### Incident Metadata

**Case Type:** Network Service Discovery (*Port Scan*)

**Reported by:** Splunk SIEM

Analyst: Haydar AKYÜREK

Date:2025-10-08Severity:HighStatus:✓ Closed

**Decision: ✓ True Positive – Non Issue** 

## **Detection Rule Details**

**Rule Name:** Network Service Discovery

**Rule Description:** An adversary uses a combination of techniques to determine the state of the ports on a remote target. Any service or application available for TCP or UDP networking will have a port open for communications over the network. Although common services have assigned port numbers, services and applications can run on arbitrary ports. Additionally, port scanning is complicated by the potential for any machine to have up to 65535 possible UDP or TCP services.

### **Correlation Rule (SPL):**

| tstats dc(All\_Traffic.dest\_port) as scannedPortCount from datamodel=Network\_Traffic.All\_Traffic by \_time span=10m, All\_Traffic.src\_ip | rename All\_Traffic.\* as \* | where scannedPortCount > 50

# Splunk Incident Analysis Notes (Port Scanning Case)

- 1. Took ownership of the incident by clicking **Edit Selected**.
- 2. The **severity level** was assigned by the engineering team can be adjusted if needed.
- 3. Confirmed that the notable falls under the **Network domain**, meaning it's a network-based detection.
- 4. Reviewed the corresponding **Correlation Search** rule to understand the trigger conditions and field mappings.
- 5. Queried the scanning IP (202.181.188.74) in **AbuseIPDB** and **VirusTotal** to assess its reputation and any previous malicious reports.
- 6. After conducting **OSINT**, analyzed our internal network activity using Fortinet logs.
  - Found connections to over 50 different destination ports (dest\_port), indicating broad probing behavior.
  - Traffic included several **non-standard** ports.

(Splunk query)

```
index=fortinet srcip=202.181.188.74
```

(*Note:* dc = distinct count, counts unique values.)

- 7. Checked outbound traffic patterns most connections were on ports **443** and **80**, typical of scanning via a web-based service.
  - Verified using dnslytics.com/reverse-ip.

```
(Splunk query)
```

```
index=fortinet srcip=202.181.188.74
| stats dc(dstport) values(dstport) by _time dstport
```

8. Determined how many times each port was targeted and checked for repeated access to the same ports.

(Splunk query)

```
index=fortinet dstport="117" action="block*" src_ip!="10.1.1.0/8"
| table _time src_ip dest_ip
```

- 9. Verified which of our internal IPs were scanned by the source.
  - Used dedup and stats count to summarize destination ports.

(Splunk query)

```
index=fortinet srcip=202.181.188.74
| table dest_port
| dedup dest_port
or
| stats count by dest_port
```

10. Checked for any **successful traffic** allowed from the same source.

(Splunk query)

```
index=fortinet srcip=202.181.188.74 action=allow*
```

11. Looked for any **return (outbound) traffic** from our side toward that IP.

(Splunk query)

```
index=fortinet dest=202.181.188.74
```

12. Checked if the source IP visited a high number of destination ports (external IPs only).

(Splunk query)

```
index=fortinet src!=202.* sourcetype=fortigate_traffic
| stats dc(dest_port) as port_count by src
| where port_count > 50
```

13. Determined whether the same source scanned multiple internal IPs.

### (Splunk query)

```
index=fortinet src!=202.* sourcetype=fortigate_traffic
| stats dc(dest) as ip_count values(dest) by src
| where ip_count > 3
```

- 14. Reviewed **historical traffic** (1 day, 1 week, 30 days).
  - Found entries with ftnt\_action=deny and action=blocked, confirming
     Fortinet firewall blocked this IP.
- 15. Verified if there was any **outbound communication** toward **202.181.188.74**.
  - Used **Security Domains** → **Network** → **Traffic Search**, right-click → "Open in search" to compare patterns.
- 16. Confirmed the IP did **not belong to our infrastructure**.
  - Only external IP lookups were observed, no domain enumeration consistent with scanning activity.
- 17. Marked the incident as **True Positive Non-Issue**, as malicious traffic was successfully blocked.
  - "All malicious traffic (port scanning) from IP (202.181.188.74) was blocked within 30 days by Fortinet firewall."
    - Added **Hive ticket reference** if applicable.
- 18. Checked **IPS logs** no triggers detected.
  - If a signature had matched, the IPS would have auto-tagged the event.

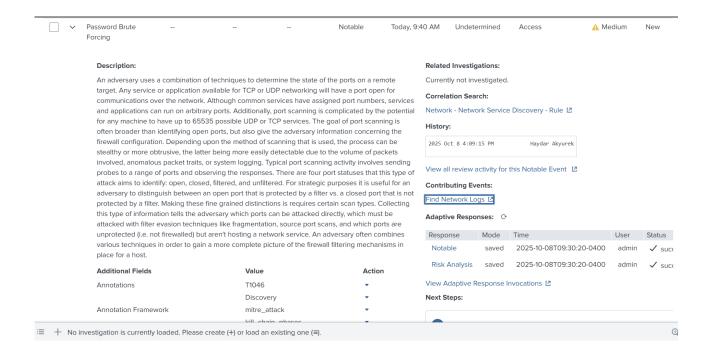
# **Decision: True Positive** (unauthorized port scanning attempt).

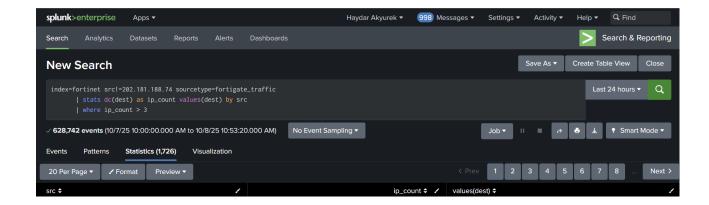
# MITRE ATT&CK Mapping

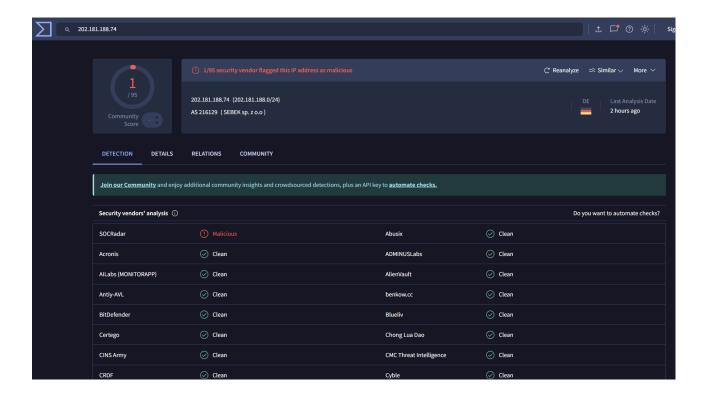
• T1046 – Network Service Discovery

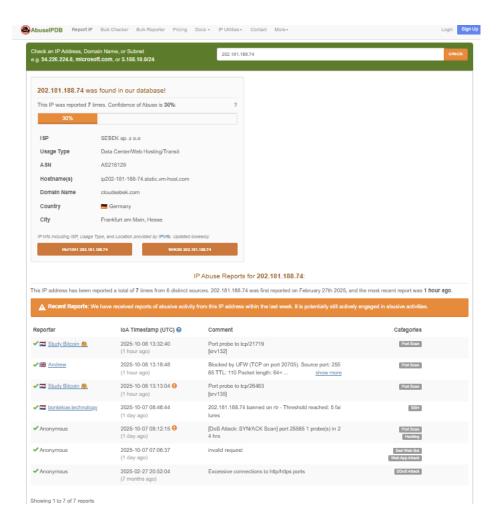
### **Recommended Actions (SOC Level)**

- Block ip address at perimeter firewall.
- Review firewall logs for additional attempts from the same ASN.
- Conduct threat hunting for related activity in EDR/SIEM (look for lateral movement attempts on the scanned hosts).
- Update detection logic to enrich alerts with ASN/WHOIS reputation.











TECHNIQUES

Software Discovery System Information Discovery System Location Discovery System Network Configuration Discovery System Network Connections Discovery System Owner/User Discovery System Service Discovery System Time Discovery Virtual Machine Discovery Virtualization/Sandbox ~ Lateral Movement Collection Command and Control Exfiltration

Impact

Home > Techniques > Enterprise > Network Service Discovery

### Network Service Discovery

Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation. Common methods to acquire this information include port, vulnerability, and/or wordlist scans using tools that are brought onto a system.<sup>[11]</sup>

Within cloud environments, adversaries may attempt to discover services running on other cloud hosts.

Additionally, if the cloud environment is connected to a on-premises environment, adversaries may be able to identify services running on non-cloud systems as well.

Within macOS environments, adversaries may use the native Bonjour application to discover services running on other macOS hosts within a network. The Bonjour mDNSResponder daemon automatically registers and advertises a host's registered services on the network. For example, adversaries can use a mDNS query (such as <a href="mailto:dns-set-b-">dns-set-b-</a> <a href="mailto:dns-set-b-">\_set-b-</a>. <a href="mailto:dns-set-b-">Jos find other systems broadcasting the ssh service</a> <a href="mailto:lns-set-b-">[28]</a>

ID: T1046

Sub-techniques: No sub-techniques

① Tactic: Discovery

 Platforms: Containers, IaaS, Linux, Network Devices, Windows, macOS

Contributors: Aaron Sullivan aka ZerkerEOD; Praetorian

Version: 3.2 Created: 31 May 2017 Last Modified: 15 April 2025

Version Permalink

#### Procedure Examples

ID	Name	Description
G1030	Agrius	Agrius used the open-source port scanner WinEggDzop to perform detailed scans of hosts of interest in victim networks. <sup>[4]</sup>
G0050	APT32	APT32 performed network scanning on the network to search for open ports, services, OS finger-printing, and other vulnerabilities.
G0087	APT39	$APT39\ has\ used\ CrackMapExec\ and\ a\ custom\ port\ scanner\ known\ as\ BLUETORCH\ for\ network\ scanning.^{[6][7]}$