conhost.exe 0xffffffff"   [Child PID: 688]
[CreateProcess] cmd.exe:4544 > "vssadmin.exe  Delete Shadows /All /Quiet "      [Child PID: 1368]
[CreateProcess] services.exe:512 > "%WinDir%\system32\vssvc.exe"        [Child PID: 164]

File Activity:
==================
[DeleteFile] Explorer.EXE:2340 > %UserProfile%\Downloads\RSodin.exe:Zone.Identifier
[CreateFile] Explorer.EXE:2340 > %LocalAppData%\Microsoft\SmartScreen\ARCC970.tmp       [SHA256:
[CreateFile] svchost.exe:984 > %WinDir%\AppCompat\Programs\Amcache.hve  [File no longer exists]
[CreateFile] svchost.exe:984 > %WinDir%\AppCompat\Programs\Amcache.hve  [File no longer exists]
[CreateFile] svchost.exe:984 > %WinDir%\AppCompat\Programs\Amcache.hve  [File no longer exists]
[CreateFile] RSodin.exe:1856 > C:\.freedownloadmanager\jm19qyca72-readme.txt    [SHA256: 20d5ea33
[CreateFile] RSodin.exe:1856 > C:\BOOTNXT       [File no longer exists]
```

Figure 7. Logfile

```
drop flow #TCP
match=parser.OFPMatch(eth_type=ether_types.ETH_TYPE_I
P,ip_proto=in_proto. IPPROTO_TCP)
mod=parser.  OFPFlowMod(datapath=datapath,  table_id=0,
priority=10000,  match=match)
datapath.send_msg(mod)
```

Figure 8. Python script

In Figure 8, the line "drop flow #TCP" is an essential part of the network control program by SDN Ryu. Here is an explanation of the existing script lines. Python script " match = parser. OFPMatch(eth_type)" serves to verify the ethernet used similarly and focus monitoring on TCP protocol, "mod = parser. OFPFlowMod(data)" helps to add a new entry flow table to OVS, with a priority of 10,000 which means a special command to override the network traffic flow.

4.4 Detection system

The detection system in this study is to check the IP address and port in the process of transmitting data packets based on the IP Address and port access-list storage database used by Sodinokibi Ransomware. The blocklist of IP addresses and ports is as in Table 9, and the implementation of the detection system is as in Figure 9.

Table 9. Blocklist IP address and port

IP s.	IP d.	Port1	Port2	Port3
185.27.134.218	192.168.10.6	139	445	443
35.204.114.36	192.168.10.6	139	445	443
156.241.187.210	192.168.10.6	139	445	443
68.71.130.58	192.168.10.6	139	445	443
134.209.222.65	192.168.10.6	139	445	443
35.202.21.90	192.168.10.6	139	445	443
92.43.216.137	192.168.10.6	139	445	443
77.111.240.141	192.168.10.6	139	445	443
172.67.162.142	192.168.10.6	139	445	443
54.39.163.98	192.168.10.6	139	445	443
104.26.13.36	192.168.10.6	139	445	443
134.119.232.33	192.168.10.6	139	445	443

Figure 9 provides illustrative information on how the system detects ransomware network traffic by identifying the IP Address and port that OpenFlow passes through and based on the data in Table 9. If the system detects a virus infection on a particular device, then the system will add a new flow entry to the table "06" on the switch, thereby enabling the blocking of malicious traffic on TCP protocol originating from the infected device.

In general, the flow table consists of three tables, namely the "03" table serves to verify the network similarity, the "06" table helps filter incoming transmission packets and determines which packets can be forwarded or dropped, and the "09" table serves to determine the appropriate output port will send data packets to the destination computer.

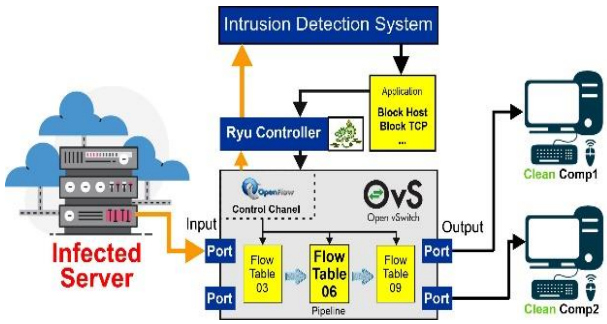


Figure 9. Detection system concept

4.5 Mitigation system

Based on the results obtained at the stage of static and dynamic analysis of the Sodin sample. The mitigation process in this research is to stop the spread of the Sodin/REvil virus through the network by the Ryu Controller software. Ryu Controller controls data flow access on the network, especially the TCP protocol, in real-time. If Ryu Controller gets a command from the detection system to close network access, the mitigation program will start running. This mitigation system starts by running a python script, as shown in Figure 8. The command for Ryu manager "Ryu-manager BlockTCP.py". The results of testing the mitigation system based on Network Behavior are as shown in Table 10, and Figure 10.

Table 10. Network protocols Sodinokibi

No	Protocol	Load (ms)	Percentage (%)
1	SMB	36	0.13
2	SMB2	1531	5.47
3	TCP	20691	73.97
4	UDP	12	0.04
5	ICMP	46	0.16
6	NBSS	5639	20.16
7	NBNS	16	0.06
8	DNS	0	0

Table 10 shows detailed information on the use of network protocols by the virus when the detection system has successfully identified the threat.

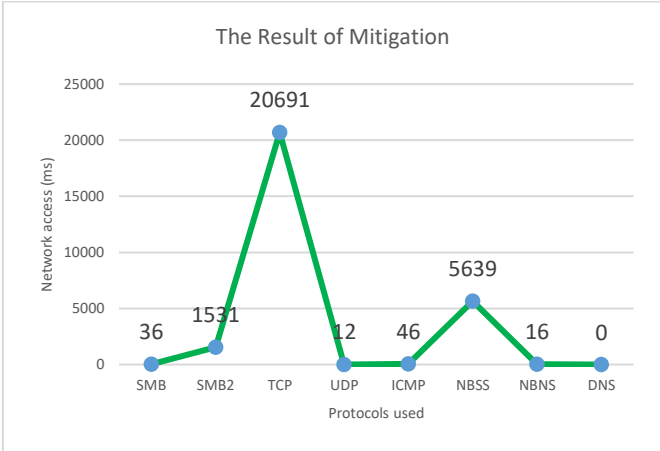
Figure 10 shows point "a" log flow switch OpenFlow describes the priority TCP packet dominates in because it is the highest priority with a value of 10.000, so the drop set instruction takes precedence in the entry flow data and applied to the transmission packet. Point "b" showed the log traffic network when the mitigation system dropped the suspicious packet transmission. Point "c" indicates the mitigation process successfully suppressed the speed of virus data flow on TCP protocol so that the packet transmission does not run, the spread and encryption process on another computer has stopped successfully. However, there is an increase of 5639 times in NBSS access protocol as the virus periodically establishes request sessions.

```
1 xclown@xcyber:~/ryu$ sudo ovs-ofctl -O OpenFlow13 dump-flows s1
2  cookie=0x0, duration=14.662s, table=0, n_packets=0, n_bytes=0, priority=10000,tcp
  actions=drop
3  cookie=0x0, duration=52.818s, table=0, n_packets=15, n_bytes=1358, priority=1,in_port="s1-
  eth1",dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
4  cookie=0x0, duration=52.814s, table=0, n_packets=14, n_bytes=1260, priority=1,in_port="s1-
  eth2",dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
5  cookie=0x0, duration=14.663s, table=0, n_packets=7, n_bytes=462, priority=0
  actions=CONTROLLER:65535
```

(a) Log flow on the switch

Source	Destination	Protocol	Length	Source Port	Destination Port	Info
192.168.10.5	192.168.10.3	SMB	97	139	49164	Logoff AndX Response
192.168.10.5	192.168.10.3	SMB2	126	139	49163	Session Logoff Response
192.168.10.3	192.168.10.5	SMB2	126	49163	139	Tree Disconnect Request
192.168.10.3	192.168.10.5	SMB	93	49164	139	Tree Disconnect Request
192.168.10.5	192.168.10.3	SMB	93	139	49164	Tree Disconnect Response
192.168.10.5	192.168.10.3	SMB2	126	139	49163	Tree Disconnect Response
192.168.10.3	192.168.10.5	SMB	97	49164	139	Logoff AndX Request
192.168.10.5	192.168.10.3	SMB	97	139	49164	Logoff AndX Response
192.168.10.3	192.168.10.5	TCP	54	49164	139	49164 → 139 [FIN, ACK] Seq=
192.168.10.5	192.168.10.3	TCP	54	139	49164	139 → 49164 [FIN, ACK] Seq=
192.168.10.3	192.168.10.5	TCP	54	49164	139	49164 → 139 [ACK] Seq=2691
192.168.10.3	192.168.10.5	TCP	54	49163	139	49163 → 139 [ACK] Seq=13026
PcsCompu_6b:e1…	Broadcast	ARP	42			Who has 192.168.10.4? Tell
192.168.10.3	192.168.10.5	SMB2	126	49163	139	Tree Disconnect Request
192.168.10.3	192.168.10.5	SMB2	126	49163	139	Tree Disconnect Request
192.168.10.5	192.168.10.3	TCP	54	139	49163	139 → 49163 [ACK] Seq=12977
192.168.10.5	192.168.10.3	SMB2	126	139	49163	Tree Disconnect Response
192.168.10.5	192.168.10.3	SMB2	126	139	49163	Tree Disconnect Response

(b) Log traffic network



(c) Graph of protocol traffic on the network

Figure 10. Mitigation results

4.6 Result

The results section of this study provides information on the comparison of packet protocols on computer networks between infection and mitigation processes by blocking TCP packet transmission. The following is a comparison of experimental data based on network protocols, as shown in Table 11.

Based on data in Table 11. It shows that mitigation by blocking the TCP protocol can reduce the infection of Sodinokibi Ransomware 17.13% from a value of 91.1% to 73.97%. The following is a graph of the data from behavioural network-based mitigation experiment results, as shown in Figure 11.

Based on Figure 11. The mitigation data obtained consists of two parts; namely, the first part is the virus infection process,

and the second part is the mitigation process by adding flow entries that block TCP protocol access using the python program on the infected device. The mitigation process involves searching for identification of network traffic behaviour throughout the LAN environment and showing suspicious anomalies [36]. Based on Figure 11, mitigation data obtained consists of two parts; that is, the first part of the graph in red is the virus infection process, and the second part of the graph in green is the mitigation process by adding a new stream-entry that blocks TCP protocol access using the python program on the infected device. The mitigation process involves searching for identification of network traffic behaviour throughout the LAN environment and showing suspicious anomalies [36]. This anomaly provides information about the unusual interactions in several protocols such as SMB2, NBSS, and TCP. Viruses generally use access to specific protocols to implement certain conversations and exchange information between two devices.

Table 11. The comparison of network protocols used

No	Protocol	Infection (%)	Mitigation (%)
1	SMB	0.1	0.13
2	SMB2	4.04	5.47
3	TCP	91.1	73.97
4	UDP	0.06	0.04
5	ICMP	0.12	0.16
6	NBSS	0.04	20.16
7	NBNS	0.12	0.06
8	DNS	4.43	0

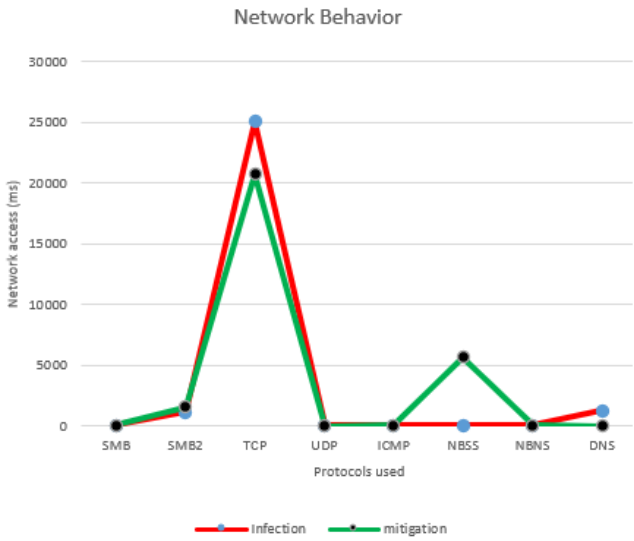


Figure 11. Behavioural network-based mitigation

5. CONCLUSIONS

Data obtained in SDN-based Sodinokibi Ransomware attack mitigation research has successfully prevented further spread and risk. The mitigation concept offered in this study can be a reference in dealing with cyber threats such as network behaviour carried out by malware and DDOS attacks through a switch system checking the compatibility of each incoming package with its table, and only the highest priority entry flows take precedence. The weakness in this mitigation is that the development of commands and applications available on the SDN Ryu controller is still limited to TCP, UDP, and ICMP protocols. The action in the detection section can address this weakness. The detection system can be developed again by maximizing between intrusion detection system, Yara rule, SDN controller, and SDN mirroring concept to allow the network to remain stable when blocking infected computers.

REFERENCES

[1] Baek, S., Jung, Y., Mohaisen, A., Lee, S., Nyang, D. (2020). SSD-assisted ransomware detection and data recovery techniques. IEEE Transactions on Computers. https://doi.org/10.1109/tc.2020.3011214

[2] Bae, S.I., Lee, G.B., Im, E.G. (2020). Ransomware detection using machine learning algorithms. Concurrency and Computation: Practice and Experience, 32(18): e5422. https://doi.org/10.1002/cpe.5422

[3] Ransomware Payments Up 33% As Maze and Sodinokibi Proliferate in Q1 2020 - Security Boulevard. https://securityboulevard.com/2020/04/ransomware-payments-up-33-as-maze-and-sodinokibi-proliferate-in-q1-2020/, accessed on Mar. 26, 2021.

[4] Ransomware actor netted over $123 million in 2020. https://news.yahoo.com/ransomware-actor-netted-over-123-175142446.html?guccounter=1, accessed on Mar. 26, 2021.

[5] Coveware, C. (2021). Key Trends 75% of all attack Ransomware 80% of attack exfiltrate data Ransomware Exfiltration Average Size of Organization Ransomware Exfiltration Techniques, 2020: 2020–2021.

[6] REvil Ransomware Attacks Food Distributors; Hackers Seek $7.5M Ransom - MSSP Alert. https://www.msspalert.com/cybersecurity-breaches-and-attacks/ransomware/revil-hits-food-distributors/, accessed on Mar. 26, 2021.

[7] Fadlil, A., Riadi, I., Nugrahantoro, A. (2020). Data security for school service top-up transactions based on AES combination blockchain technology. Modification, 11(3): 155-166. https://doi.org/10.24843/lkjiti.2020.v11.i03.p04

[8] Adamov, A., Carlsson, A. (2020). Reinforcement learning for anti-ransomware testing. In 2020 IEEE East-West Design & Test Symposium (EWDTS), pp. 1-5. https://doi.org/10.1109/EWDTS50664.2020.9225141

[9] Pillai, A., Kadikar, R., Vasanthi, M.S., Amutha, B. (2018). Analysis of AES-CBC encryption for interpreting crypto-wall ransomware. In 2018 International Conference on Communication and Signal Processing (ICCSP), pp. 0599-0604. https://doi.org/10.1109/ICCSP.2018.8524494

[10] Sodinokibi - Malware Trends Tracker by ANY.RUN. https://any.run/malware-trends/sodinokibi, accessed on Mar. 26, 2021.

[11] Mohammad, A.H. (2020). Ransomware evolution, growth and recommendation for detection. Modern Applied Science, 14(3): 68. https://doi.org/10.5539/mas.v14n3p68

[12] Adamov, A., Carlsson, A. (2017). The state of ransomware. Trends and mitigation techniques. In 2017 IEEE East-West Design & Test Symposium (EWDTS), pp. 1-8. https://doi.org/10.1109/EWDTS.2017.8110056

[13] Almashhadani, A.O., Kaiiali, M., Sezer, S., O'Kane, P. (2019). A multi-classifier network-based crypto

ransomware detection system: A case study of Locky ransomware. IEEE Access, 7: 47053-47067. https://doi.org/10.1109/ACCESS.2019.2907485
[14] Kurniawan, A., Riadi, I. (2018). Detection and analysis cerber ransomware based on network forensics behavior. International Journal of Network Security, 20(5): 836-843. https://doi.org/10.6633/IJNS.201809_20(5).04
[15] Chakkaravarthy, S.S., Sangeetha, D., Cruz, M.V., Vaidehi, V., Raman, B. (2020). Design of intrusion detection honeypot using social leopard algorithm to detect IoT Ransomware attacks. IEEE Access, 8: 169944-169956. https://doi.org/10.1109/access.2020.3023764
[16] Xia, T., Sun, Y., Zhu, S., Rasheed, Z., Hassan-Shafique, K. (2000). A network-assisted approach for ransomware detection. arXiv: 2008.12428.
[17] Agrawal, R., Stokes, J.W., Selvaraj, K., Marinescu, M. (2019). University of California, Santa Cruz, Santa Cruz, CA 95064 USA Microsoft Corp. One Microsoft Way, Redmond, WA 98052 USA, 3222-3226.
[18] Ganfure, G.O., Wu, C.F., Chang, Y.H., Shih, W.K. (2020). DeepGuard: Deep generative user-behavior analytics for ransomware detection. In 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 1-6. https://doi.org/10.1109/ISI49825.2020.9280508
[19] Ayub, M.A., Continella, A., Siraj, A. (2020). An I/O Request Packet (IRP) driven effective ransomware detection scheme using artificial neural network. In 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI), pp. 319-324. https://doi.org/10.1109/IRI49571.2020.00053
[20] Fakiha, B. (2021). Business organization security strategies to cyber security threats. International Journal of Safety and Security Engineering, 11(1): 101-104. https://doi.org/10.18280/ijsse.110111
[21] Akbanov, M., Vassilakis, V.G., Logothetis, M.D. (2019). Ransomware detection and mitigation using software-defined networking: The case of WannaCry. Computers & Electrical Engineering, 76: 111-121. https://doi.org/10.1016/j.compeleceng.2019.03.012
[22] Keijzer, N. (2020). The new generation of ransomware: An in depth study of Ransomware-as-a-Service. Master's thesis, University of Twente.
[23] Berrueta, E., Morato, D., Magana, E., Izal, M. (2019). A survey on detection techniques for cryptographic ransomware. IEEE Access, 7: 144925-144944. https://doi.org/10.1109/ACCESS.2019.2945839
[24] Kodli, S., Terdal, S. (2021). Hybrid max-min genetic algorithm for load balancing and task scheduling in cloud environment. Intelligent Engineering & Systems, 14(1): 63-71. https://doi.org/10.22266/IJIES2021.0228.07.
[25] Davies, SR., Macfarlane, R., Buchanan, W.J. (2020). Evaluation of live forensic techniques in ransomware attack mitigation. Forensic Science International: Digital Investigation, 33: 300979. https://doi.org/10.1016/j.fsidi.2020.300979
[26] Leng, B., Huang, L., Qiao, C., Xu, H., Wang, X. (2017). FTRS: A mechanism for reducing flow table entries in software defined networks. Computer Networks, 122: 1-15. https://doi.org/10.1016/j.comnet.2017.04.022.
[27] Spiekermann, D., Keller, J., Eggendorfer, T. (2017). Network forensic investigation in OpenFlow networks with ForCon. Digital Investigation, 20: S66-S74. https://doi.org/10.1016/j.diin.2017.01.007
[28] Salib, E.H., Lester, J.D. (2018). Hands-on labs and tools for teaching software defined network (SDN) to undergraduates. In 2018 ASEE Annual Conference & Exposition. https://doi.org/10.18260/1-2--30570
[29] Sanjab, A., Saad, W., Guvenc, I., Sarwat, A., Biswas, S. (2016). Smart grid security: Threats, challenges, and solutions. arXiv preprint arXiv:1606.06992.
[30] Kok, S.H., Abdullah, A., Jhanjhi, N.Z. (2020). Early detection of crypto-ransomware using pre-encryption detection algorithm. Journal of King Saud University-Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2020.06.012
[31] Kusuma, R.S., Umar, R., Riadi, I. (2021). Network Forensics Against Ryuk Ransomware Using Trigger, Acquire, Analysis, Report, and Action (TAARA) Method. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control.
[32] Sirisha, A., Chaitanya, K., Krishna, K.V.S.S.R., Kanumalli, S.S. (2021). Intrusion detection models using supervised and unsupervised algorithms - a comparative estimation. International Journal of Safety and Security Engineering, 11(1): 51-58. https://doi.org/10.18280/ijsse.110106
[33] Alwan, K.M., AbuEl-Atta, AH., Zayed, H. (2020) Feature selection models based on hybrid firefly algorithm with mutation operator for network intrusion detection. Intelligent Engineering & Systems, 14(1): 192-202. https://doi.org/10.22266/IJIES2021.0228.19
[34] Arabo, A., Dijoux, R., Poulain, T., Chevalier, G. (2020). Detecting ransomware using process behavior analysis. Procedia Computer Science, 168: 289-296. https://doi.org/10.1016/j.procs.2020.02.249.
[35] Usman, L., Prayudi, Y., Riadi, I. (2017). Ransomware analysis based on the surface, runtime and static code method. Journal of Theoretical & Applied Information Technology, 95(11).
[36] Manzano, C., Meneses, C., Leger, P. (2020). An empirical comparison of supervised algorithms for ransomware identification on network traffic. In 2020 39th International Conference of the Chilean Computer Science Society (SCCC), pp. 1-7. https://doi.org/10.1109/SCCC51225.2020.9281283