

```

C:\Snort\Rules\local.rules - Notepad++
File Edit Search View Encoding Language Settings Tools Macros Run Plugins Window ?
LOCAL RULES

# Rule 1: Logging all incoming and outgoing ICMP Packets from any IP Address
alert icmp any any -> any any (msg: "testing ICMP"; sid:1000001)

# Rule 2: Logging all incoming and outgoing TCP Packets from any IP Address
alert tcp any any -> any any (msg: "testing ICMP"; sid:1000002)

# Rule 3: Logging all incoming and outgoing UDP Packets from any IP Address
alert udp any any -> any any (msg: "testing UDP"; sid:1000003)

# Rule 4: Alerting User of any FIF Connections Attempt
alert tcp any any -> CHROME_NEX_21 (msg: "FIF Connection Attempt"; sid:1000004)

#Rule 5: Alerting on Specific IP (SASTRA in this Case) for SSH Connection Attempt
ipvar SASTRA 192.168.10.11
alert tcp any any -> SASTRA_22 (msg: "SSH Connection Attempted on SASTRA"; sid:1000005)

# Rule 6: Alert on any Specific Website Visited (Google in this case):
ipvar GOOGLE 142.250.195.210
alert tcp any any -> GOOGLE_HTTP_PORTS (msg: "Test google.com rule"; content:"google.com"; nocase; sid:1000006)

#Rule 7: Alerting User of any SQL Injection Attacks
alert tcp any any -> CHROME_NEX_52 (msg: "Possible SQL Injection Attempt"; flow:to_server,established; content:" OR '1'='1"; nocase; sid:1000007)

# Rule 8: Alerting User of any Cross Site Scripting Attacks (XSS)
alert tcp any any -> any 80 (msg: "Cross-Site Scripting (XSS) Attack"; flow:to_server,established; content:"<script>"; nocase; pcre:"/<script>.</script>/iU");

```

Fig 4a.a - Customized rules

```

Administrator: Command Prompt
Preprocessor Object: SF_I4S Version 1.1 <Build 4>
Preprocessor Object: SF_DKP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Commencing packet processing (pid=27804)
00/22-19:05:52.634182 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.634182 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.634182 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.640221 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2404:6860:4067:0812:0000:0000:200e:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1908
00/22-19:05:52.640221 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2404:6860:4067:0812:0000:0000:200e:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1908
00/22-19:05:52.640365 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1806
00/22-19:05:52.643515 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1806
00/22-19:05:52.680335 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 192.19.234.39:443 -> 192.168.55.212:1182
00/22-19:05:52.680510 [**] [1:1000003:0] &CfTesting UDP&CfC0 [**] [Priority: 0] {UDP} 8.211.162.182.22101 -> 192.168.55.212:55511
00/22-19:05:52.731262 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1011
00/22-19:05:52.732247 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.732247 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.732247 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.737778 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 192.19.234.39:443 -> 192.168.55.212:1182
00/22-19:05:52.821596 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1806
00/22-19:05:52.821596 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1806
00/22-19:05:52.872517 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.872517 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1806
00/22-19:05:52.872517 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.872517 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1806
00/22-19:05:52.872517 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.872517 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.872517 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.947452 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811
00/22-19:05:52.947452 [**] [1:1000002:0] &CfTesting ICMP&CfC0 [**] [Priority: 0] {TCP} 2405:0200:1630:ff12:0000:0000:000c:443 -> 2409:408d:3c14:67c2:5cf6:7063:3eb7:2cac:1811

```

Fig 4a.b - Alerts raised for the rules in command prompt

Aim:

To apply customized rules and detecting the appropriate alerts through the Snort Intrusion Detection Tool

Procedure:

- I) Create customized rules in local.rules in "rules folder"
- II) Configure Snort for the following Ruleactions:
 - I) ALERT: Generates an alert when suspicious packet is detected.
 - II) BLOCK: Blocks the suspicious & all subsequent packets coming from that network flow of any protocol specified.
 - III) DROP: Drops the packet as soon as alert is generated.
 - IV) LOG: Logs the details of packet as soon as alert is generated.
 - V) PASS: Ignores the suspicious packet & marks it as "Passed"

III) Once customized rules according to the above rule patterns are written, save the file and execute the Command

Snort -i <Interface Index> -C <Conf file path> -A console.

Result:

Thus, we analyzed & implemented custom rules for Intrusion Detection Systems using Snort in Windows 11 Operating System.

Teacher's Signature:

Aim:

To apply Signature Based Rule Permission (Allow & disallow coding in particular app)

Procedure:

We are doing Coding for "File Permission" for extracting & displaying file permissions and access control for given file by user

language: Python

module used: pywin32, os, win32Security

- 1) First, we import the modules, os and win32Security mainly for file system information and windows security settings
- 2) Defining SDDL to be translated as a suitable function
Its a way to represent Security Systems in windows
fn: ~~sdet~~.translate_sddl, takes SDDL String as input & translates to human readable format, by examining access control list (ACL).
- 3) Defining mainfn for main logic, to get file path as well as exit command input from user.
- 4) Retrieving file permission: when user provides file path, code attempts to retrieve & disp permission for that file using "win32Security.GetfileSecurity" function including discretionary access control list which houses the permission.
- 5) Displaying file permission: Script extracts info from DACL including trustee and access rights, displays it to us as output

Teacher's Signature:

OUTPUT SNIP :-

સ્પર્ધાની "no1234567890" નાં પાબાં પ્રિફ્ટ કરી શકતું હોય
(બેસાનું દોષ નથો)

```
File Edit Shell Debug Options Window Help
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: D:\sastra\IV Year\FIDS\FilePermissions.py =====
Enter the path of the file (or 'exit' to quit): C:\Program Files\Cisco Packet Tracer 8.2.0\unins000.exe
File Permissions for 'C:\Program Files\Cisco Packet Tracer 8.2.0\unins000.exe':
Trustee: PySID:S-1-5-18, Access Rights: 2032127
Trustee: PySID:S-1-5-32-544, Access Rights: 2032127
Trustee: PySID:S-1-5-32-545, Access Rights: 1179817
Trustee: PySID:S-1-15-2-1, Access Rights: 1179817
Trustee: PySID:S-1-15-2-2, Access Rights: 1179817
Enter the path of the file (or 'exit' to quit): C:\Drivers\network\CCXT7\drivers\Production\Windows10-x64\W
LAN_driver\qcmainext10x.cat
File Permissions for 'C:\Drivers\network\CCXT7\drivers\Production\Windows10-x64\WLAN_driver\qcmainext10x.c
at':
Trustee: PySID:S-1-5-32-544, Access Rights: 2032127
Trustee: PySID:S-1-5-18, Access Rights: 2032127
Trustee: PySID:S-1-5-32-545, Access Rights: 1179817
Trustee: PySID:S-1-5-11, Access Rights: 1245631
Enter the path of the file (or 'exit' to quit): exit
>>>
```

6)

If any errors are detected in the process, such as lack of permission, file in existant and soon, error msg will be shown in form of access right code.

7)

Enter file path or exit after getting results from the program.

Program:

```
import os
import win32 security
def translate_sddl(sddl):
    try:
        Sd = win32 security.ACL()
        win32 security.SetSecurityDescriptorDacl(Sd, True, win32 security.ACL())
        win32 security.SetSecurityDescriptorSddlDacl(Sd, sddl, False)

        dacl = Sd.GetSecurityDescriptorDacl()
        ace = dacl.GetAce(0)
        permissions = []
        for i in range(dacl.GetAceCount()):
            trustee, access_rights_ = ace
            permissions.append((trustee, access_rights_))
            ace = dacl.GetAce(i+1)
        return permissions
    except Exception as e:
        return str(e)
```

```
def main():
    while True:
```

file_Path = input("Enter path of file (or 'exit' to quit):")

if file_Path.lower() == 'exit':

break

try:

Teacher's Signature:

22/07/2023

Sd = win32Security.GetFileSecurity(file_path, win32Security.Owner_SECURITY_INFORMATION)
- INFORMATION | win32Security.DACL-SECURITY INFORMATION
dacl = sd.GetSecurityDescriptorDacl()

permissions []

for i in range(dacl.GetAceCount()):

ace = dacl.GetAce(i)

trustee = ace[2]

access_rights = ace[1]

permissions.append((trustee, access_rights))

print(f"File permission for '{file_Path}':")

for trustees, access_rights in permissions:

print(f"Trustee: {trustee}, Access Rights: {access_rights}")

except Exception as e:

print(f"Error: {str(e)}")

if __name__ == "__main__":

main()

Result:

Finally, Successfully applied Signature based role permission

Teacher's Signature: