Expanding Chainsaw

Tool: https://github.com/countercept/chainsaw
Documentation: https://github.com/countercept/chainsaw/tree/documentation_improvements

Description:
Chainsaw provides a powerful 'first-response' capability to quickly identify threats within Windows event logs. It offers a generic and fast method of searching through event logs for keywords, and by identifying threats using built-in detection logic and via support for Sigma detection rules.

Following a recent request, the developer of Chainsaw provided some documentation to help users expand on the tool. This included how the Chainsaw mappings worked and how to link them to sigma rules and also how to create your own mappings and associate them with rules of interest (either preexisting or bespoke self creations)

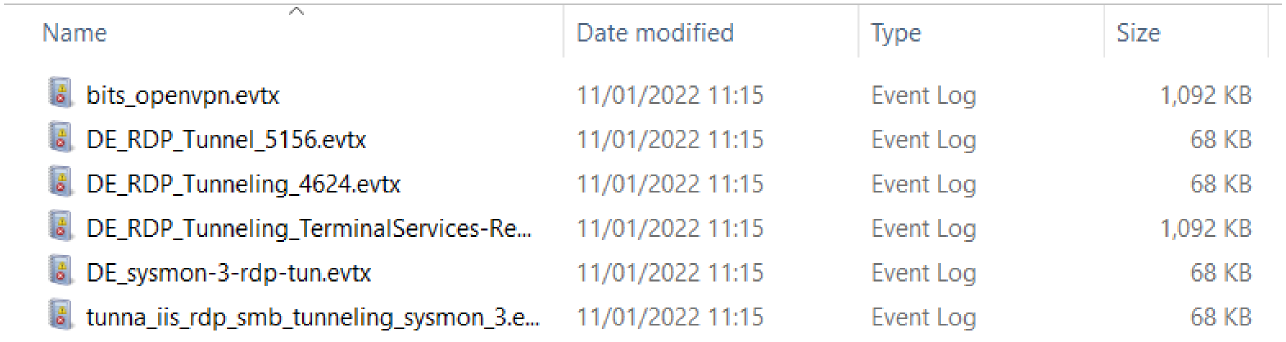
To help support this new guidance documentation and help understand the mappings better myself, I have run some tests with some new mappings and sigma rules I have created and thought I'd write up and step by step as to how I achieved this.

Firstly, I created a very simple rule that simply looks at my own workstations Security.evtx for a 4624 login from my username and local loopback address as I knew it existed and wanted to see if it appeared in the chainsaw standard out in the terminal amongst it's usual built in logic returns.

This was relatively straightforward and I was pleased to see, was successful. I also tested using the –csv switch and successfully outputted the results .csv including my test rule.

Following my success with this somewhat easy and useless rule, I decided I would use one of Chainsaw's provided EVTX attack examples to create the necessary mappings and rules to detect all of the known IOCs from these provided .evtx files. The attack example I chose was:
https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES/tree/b947ed80cd67a7413f6689d032eb1151f85f9df6/Command%20and%20Control

The original evtx files were all contained within a folder with a pcap, log files and a README.md. To save the frustration of potential conflicts or errors when pointing chainsaw at this folder, I moved the evtx files into their own directory leaving me with:



Name	Date modified	Type	Size
bits_openvpn.evtx	11/01/2022 11:15	Event Log	1,092 KB
DE_RDP_Tunnel_5156.evtx	11/01/2022 11:15	Event Log	68 KB
DE_RDP_Tunneling_4624.evtx	11/01/2022 11:15	Event Log	68 KB
DE_RDP_Tunneling_TerminalServices-Re...	11/01/2022 11:15	Event Log	1,092 KB
DE_sysmon-3-rdp-tun.evtx	11/01/2022 11:15	Event Log	68 KB
tunna_iis_rdp_smb_tunneling_sysmon_3.e...	11/01/2022 11:15	Event Log	68 KB

I then focused on the 4 DE_RDP evtx files for the purposes of my testing.

The README.md provided the following information:

```
Connection Proxy:
1. RDP Tunneling via SSH - eventid 4624 - Logon Type 10 and Source IP eq to loopback IP address
2. RDP Tunneling via SSH - eventid 1149 - TerminalServices-RemoteConnectionManagerOperational - RDP source IP loopback IP address
3. RDP Tunneling via SSH - Sysmon eventid 3 - local port forwarding to/from loopback IP (svchost.exe <-> plink.exe)
4. RDP Tunneling via SSH - eventid 5156 - local port forwarding to/from loopback IP to 3389 rdp port
```

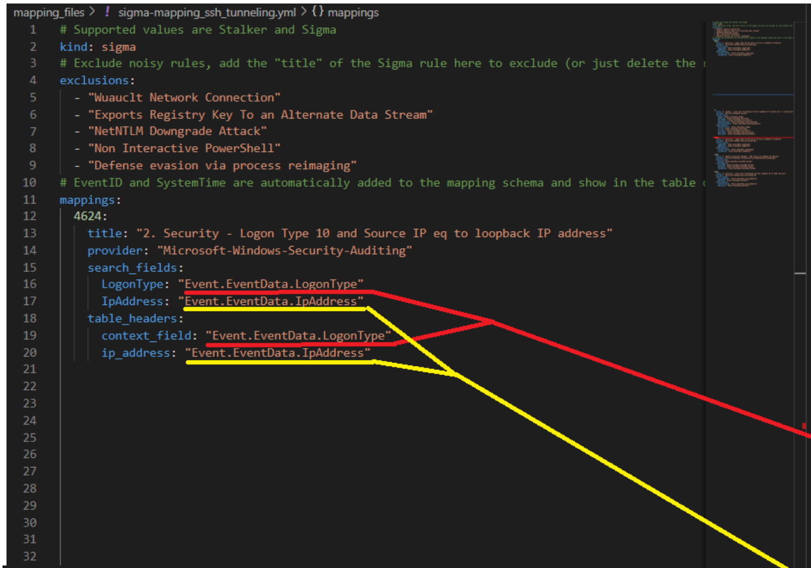
NOTE: I'm using these .evtx samples simply as a reference point to create the mappings and rules. This is not meant to represent the best or definitive way for writing these mappings/rules for this specific IOC. If you would like to learn more about RDP tunneling, here is a great article: https://www.real-sec.com/2019/04/bypassing-network-restrictions-through-rdp-tunneling/

STEP 1 - EventID 4634

`RDP Tunneling via SSH - eventid 4624 - Logon Type 10 and Source IP eq to loopback IP address`

I needed to create a rule that would look for EventID 4624 for Logon Type 10 with the Source IP as the loopback IP

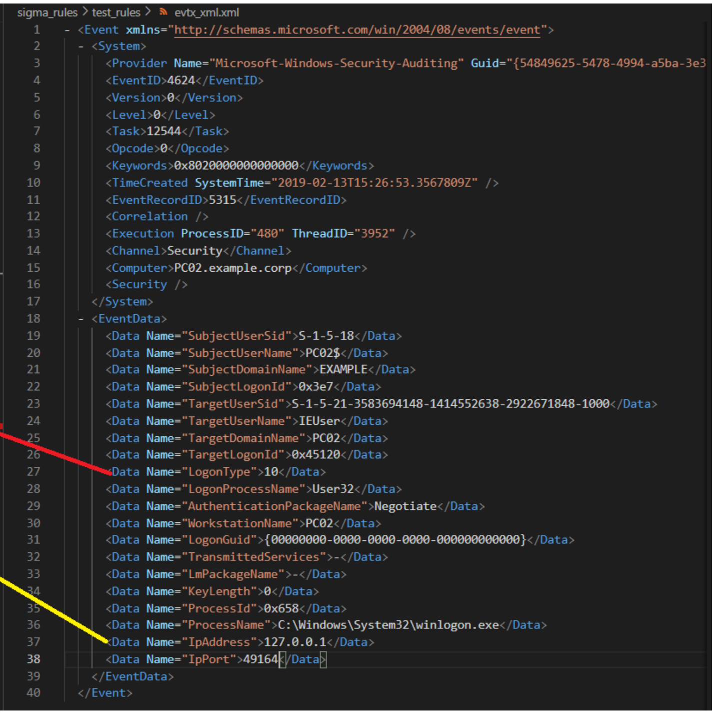
LEFT: mappings file



mapping_files > ! sigma-mapping_ssh_tunneling.yml > {} mappings

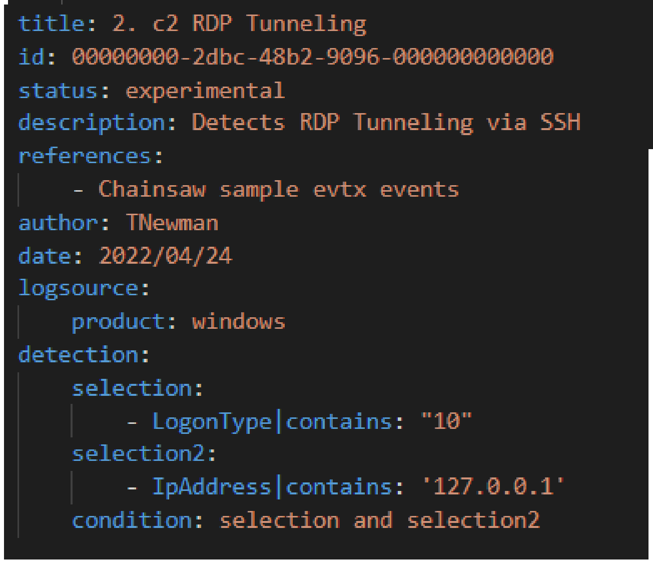
```yaml
# Supported values are Stalker and Sigma
kind: sigma
# Exclude noisy rules, add the "title" of the Sigma rule here to exclude (or just delete the
exclusions:
  - "Wuauclt Network Connection"
  - "Exports Registry Key To an Alternate Data Stream"
  - "NetNTLM Downgrade Attack"
  - "Non Interactive PowerShell"
  - "Defense evasion via process reimaging"
# EventID and SystemTime are automatically added to the mapping schema and show in the table
mappings:
  4624:
    title: "2. Security - Logon Type 10 and Source IP eq to loopback IP address"
    provider: "Microsoft-Windows-Security-Auditing"
    search_fields:
      LogonType: "Event.EventData.LogonType"
      IpAddress: "Event.EventData.IpAddress"
    table_headers:
      context_field: "Event.EventData.LogonType"
      ip_address: "Event.EventData.IpAddress"
```

RIGHT: XML of EventID 4624 (Security.evtx)



sigma_rules > test_rules > evtx_xml.xml

```xml
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  - <System>
      <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3
      <EventID>4624</EventID>
      <Version>0</Version>
      <Level>0</Level>
      <Task>12544</Task>
      <Opcode>0</Opcode>
      <Keywords>0x8020000000000000</Keywords>
      <TimeCreated SystemTime="2019-02-13T15:26:53.3567809Z" />
      <EventRecordID>5315</EventRecordID>
      <Correlation />
      <Execution ProcessID="480" ThreadID="3952" />
      <Channel>Security</Channel>
      <Computer>PC02.example.corp</Computer>
      <Security />
    </System>
  - <EventData>
      <Data Name="SubjectUserSid">S-1-5-18</Data>
      <Data Name="SubjectUserName">PC02$</Data>
      <Data Name="SubjectDomainName">EXAMPLE</Data>
      <Data Name="SubjectLogonId">0x3e7</Data>
      <Data Name="TargetUserSid">S-1-5-21-3583694148-1414552638-2922671848-1000</Data>
      <Data Name="TargetUserName">IEUser</Data>
      <Data Name="TargetDomainName">PC02</Data>
      <Data Name="TargetLogonId">0x45120</Data>
      <Data Name="LogonType">10</Data>
      <Data Name="LogonProcessName">User32</Data>
      <Data Name="AuthenticationPackageName">Negotiate</Data>
      <Data Name="WorkstationName">PC02</Data>
      <Data Name="LogonGuid">{00000000-0000-0000-0000-000000000000}</Data>
      <Data Name="TransmittedServices">-</Data>
      <Data Name="LmPackageName">-</Data>
      <Data Name="KeyLength">0</Data>
      <Data Name="ProcessId">0x658</Data>
      <Data Name="ProcessName">C:\Windows\System32\winlogon.exe</Data>
      <Data Name="IpAddress">127.0.0.1</Data>
      <Data Name="IpPort">49164</Data>
    </EventData>
  </Event>
```



```yaml
title: 2. c2 RDP Tunneling
id: 00000000-2dbc-48b2-9096-000000000000
status: experimental
description: Detects RDP Tunneling via SSH
references:
    - Chainsaw sample evtx events
author: TNewman
date: 2022/04/24
logsource:
    product: windows
detection:
    selection:
        - LogonType|contains: "10"
    selection2:
        - IpAddress|contains: '127.0.0.1'
    condition: selection and selection2
```

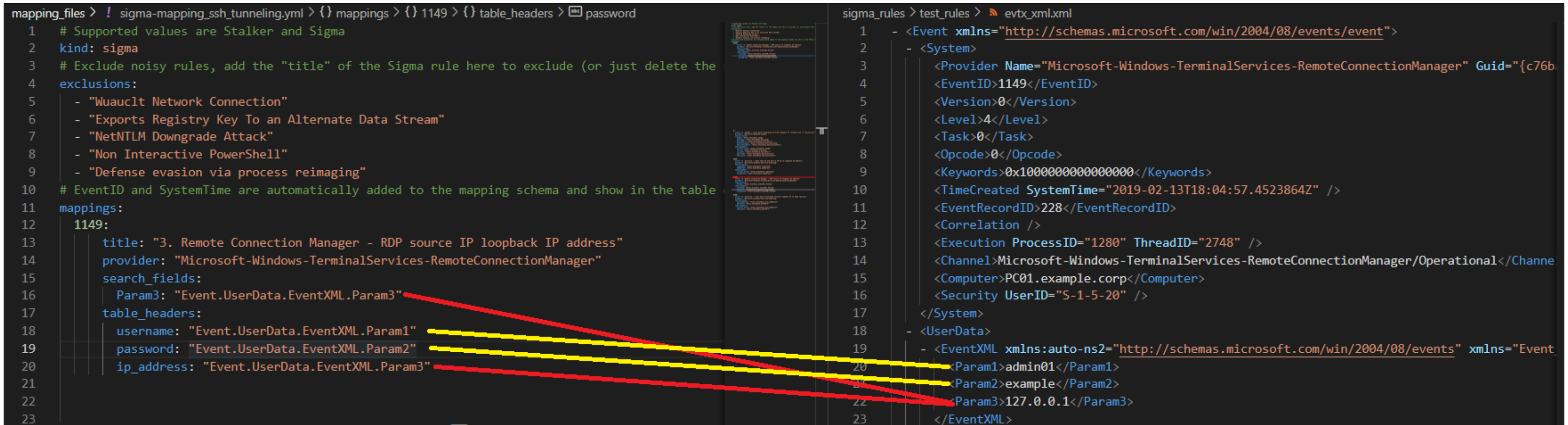
LEFT: Sigma Rule created to look for our specific IOCs:

detection>selection>LogonType
detection>selection2>IpAddress
^^This is where we apply our actual IOCs (Logon Type 10 & loopback IP)
Condition == rule must match both detections together

STEP 2 - EventID 1149

2. RDP Tunneling via SSH - eventid 1149 - TerminalServices-RemoteConnectionManagerOperational - RDP source IP loopback IP address

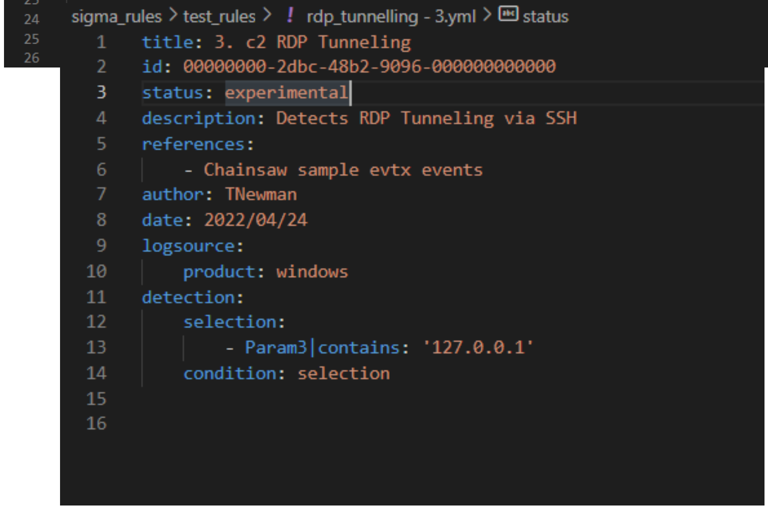
Next is EventID 1149 from the Remote Connection Manager evtx looking for the local loopback address



mapping_files > sigma-mapping_ssh_tunneling.yml > mappings > 1149 > table_headers > password

```yaml
# Supported values are Stalker and Sigma
kind: sigma
# Exclude noisy rules, add the "title" of the Sigma rule here to exclude (or just delete the
exclusions:
  - "Wuauclt Network Connection"
  - "Exports Registry Key To an Alternate Data Stream"
  - "NetNTLM Downgrade Attack"
  - "Non Interactive PowerShell"
  - "Defense evasion via process reimaging"
# EventID and SystemTime are automatically added to the mapping schema and show in the table
mappings:
  1149:
      title: "3. Remote Connection Manager - RDP source IP loopback IP address"
      provider: "Microsoft-Windows-TerminalServices-RemoteConnectionManager"
      search_fields:
        Param3: "Event.UserData.EventXML.Param3"
      table_headers:
        username: "Event.UserData.EventXML.Param1"
        password: "Event.UserData.EventXML.Param2"
        ip_address: "Event.UserData.EventXML.Param3"
```

```xml
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  - <System>
      <Provider Name="Microsoft-Windows-TerminalServices-RemoteConnectionManager" Guid="{c76b
      <EventID>1149</EventID>
      <Version>0</Version>
      <Level>4</Level>
      <Task>0</Task>
      <Opcode>0</Opcode>
      <Keywords>0x1000000000000000</Keywords>
      <TimeCreated SystemTime="2019-02-13T18:04:57.4523864Z" />
      <EventRecordID>228</EventRecordID>
      <Correlation />
      <Execution ProcessID="1280" ThreadID="2748" />
      <Channel>Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational</Channe
      <Computer>PC01.example.corp</Computer>
      <Security UserID="S-1-5-20" />
    </System>
  - <UserData>
    - <EventXML xmlns:auto-ns2="http://schemas.microsoft.com/win/2004/08/events" xmlns="Event_
        <Param1>admin01</Param1>
        <Param2>example</Param2>
        <Param3>127.0.0.1</Param3>
      </EventXML>
    </UserData>
</Event>
```



```yaml
title: 3. c2 RDP Tunneling
id: 00000000-2dbc-48b2-9096-000000000000
status: experimental
description: Detects RDP Tunneling via SSH
references:
    - Chainsaw sample evtx events
author: TNewman
date: 2022/04/24
logsource:
    product: windows
detection:
    selection:
      - Param3|contains: '127.0.0.1'
    condition: selection
```

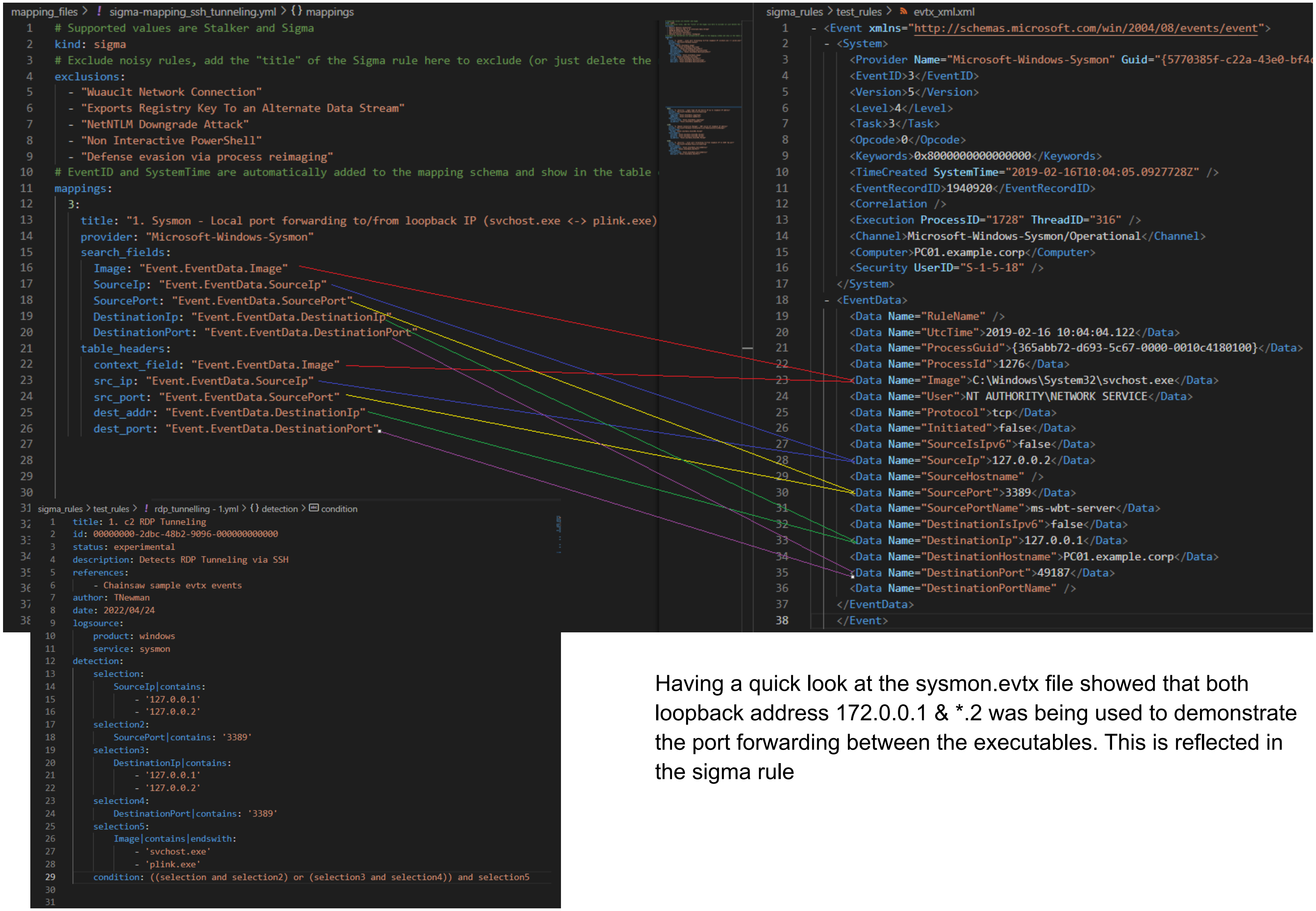
You'll notice with this mapping (top left), I have used additional parameters under the table_headers. These are the values we want chainsaw to output to the terminal and in this case I asked for the **username/password** as well as our searchable parameter, the **loopback IP**

The sigma rule only requires our searchable fields from the mapping file.

STEP 3 - EventID 3

3. RDP Tunneling via SSH - Sysmon eventid 3 - local port forwarding to/from loopback IP (svchost.exe <-> plink.exe)

Sysmon EventID 3 is next. This is looking for port forwarding to and from our loopback IP between svchost and plink.exe



mapping_files > ! sigma-mapping_ssh_tunneling.yml > {} mappings

```yaml
# Supported values are Stalker and Sigma
kind: sigma
# Exclude noisy rules, add the "title" of the Sigma rule here to exclude (or just delete the
exclusions:
  - "Wuauclt Network Connection"
  - "Exports Registry Key To an Alternate Data Stream"
  - "NetNTLM Downgrade Attack"
  - "Non Interactive PowerShell"
  - "Defense evasion via process reimaging"
# EventID and SystemTime are automatically added to the mapping schema and show in the table
mappings:
  3:
    title: "1. Sysmon - Local port forwarding to/from loopback IP (svchost.exe <-> plink.exe)
    provider: "Microsoft-Windows-Sysmon"
    search_fields:
      Image: "Event.EventData.Image"
      SourceIp: "Event.EventData.SourceIp"
      SourcePort: "Event.EventData.SourcePort"
      DestinationIp: "Event.EventData.DestinationIp"
      DestinationPort: "Event.EventData.DestinationPort"
    table_headers:
      context_field: "Event.EventData.Image"
      src_ip: "Event.EventData.SourceIp"
      src_port: "Event.EventData.SourcePort"
      dest_addr: "Event.EventData.DestinationIp"
      dest_port: "Event.EventData.DestinationPort"
```

sigma_rules > test_rules > ! rdp_tunnelling - 1.yml > {} detection > condition

```yaml
title: 1. c2 RDP Tunneling
id: 00000000-2dbc-48b2-9096-000000000000
status: experimental
description: Detects RDP Tunneling via SSH
references:
    - Chainsaw sample evtx events
author: TNewman
date: 2022/04/24
logsource:
    product: windows
    service: sysmon
detection:
    selection:
        SourceIp|contains:
            - '127.0.0.1'
            - '127.0.0.2'
    selection2:
        SourcePort|contains: '3389'
    selection3:
        DestinationIp|contains:
            - '127.0.0.1'
            - '127.0.0.2'
    selection4:
        DestinationPort|contains: '3389'
    selection5:
        Image|contains|endswith:
            - 'svchost.exe'
            - 'plink.exe'
    condition: ((selection and selection2) or (selection3 and selection4)) and selection5
```

sigma_rules > test_rules > evtx_xml.xml

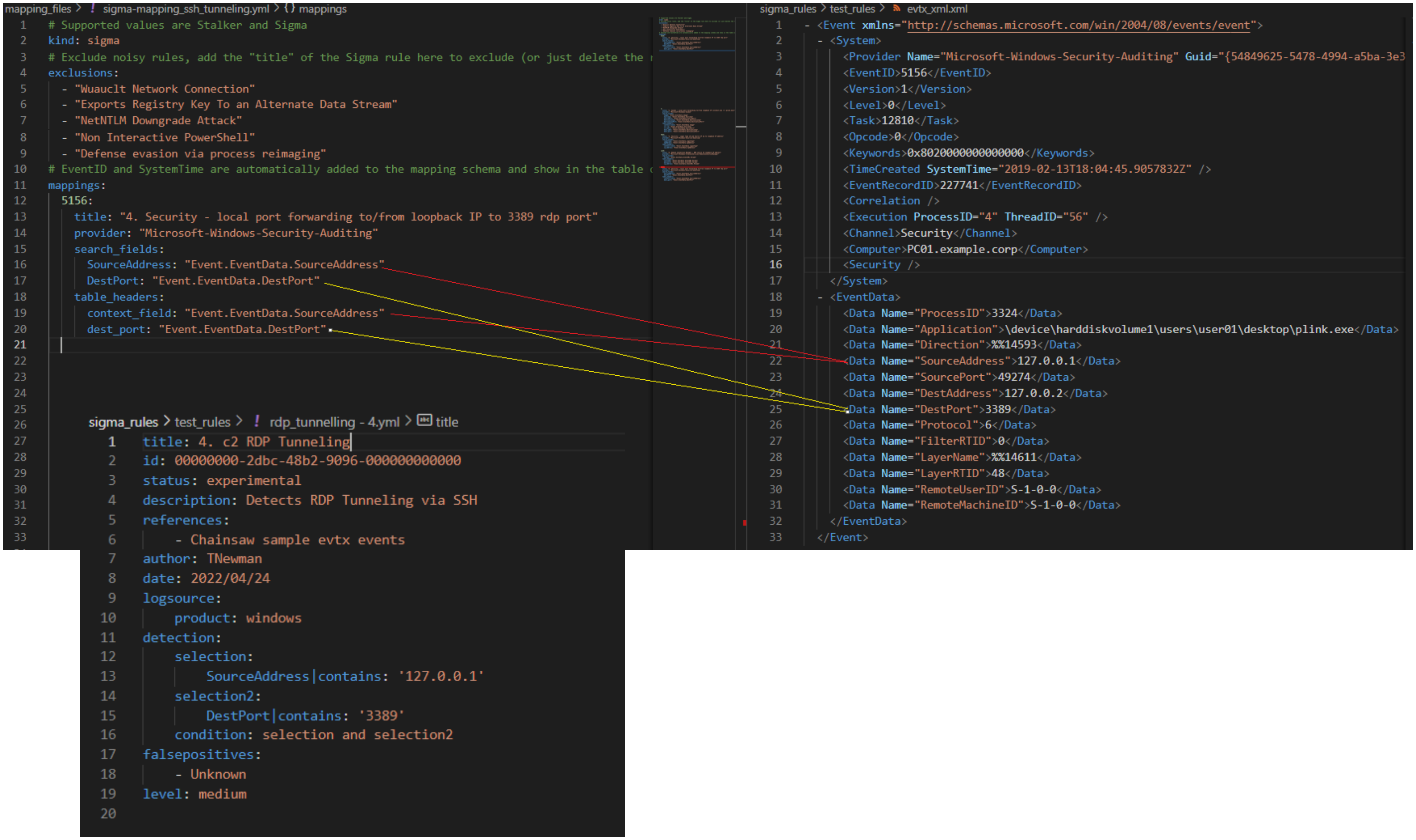
```xml
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  - <System>
      <Provider Name="Microsoft-Windows-Sysmon" Guid="{5770385f-c22a-43e0-bf4c
      <EventID>3</EventID>
      <Version>5</Version>
      <Level>4</Level>
      <Task>3</Task>
      <Opcode>0</Opcode>
      <Keywords>0x8000000000000000</Keywords>
      <TimeCreated SystemTime="2019-02-16T10:04:05.0927728Z" />
      <EventRecordID>1940920</EventRecordID>
      <Correlation />
      <Execution ProcessID="1728" ThreadID="316" />
      <Channel>Microsoft-Windows-Sysmon/Operational</Channel>
      <Computer>PC01.example.corp</Computer>
      <Security UserID="S-1-5-18" />
    </System>
  - <EventData>
      <Data Name="RuleName" />
      <Data Name="UtcTime">2019-02-16 10:04:04.122</Data>
      <Data Name="ProcessGuid">{365abb72-d693-5c67-0000-0010c4180100}</Data>
      <Data Name="ProcessId">1276</Data>
      <Data Name="Image">C:\Windows\System32\svchost.exe</Data>
      <Data Name="User">NT AUTHORITY\NETWORK SERVICE</Data>
      <Data Name="Protocol">tcp</Data>
      <Data Name="Initiated">false</Data>
      <Data Name="SourceIsIpv6">false</Data>
      <Data Name="SourceIp">127.0.0.2</Data>
      <Data Name="SourceHostname" />
      <Data Name="SourcePort">3389</Data>
      <Data Name="SourcePortName">ms-wbt-server</Data>
      <Data Name="DestinationIsIpv6">false</Data>
      <Data Name="DestinationIp">127.0.0.1</Data>
      <Data Name="DestinationHostname">PC01.example.corp</Data>
      <Data Name="DestinationPort">49187</Data>
      <Data Name="DestinationPortName" />
    </EventData>
  </Event>
```

Having a quick look at the sysmon.evtx file showed that both loopback address 172.0.0.1 & *.2 was being used to demonstrate the port forwarding between the executables. This is reflected in the sigma rule

STEP 4 - EventID 5156

`4. RDP Tunneling via SSH - eventid 5156 - local port forwarding to/from loopback IP to 3389 rdp port`

Finally, EventID 5156 from Security.evtx is similar to Sysmon EventID 3 but with less information so as to more easily differentiate between the two I'm only looking for plink.exe outbound traffic to port 3389



mapping_files > sigma-mapping_ssh_tunneling.yml > {} mappings

```yaml
# Supported values are Stalker and Sigma
kind: sigma
# Exclude noisy rules, add the "title" of the Sigma rule here to exclude (or just delete the r
exclusions:
  - "Wuauclt Network Connection"
  - "Exports Registry Key To an Alternate Data Stream"
  - "NetNTLM Downgrade Attack"
  - "Non Interactive PowerShell"
  - "Defense evasion via process reimaging"
# EventID and SystemTime are automatically added to the mapping schema and show in the table c
mappings:
  5156:
    title: "4. Security - local port forwarding to/from loopback IP to 3389 rdp port"
    provider: "Microsoft-Windows-Security-Auditing"
    search_fields:
      SourceAddress: "Event.EventData.SourceAddress"
      DestPort: "Event.EventData.DestPort"
    table_headers:
      context_field: "Event.EventData.SourceAddress"
      dest_port: "Event.EventData.DestPort"
```

sigma_rules > test_rules > evtx_xml.xml

```xml
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  - <System>
      <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3
      <EventID>5156</EventID>
      <Version>1</Version>
      <Level>0</Level>
      <Task>12810</Task>
      <Opcode>0</Opcode>
      <Keywords>0x8020000000000000</Keywords>
      <TimeCreated SystemTime="2019-02-13T18:04:45.9057832Z" />
      <EventRecordID>227741</EventRecordID>
      <Correlation />
      <Execution ProcessID="4" ThreadID="56" />
      <Channel>Security</Channel>
      <Computer>PC01.example.corp</Computer>
      <Security />
    </System>
  - <EventData>
      <Data Name="ProcessID">3324</Data>
      <Data Name="Application">\device\harddiskvolume1\users\user01\desktop\plink.exe</Data>
      <Data Name="Direction">%%14593</Data>
      <Data Name="SourceAddress">127.0.0.1</Data>
      <Data Name="SourcePort">49274</Data>
      <Data Name="DestAddress">127.0.0.2</Data>
      <Data Name="DestPort">3389</Data>
      <Data Name="Protocol">6</Data>
      <Data Name="FilterRTID">0</Data>
      <Data Name="LayerName">%%14611</Data>
      <Data Name="LayerRTID">48</Data>
      <Data Name="RemoteUserID">S-1-0-0</Data>
      <Data Name="RemoteMachineID">S-1-0-0</Data>
    </EventData>
</Event>
```

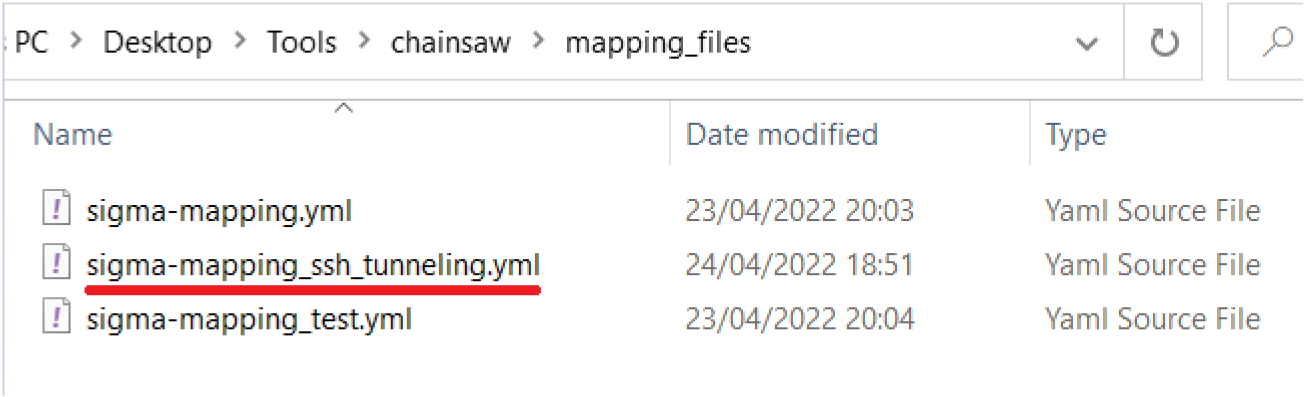
sigma_rules > test_rules > rdp_tunnelling - 4.yml > title

```yaml
title: 4. c2 RDP Tunneling
id: 00000000-2dbc-48b2-9096-000000000000
status: experimental
description: Detects RDP Tunneling via SSH
references:
    - Chainsaw sample evtx events
author: TNewman
date: 2022/04/24
logsource:
    product: windows
detection:
    selection:
        SourceAddress|contains: '127.0.0.1'
    selection2:
        DestPort|contains: '3389'
    condition: selection and selection2
falsepositives:
    - Unknown
level: medium
```

STEP 5 - Layout

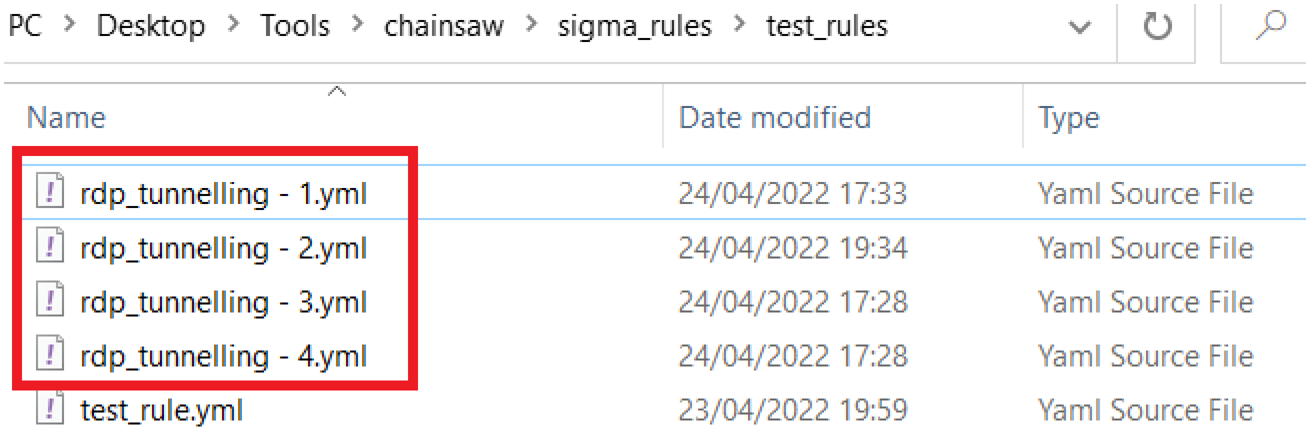
The mappings and rules were placed in the following paths:

<path_to>\chainsaw\mapping_files



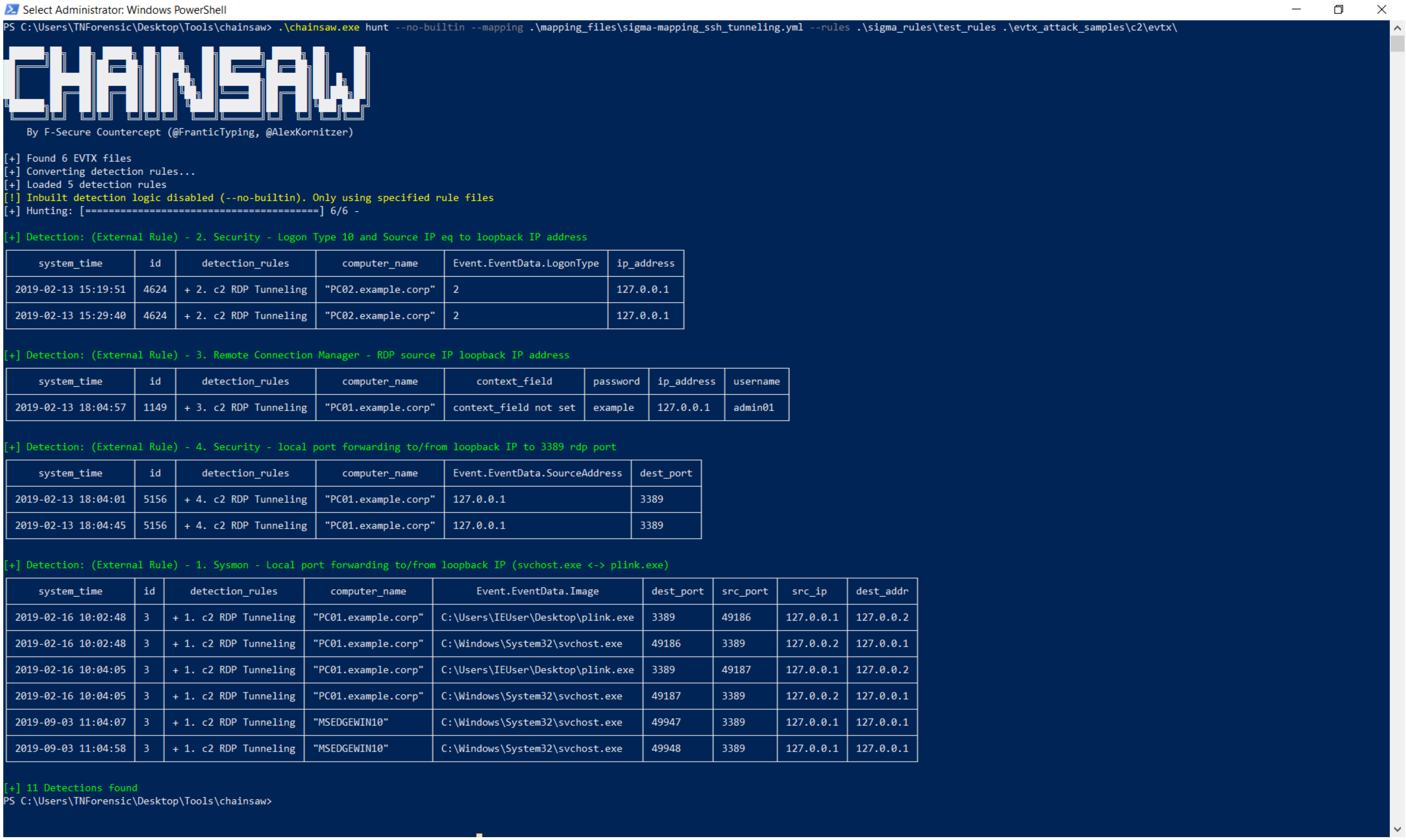
Name	Date modified	Type
sigma-mapping.yml	23/04/2022 20:03	Yaml Source File
sigma-mapping_ssh_tunneling.yml	24/04/2022 18:51	Yaml Source File
sigma-mapping_test.yml	23/04/2022 20:04	Yaml Source File

<path_to>\chainsaw\sigma-rules\test_rules



Name	Date modified	Type
rdp_tunnelling - 1.yml	24/04/2022 17:33	Yaml Source File
rdp_tunnelling - 2.yml	24/04/2022 19:34	Yaml Source File
rdp_tunnelling - 3.yml	24/04/2022 17:28	Yaml Source File
rdp_tunnelling - 4.yml	24/04/2022 17:28	Yaml Source File
test_rule.yml	23/04/2022 19:59	Yaml Source File

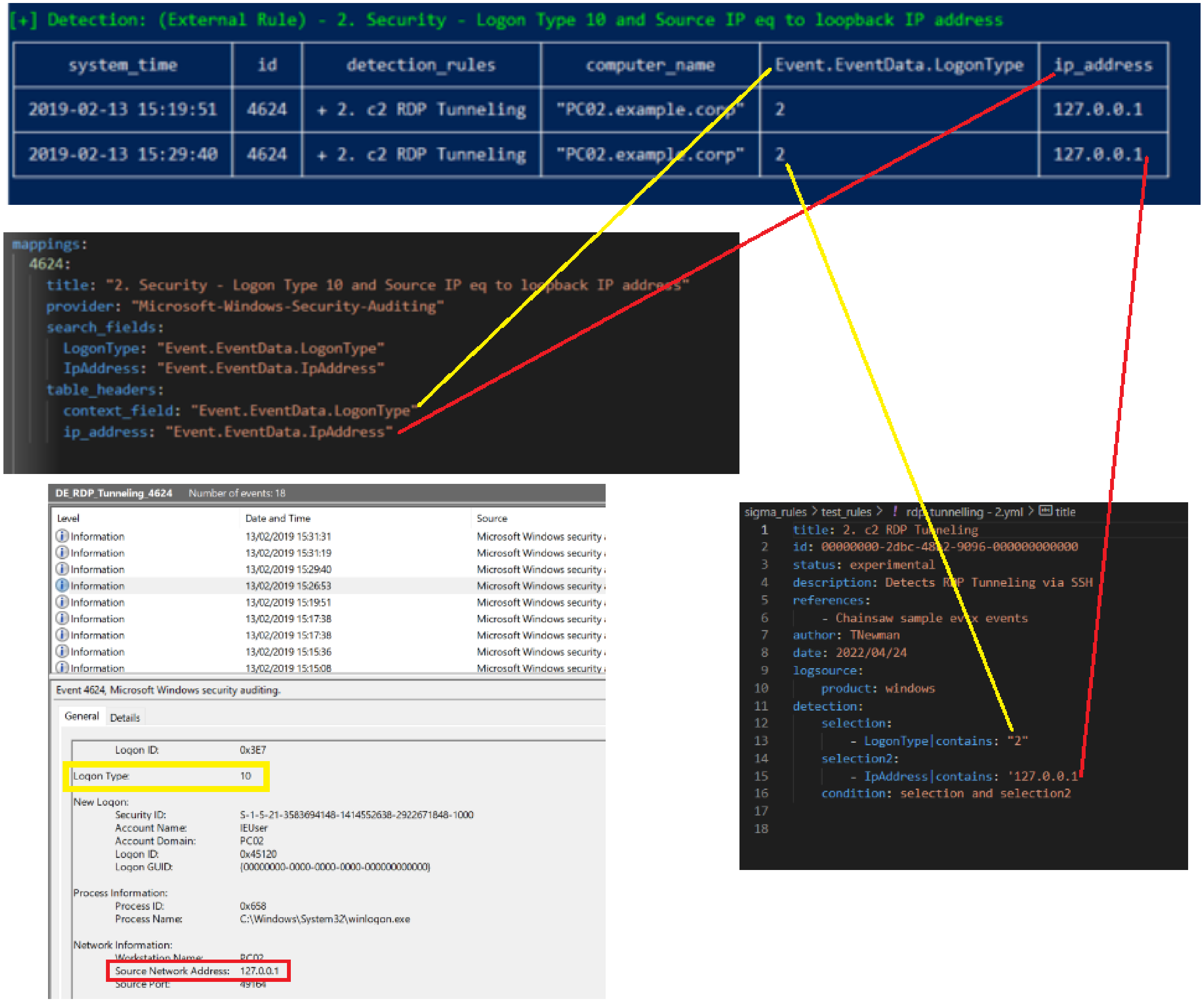
Test Drive



```
PS C:\Users\TNForensic\Desktop\Tools\chainsaw> .\chainsaw.exe hunt --no-builtin --mapping .\mapping_files\sigma-mapping_ssh_tunneling.yml --rules .\sigma_rules\test_rules .\evtx_attack_samples\c2\evtx\

CHAINSAW

    By F-Secure Countercept (@FranticTyping, @AlexKornitzer)

[+] Found 6 EVTX files
[+] Converting detection rules...
[+] Loaded 5 detection rules
[!] Inbuilt detection logic disabled (--no-builtin). Only using specified rule files
[+] Hunting: [========================================] 6/6 -

[+] Detection: (External Rule) - 2. Security - Logon Type 10 and Source IP eq to loopback IP address
┌─────────────────────┬──────┬───────────────────────┬────────────────────┬───────────────────────────┬────────────┐
│     system_time     │  id  │    detection_rules    │   computer_name    │ Event.EventData.LogonType │ ip_address │
├─────────────────────┼──────┼───────────────────────┼────────────────────┼───────────────────────────┼────────────┤
│ 2019-02-13 15:19:51 │ 4624 │ + 2. c2 RDP Tunneling │ "PC02.example.corp" │ 2                         │ 127.0.0.1  │
├─────────────────────┼──────┼───────────────────────┼────────────────────┼───────────────────────────┼────────────┤
│ 2019-02-13 15:29:40 │ 4624 │ + 2. c2 RDP Tunneling │ "PC02.example.corp" │ 2                         │ 127.0.0.1  │
└─────────────────────┴──────┴───────────────────────┴────────────────────┴───────────────────────────┴────────────┘

[+] Detection: (External Rule) - 3. Remote Connection Manager - RDP source IP loopback IP address
┌─────────────────────┬──────┬───────────────────────┬────────────────────┬───────────────────────┬──────────┬────────────┬──────────┐
│     system_time     │  id  │    detection_rules    │   computer_name    │     context_field     │ password │ ip_address │ username │
├─────────────────────┼──────┼───────────────────────┼────────────────────┼───────────────────────┼──────────┼────────────┼──────────┤
│ 2019-02-13 18:04:57 │ 1149 │ + 3. c2 RDP Tunneling │ "PC01.example.corp" │ context_field not set │ example  │ 127.0.0.1  │ admin01  │
└─────────────────────┴──────┴───────────────────────┴────────────────────┴───────────────────────┴──────────┴────────────┴──────────┘

[+] Detection: (External Rule) - 4. Security - local port forwarding to/from loopback IP to 3389 rdp port
┌─────────────────────┬──────┬───────────────────────┬────────────────────┬───────────────────────────────┬───────────┐
│     system_time     │  id  │    detection_rules    │   computer_name    │ Event.EventData.SourceAddress │ dest_port │
├─────────────────────┼──────┼───────────────────────┼────────────────────┼───────────────────────────────┼───────────┤
│ 2019-02-13 18:04:01 │ 5156 │ + 4. c2 RDP Tunneling │ "PC01.example.corp" │ 127.0.0.1                     │ 3389      │
├─────────────────────┼──────┼───────────────────────┼────────────────────┼───────────────────────────────┼───────────┤
│ 2019-02-13 18:04:45 │ 5156 │ + 4. c2 RDP Tunneling │ "PC01.example.corp" │ 127.0.0.1                     │ 3389      │
└─────────────────────┴──────┴───────────────────────┴────────────────────┴───────────────────────────────┴───────────┘

[+] Detection: (External Rule) - 1. Sysmon - Local port forwarding to/from loopback IP (svchost.exe <-> plink.exe)
┌─────────────────────┬────┬───────────────────────┬─────────────────────┬───────────────────────────────────┬───────────┬──────────┬───────────┬───────────┐
│     system_time     │ id │    detection_rules    │    computer_name    │       Event.EventData.Image       │ dest_port │ src_port │  src_ip   │ dest_addr │
├─────────────────────┼────┼───────────────────────┼─────────────────────┼───────────────────────────────────┼───────────┼──────────┼───────────┼───────────┤
│ 2019-02-16 10:02:48 │ 3  │ + 1. c2 RDP Tunneling │ "PC01.example.corp" │ C:\Users\IEUser\Desktop\plink.exe │ 3389      │ 49186    │ 127.0.0.1 │ 127.0.0.2 │
├─────────────────────┼────┼───────────────────────┼─────────────────────┼───────────────────────────────────┼───────────┼──────────┼───────────┼───────────┤
│ 2019-02-16 10:02:48 │ 3  │ + 1. c2 RDP Tunneling │ "PC01.example.corp" │ C:\Windows\System32\svchost.exe   │ 49186     │ 3389     │ 127.0.0.2 │ 127.0.0.1 │
├─────────────────────┼────┼───────────────────────┼─────────────────────┼───────────────────────────────────┼───────────┼──────────┼───────────┼───────────┤
│ 2019-02-16 10:04:05 │ 3  │ + 1. c2 RDP Tunneling │ "PC01.example.corp" │ C:\Users\IEUser\Desktop\plink.exe │ 3389      │ 49187    │ 127.0.0.1 │ 127.0.0.2 │
├─────────────────────┼────┼───────────────────────┼─────────────────────┼───────────────────────────────────┼───────────┼──────────┼───────────┼───────────┤
│ 2019-02-16 10:04:05 │ 3  │ + 1. c2 RDP Tunneling │ "PC01.example.corp" │ C:\Windows\System32\svchost.exe   │ 49187     │ 3389     │ 127.0.0.2 │ 127.0.0.1 │
├─────────────────────┼────┼───────────────────────┼─────────────────────┼───────────────────────────────────┼───────────┼──────────┼───────────┼───────────┤
│ 2019-09-03 11:04:07 │ 3  │ + 1. c2 RDP Tunneling │ "MSEDGEWIN10"       │ C:\Windows\System32\svchost.exe   │ 49947     │ 3389     │ 127.0.0.1 │ 127.0.0.1 │
├─────────────────────┼────┼───────────────────────┼─────────────────────┼───────────────────────────────────┼───────────┼──────────┼───────────┼───────────┤
│ 2019-09-03 11:04:58 │ 3  │ + 1. c2 RDP Tunneling │ "MSEDGEWIN10"       │ C:\Windows\System32\svchost.exe   │ 49948     │ 3389     │ 127.0.0.1 │ 127.0.0.1 │
└─────────────────────┴────┴───────────────────────┴─────────────────────┴───────────────────────────────────┴───────────┴──────────┴───────────┴───────────┘

[+] 11 Detections found
PS C:\Users\TNForensic\Desktop\Tools\chainsaw>
```

Perfect!

Looking at our Security.evtx Logon Type 10 results a bit closer…



```
[+] Detection: (External Rule) - 2. Security - Logon Type 10 and Source IP eq to loopback IP address
┌─────────────────────┬──────┬───────────────────────┬──────────────────────┬────────────────────────────┬────────────┐
│     system_time     │  id  │    detection_rules    │    computer_name     │ Event.EventData.LogonType  │ ip_address │
├─────────────────────┼──────┼───────────────────────┼──────────────────────┼────────────────────────────┼────────────┤
│ 2019-02-13 15:19:51 │ 4624 │ + 2. c2 RDP Tunneling │ "PC02.example.corp"  │ 2                          │ 127.0.0.1  │
├─────────────────────┼──────┼───────────────────────┼──────────────────────┼────────────────────────────┼────────────┤
│ 2019-02-13 15:29:40 │ 4624 │ + 2. c2 RDP Tunneling │ "PC02.example.corp"  │ 2                          │ 127.0.0.1  │
└─────────────────────┴──────┴───────────────────────┴──────────────────────┴────────────────────────────┴────────────┘
```

```
mappings:
  4624:
    title: "2. Security - Logon Type 10 and Source IP eq to loopback IP address"
    provider: "Microsoft-Windows-Security-Auditing"
    search_fields:
      LogonType: "Event.EventData.LogonType"
      IpAddress: "Event.EventData.IpAddress"
    table_headers:
      context_field: "Event.EventData.LogonType"
      ip_address: "Event.EventData.IpAddress"
```

DE_RDP_Tunneling_4624 Number of events: 18

Level	Date and Time	Source
Information	13/02/2019 15:31:31	Microsoft Windows security
Information	13/02/2019 15:31:19	Microsoft Windows security
Information	13/02/2019 15:29:40	Microsoft Windows security
Information	13/02/2019 15:26:53	Microsoft Windows security
Information	13/02/2019 15:19:51	Microsoft Windows security
Information	13/02/2019 15:17:38	Microsoft Windows security
Information	13/02/2019 15:17:38	Microsoft Windows security
Information	13/02/2019 15:15:36	Microsoft Windows security
Information	13/02/2019 15:15:08	Microsoft Windows security

Event 4624, Microsoft Windows security auditing.

```
        Logon ID:               0x3E7

Logon Type:                     10

New Logon:
        Security ID:            S-1-5-21-3583694148-1414552638-2922671848-1000
        Account Name:           IEUser
        Account Domain:         PC02
        Logon ID:               0x45120
        Logon GUID:             {00000000-0000-0000-0000-000000000000}

Process Information:
        Process ID:             0x658
        Process Name:           C:\Windows\System32\winlogon.exe

Network Information:
        Workstation Name:       PC02
        Source Network Address: 127.0.0.1
        Source Port:            49164
```

sigma_rules > test_rules > ! rdp_tunnelling - 2.yml > title

```
title: 2. c2 RDP Tunneling
id: 00000000-2dbc-48[illegible]2-9096-000000000000
status: experimental
description: Detects RDP Tunneling via SSH
references:
    - Chainsaw sample evtx events
author: TNewman
date: 2022/04/24
logsource:
    product: windows
detection:
    selection:
        - LogonType|contains: "2"
    selection2:
        - IpAddress|contains: '127.0.0.1
    condition: selection and selection2
```

Note - The system_time, id, detection_rules and computer_name are all parsed and displayed automatically by chainsaw
I also tested these rules using the –csv switch and they were all parsed across with no errors.

Conclusion

What is great about the functionality of chainsaw's mapping tool is that you can either:

1. Add to the native mappings - this could be used for event log types that you want extracted every time you run chainsaw (adding to their Suspicious Logons, Suspicious Process creation etc…)

2. Create an entirely separate mapping file bespoke to your investigation (as I have done with the RDP Tunneling) which could easily be shared and added to as an investigation progresses.

Being able to create individual mapping files with associated sigma rules will be a fantastic way for forensic examiners or threat intelligence analysts to build out packages for specific malware or threat actor TTPs/IOCs that are discovered in evtx logs and share with the community/industry as a whole.