📁 Case Metadata

Field	Value
Case ID:	CS-2025-001
Case Type:	Network Service Discovery (Port Scan)
Reported by:	Splunk SIEM
Analyst:	Haydar AKYÜREK
Date:	2025-10-08
Severity:	High
Status:	Closed
Environment:	Production

📜 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. 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 is 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.

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
```

Triggered Field(s):
srcip, dstip, dstport, _time
(Alarm was triggered because the same srcip attempted connections to more than 50 distinct destination ports within a 10-minute window.)

📌 **Detection Summary**
An adversary may attempt to identify running services on remote systems by scanning multiple ports across one or more hosts. This behavior often precedes lateral movement or exploitation. The detected activity matches known scanning patterns, where a single source IP rapidly probed multiple destination hosts and ports within a short time window.

🔍 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.

🔍 Additional Key Points

- Extending **time range** (up to 30 days) helps identify recurring or distributed scanning.
- Reviewing **ASN / ISP data** for `202.181.188.74` (via AbuseIPDB) can indicate if the IP belongs to a known hosting or proxy network.
- Combining

  ```
  | dedup dest_port | stats count by dest_port, dest_ip
  ```

  is useful to visualize scanning behavior.
- Always verify **Correlation Search throttling** if detections stop triggering unexpectedly

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

Investigation Steps:
- Queried Splunk for similar events within the last 24 hours.
- Checked IOC reputation (domain/IP) in VirusTotal and abuseipdb.com

🧠 MITRE ATT&CK Mapping
- T1046 – Network Service Discovery

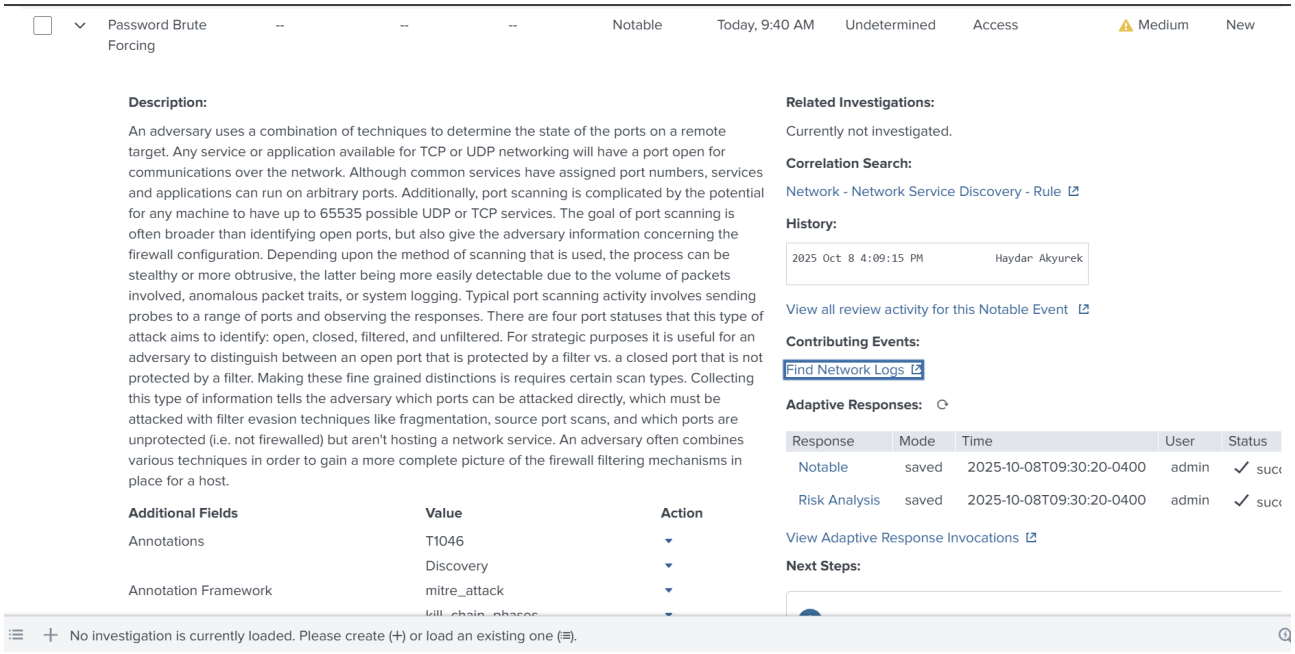
🔧 Recommended Actions (SOC Level)
- Block ip address at perimeter firewall.
- Review firewall logs for additional attempts from the same ASN.
- Conduct threat hunting for related activity in EDR/SIEM (look for lateral movement attempts on the scanned hosts).
- Update detection logic to enrich alerts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

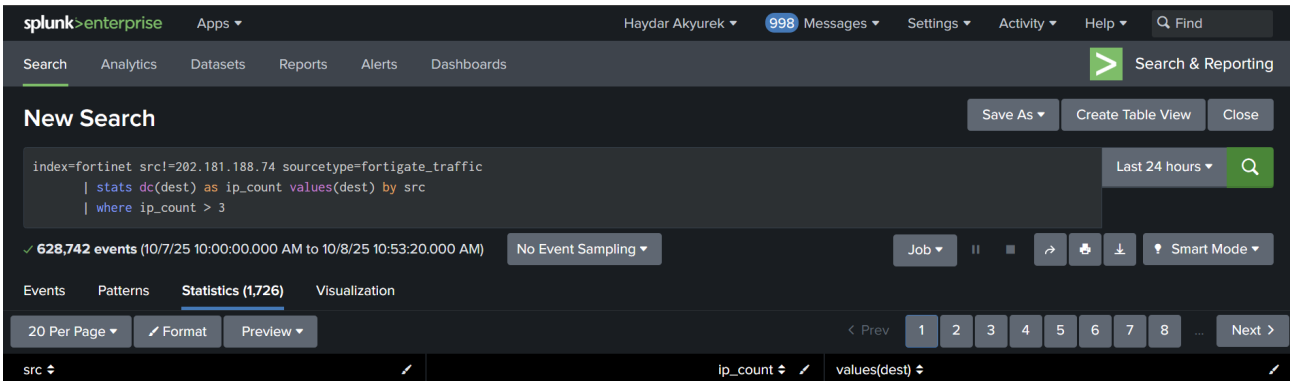
🏢 Organizational / HR Actions
- Notify network team and service owners of affected hosts (96.73.98.x subnet).
- Document incident in the central tracker and link to whitelist/exceptions list if legitimate scanner is identified later.

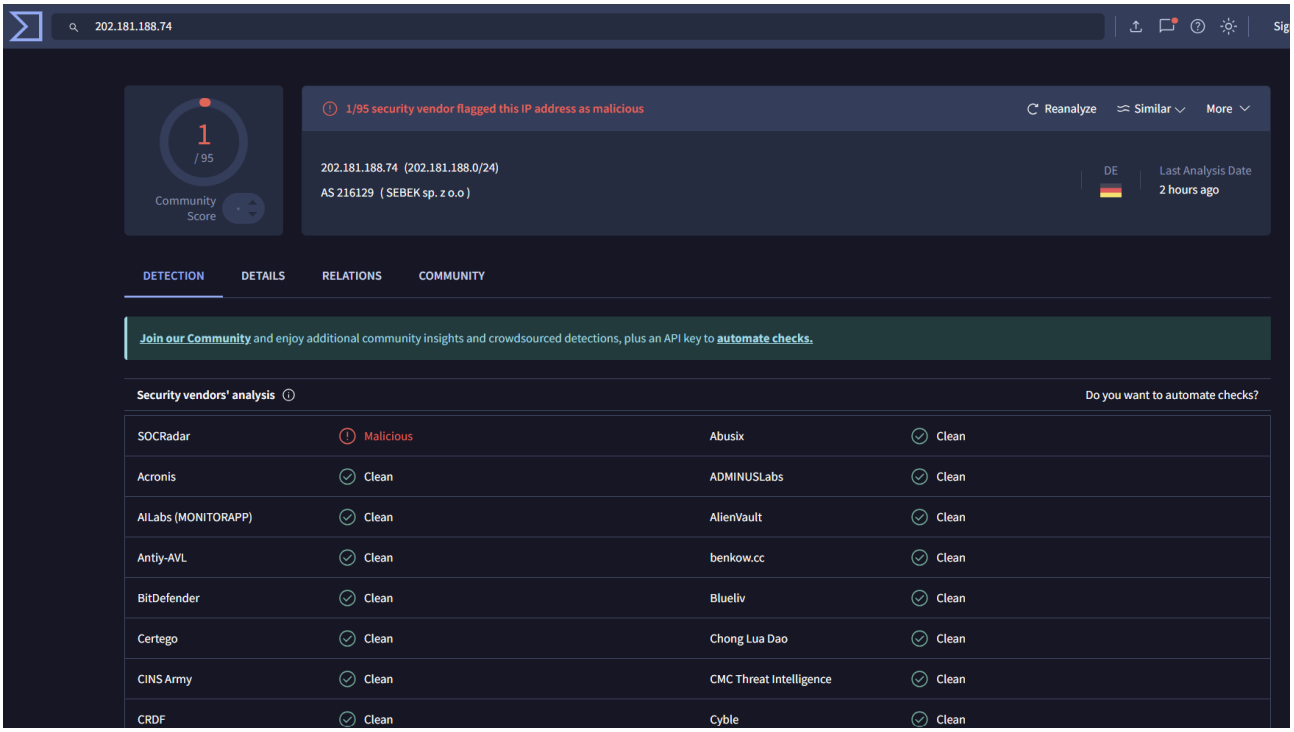
📊 Business Impact
- Unauthorized reconnaissance against production assets increases the risk of exploitation of exposed services (RDP, SMB, SQL, etc.).
- If scanning is successful, adversary may discover vulnerable services, leading to lateral movement or data exfiltration.

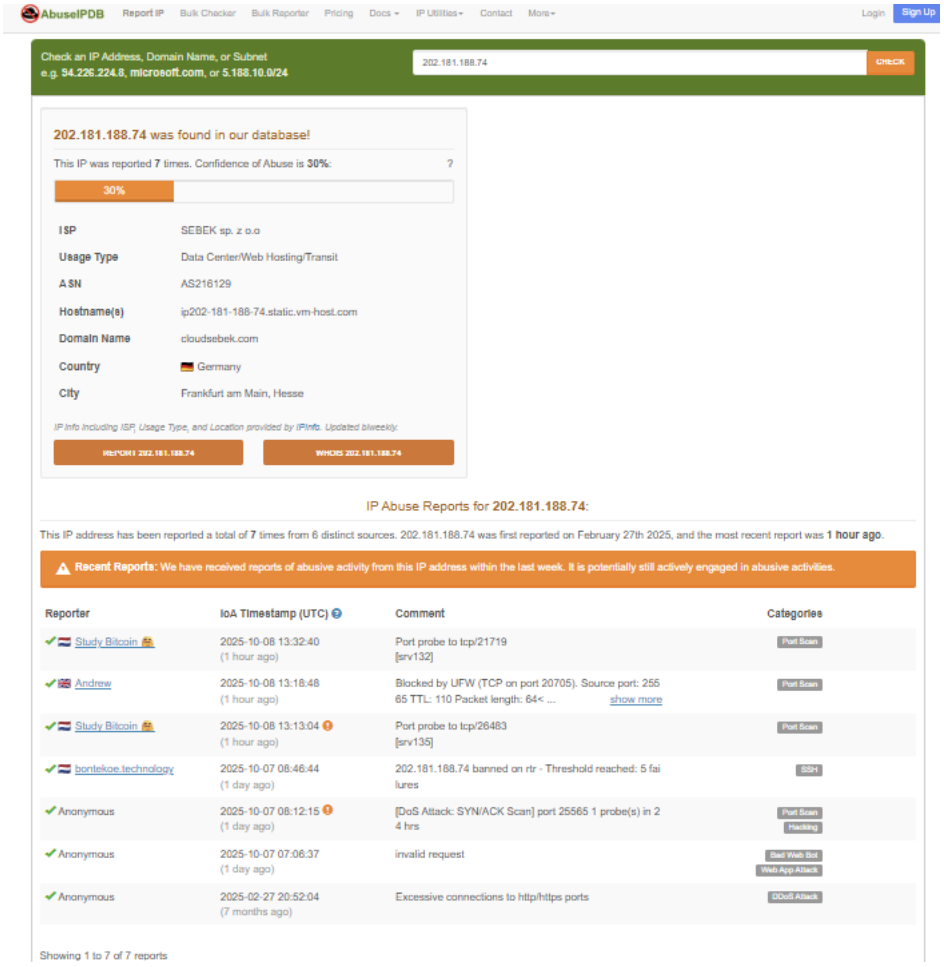
📁 Evidence in Case
- Splunk search query results and screenshot (showing srcip, dstip, dstport, and event counts)
- Firewall / Network traffic logs indicating multiple connection attempts from source 202.181.188.74
- DNS query logs (to validate whether scanning activity also attempted name resolution)

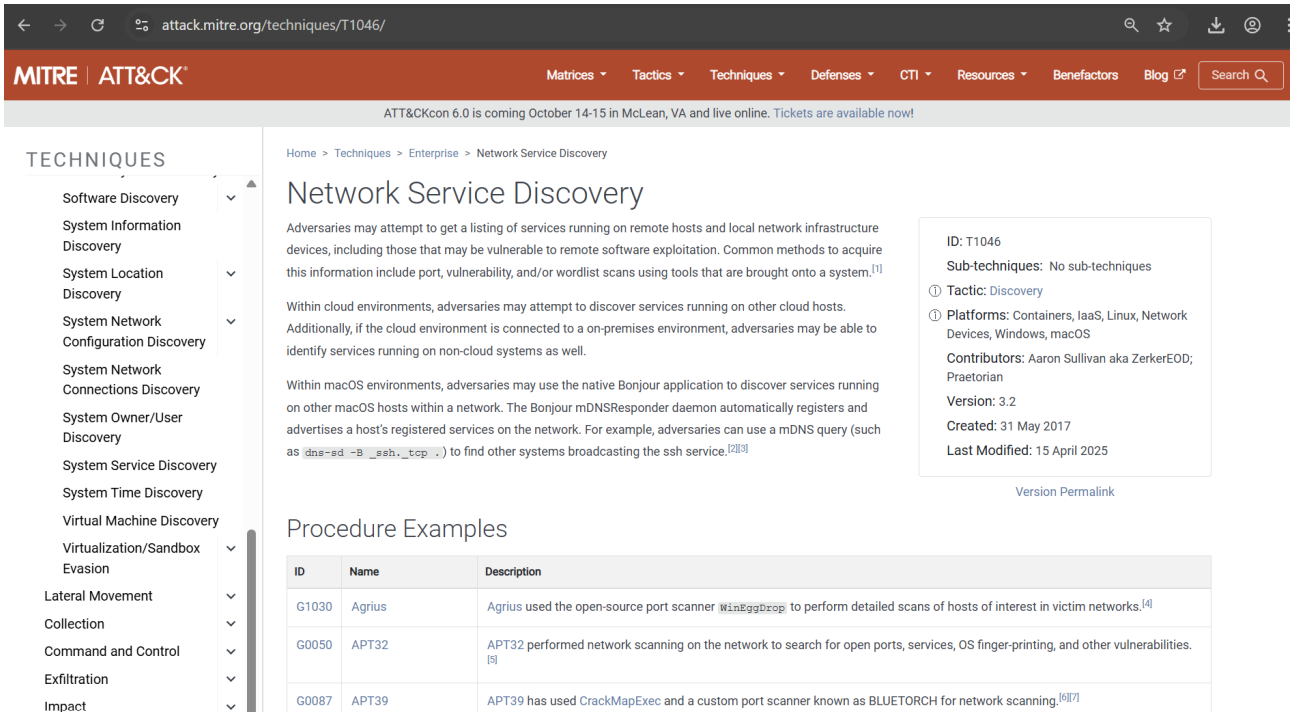
- Proxy traffic logs (to confirm if outbound traffic patterns deviate from baseline)
- IOC reputation check results (VirusTotal, AbuseIPDB, Talos lookups for 202.181.188.74)
- MITRE ATT&CK Navigator mapping highlighting technique **T1046 – Network Service Discovery**






202.181.188.74
1
/ 95
Community Score
1/95 security vendor flagged this IP address as malicious
Reanalyze
Similar
More
202.181.188.74 (202.181.188.0/24)
AS 216129 (SEBEK sp. z o.o)
DE
Last Analysis Date
2 hours ago
DETECTION
DETAILS
RELATIONS
COMMUNITY
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.
Security vendors' analysis
Do 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


AbuseIPDB
Report IP
Bulk Checker
Bulk Reporter
Pricing
Docs
IP Utilities
Contact
More
Login
Sign Up
Check an IP Address, Domain Name, or Subnet
e.g. 94.226.224.8, 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.
REPORT 202.181.188.74
WHOIS 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< ... show more
Port Scan
Study Bitcoin
2025-10-08 13:13:04
(1 hour ago)
Port probe to tcp/26483
[srv135]
Port Scan
bontekoe.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
Hacking
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


attack.mitre.org/techniques/T1046/
MITRE | ATT&CK
Matrices
Tactics
Techniques
Defenses
CTI
Resources
Benefactors
Blog
Search
ATT&CKcon 6.0 is coming October 14-15 in McLean, VA and live online. Tickets are available now!
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.[1]
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.[2][3]
ID: T1046
Sub-techniques: No sub-techniques
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 WinEggDrop to perform detailed scans of hosts of interest in victim networks.[4]
G0050
APT32
APT32 performed network scanning on the network to search for open ports, services, OS finger-printing, and other vulnerabilities.[5]
G0087
APT39
APT39 has used CrackMapExec and a custom port scanner known as BLUETORCH for network scanning.[6][7]