

## Incident Metadata

Case Type:	Network Service Discovery ( <i>Port Scan</i> )
Reported by:	Splunk SIEM
Analyst:	Haydar AKYÜREK
Date:	2025-10-08
Severity:	<span style="color: red;">●</span> <b>High</b>
Status:	<span style="color: green;">✓</span> Closed
Decision:	<span style="color: green;">✓</span> <b>True Positive – Non Issue</b>

## 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](https://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.

**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. The goal of port scanning is often broader than identifying open ports, but also give the adversary information concerning the firewall configuration. Depending upon the method of scanning that is used, the process can be stealthy or more obtrusive, the latter being more easily detectable due to the volume of packets involved, anomalous packet traits, or system logging. Typical port scanning activity involves sending probes to a range of ports and observing the responses. There are four port statuses that this type of attack aims to identify: open, closed, filtered, and unfiltered. For strategic purposes it is useful for an adversary to distinguish between an open port that is protected by a filter vs. a closed port that is not protected by a filter. Making these fine grained distinctions requires certain scan types. Collecting this type of information tells the adversary which ports can be attacked directly, which must be attacked with filter evasion techniques like fragmentation, source port scans, and which ports are unprotected (i.e. not firewalled) but aren't hosting a network service. An adversary often combines various techniques in order to gain a more complete picture of the firewall filtering mechanisms in place for a host.

Additional Fields	Value	Action
Annotations	T1046	
	Discovery	
Annotation Framework	mitre_attack	
	kill_chain_phases	

+ No investigation is currently loaded. Please create (+) or load an existing one (≡).

**Related Investigations:**

Currently not investigated.

**Correlation Search:**

[Network - Network Service Discovery - Rule](#) 

### History:

2025 Oct 8 4:09:15 PM Haydar Akyurek

[View all review activity for this Notable Event](#) 

**Contributing Events:**

[Find Network Logs](#) Adaptive Responses: 

Response	Mode	Time	User	Status
Notable	saved	2025-10-08T09:30:20-0400	admin	✓ succ
Risk Analysis	saved	2025-10-08T09:30:20-0400	admin	✓ succ

[View Adaptive Response Invocations](#) 

**Next Steps:**

splunk>enterprise

Apps ▾

Haydar Akyurek ▾

998 Messages ▾

Settings ▾

Activity ▾

Help ▾

Find

Search

Analytics

Datasets

Reports

Alerts

Dashboards

Search & Reporting

New Search

Save As ▾ Create Table View Close

```
index=fortinet src!=202.181.188.74 sourcetype=fortigate_traffic
| stats dc(dest) as ip_count values(dest) by src
| where ip_count > 3
```

Last 24 hours ▾ 🔍

✓ 628,742 events (10/7/25 10:00:00.000 AM to 10/8/25 10:53:20.000 AM)

No Event Sampling ▾

Job ▾ || ■ ↺ 🔄 ⬇ ⚡ Smart Mode ▾

Events Patterns Statistics (1,726) Visualization

20 Per Page ▾ ↗ Format Preview ▾

< Prev

1

2

3

4

5

6

7

8

...

Next >

src ▾ ↗

ip\_count ▾ ↗

values(dest) ▾ ↗

202.181.188.74

1/95

Community Score

1/95 security vendor flagged this IP address as malicious

ReanalyzeSimilarMore

202.181.188.74 (202.181.188.0/24)  
AS 216129 (SEBEK sp. z o.o.)  
DE  
Last Analysis Date  
2 hours ago

DETECTIONDETAILSRELATIONSCOMMUNITY

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysisDo you want to automate checks?

SOCRadar	Malicious	Abusix	Clean
Acronis	Clean	ADMINUSLabs	Clean
AILabs (MONITORAPP)	Clean	AlienVault	Clean
Antiy-AVL	Clean	benkow.cc	Clean
BitDefender	Clean	Blueliv	Clean
Certego	Clean	Chong Lua Dao	Clean
CINS Army	Clean	CMC Threat Intelligence	Clean
CRDF	Clean	Cyble	Clean

AbuseIPDBReport IPBulk CheckerBulk ReporterPricingDocsIP UtilitiesContactMoreLoginSign Up

Check an IP Address, Domain Name, or Subnet  
e.g. 54.226.224.8, microsoft.com, or 5.188.10.0/24

202.181.188.74

CHECK

202.181.188.74 was found in our database!

This IP was reported 7 times. Confidence of Abuse is 30%.

30%

ISP

SEBEK sp. z o.o

Usage Type

Data Center/Web Hosting/Transit

ASN

AS216129

Hostname(s)

ip202-181-188-74.static.vm-host.com

Domain Name

cloudsebek.com

Country

Germany

City

Frankfurt am Main, Hesse

IP Info including ISP, Usage Type, and Location provided by IPinfo. Updated biweekly.

IPinfo 202.181.188.74IPinfo 202.181.188.74

IP Abuse Reports for 202.181.188.74:

This IP address has been reported a total of 7 times from 6 distinct sources. 202.181.188.74 was first reported on February 27th 2025, and the most recent report was 1 hour ago.

Recent Reports: We have received reports of abusive activity from this IP address within the last week. It is potentially still actively engaged in abusive activities.

Reporter	IoA Timestamp (UTC)	Comment	Categories
Study Bitcoin	2025-10-08 13:32:40 (1 hour ago)	Port probe to tcp/21719 [srv132]	Port Scan
Andrew	2025-10-08 13:18:48 (1 hour ago)	Blocked by UFW (TCP on port 20705). Source port: 255 65 TTL: 110 Packet length: 64< ... <a href="#">show more</a>	Port Scan
Study Bitcoin	2025-10-08 13:13:04 (1 hour ago)	Port probe to tcp/26483 [srv135]	Port Scan
hontekoe.technology	2025-10-07 08:46:44 (1 day ago)	202.181.188.74 banned on rtr - Threshold reached: 5 fai lures	SSH
Anonymous	2025-10-07 08:12:15 (1 day ago)	[DoS Attack: SYN/ACK Scan] port 25565 1 probe(s) in 2 4 hrs	Port Scan Flooding
Anonymous	2025-10-07 07:06:37 (1 day ago)	invalid request	Bad Web Bot Web App Attack
Anonymous	2025-02-27 20:52:04 (7 months ago)	Excessive connections to http/https ports	DDoS Attack

Showing 1 to 7 of 7 reports

- 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 Evasion ▾

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>[1]</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 `dns-sd -B _ssh._tcp .`) to find other systems broadcasting the ssh service.<sup>[2][3]</sup>

ID: T1046

Sub-techniques: No sub-techniques

① **Tactic:** [Discovery](#)

① **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 <a href="#">WinEggdrop</a> 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. <sup>[5]</sup>
G0087	APT39	APT39 has used CrackMapExec and a custom port scanner known as BLUETORCH for network scanning. <sup>[6][7]</sup>