Nidhogg

Installationsdokumentation

Ives Schneider

Index

1. Info

If you're planning to run nidhogg with it's scanning capability, you won't get around to sadly run it as root. (or with cap rights)
This is due the nmap scanning parameter and arp scanning mechanism.
Ideally you're using the already created docker, to mitigate some of the security concerns, not all though.

2. Requirements

Linux

- libpcap0.8
- nmap
- curl
- sqlite3-0

WARNING	If you're running Ubuntu chances are high that you'll need to create a symlink for libpcap `ln -s /usr/lib/x86_64-linux-gnu/libpcap.so.1.9.1 /usr/lib/x86_64-linux-gnu/libpcap.so.1`

3. Installation

From source

1. Clone repository or download source
2. Build nidhogg `cargo build --release`
3. Manually create config files (see example config.yml)
4. Copy service file to /etc/systemd/system
5. Activate Unit `systemctl enable nidhogg`

Binary (Ubuntu)

1. Download latest .deb
2. Install with dpkg `dpkg -i xx.deb`
3. Configure application (/etc/nidhogg/)
4. Enable unit `systemd enable nidhogg`

Binary (Windows)

Get the Docker image from:
https://hub.docker.com/r/b401/nidhogg

Docker

https://hub.docker.com/r/b401/nidhogg

```
docker run -it --net=host --privileged -v config.yml:/etc/nidhogg/config.yml -v
mappings.xml:/etc/nidhogg/mappings.xml -v portspecs.yml:/etc/nidhogg/portspecs.yml
b401/nidhogg:final
```

4. Configuration

WARNING	All configflags are mandatory

See examples for more indepth settings.

4.1. Main configuration file

Defines most configuration aspects of nidhogg.
All config flags are mandatory but every functionality can be disabled.

config.yml

```
# Choose on which address and port the webserver should listen
webserver:
    enable: true
    username: "admin"
    password: "admin"
    address: "0.0.0.0"
    port: "8080"

# Enable or disable mail notifications
mail:
    enable: true
    server: "smtp.gmail.com"
    username: "user@gmail.com"
    password: ""
    email: "user@i-401.xyz"
    from: "user@gmail.com"

# Set address and login to remote Splunk endpoint (multiple endpoints are possible)
splunk:
    enable: true
    server: "splunk:8089"
    username: ""
    password: ""
    interval: 500

# Set remote snmp server address and community + oids.
# Multiple oids and servers are possible
snmp:
    enable: true
    server: "127.0.0.1"
    community: "my_comm"
    oid: "1.3.6.100.1.2.3.5.1.1.0"
```

```
# Enable/disable portscanning (requires root rights)
portscan:
    enable: true
    portspec: "/etc/nidhogg/portspecs.yml"
    mappings: "/etc/nidhogg/mappings.xml"
    timeout: 500

# Enable/disable arpscanning (requires root rights)
# Use mac list to whitelist devices. (You won't get notifications if those devices getting
connected)
arpscan:
    enable: true
    interface: "eth0"
    db: "/etc/nidhogg/arp.db"
    timeout: 500
    mac:
      - "00:17:88:28:9f:ca"
      - "00:55:da:50:40:64"
      - "34:7e:5c:31:10:e8"
      - "c8:3c:85:3e:e8:dd"
      - "f4:4d:30:68:9b:d4"
```

4.2. Portscan configuration files

Defines which target and which ports should be in a special state.
If a port is undefined, it will be ignored in the final report.

mappings.xml is used to bind a spec to a target.

Special thanks to nmap-analyze

portspecs.yml

```
portspecs:
  - name: artoria
    ports:
      - id: 22
        state: open
      - id: 25
        state: closed
```

mappings.xml

```
{ "mappings":
    [
        {
            "hostname": "artoria",
            "id": "i-0",
            "ips": ["10.2.1.121"],
            "name": "artoria",
            "portspec": "artoria"
        }
    ]
}
```

5. Uninstall

Uninstalling is as easy as installing.
If you've installed nidhogg via .deb, just remove the deb with apt.

Ubuntu / apt based
`apt remove nidhogg`

Compiled from source / Binary
`rm -r /etc/nidhogg && rm /usr/bin/nidhogg`