Martin Hagmann

Lab: Maintaining Access - Step 2 - Host & Service Discovery

Teamwork - Team DaMa:

- Daniel Müller
- Martin Hagmann

Description

We are particularly interested in which hosts are available, what services they offer and additional information we can collect about potential targets (like OS types, versions, etc.).

Questions to Answer

Q1:
How many systems did you identify?

A1:
The *host discovery* with the specific nampa command provides us *eight* active hosts.

```
nmap -n -sn 10.0.1.0/24 -oA host_discovery --min-rate=20000`
```

```
# cat hosts.txt
10.0.1.1
10.0.1.10
10.0.1.15
10.0.1.100
10.0.1.101
10.0.1.102
10.0.1.103
10.0.1.254

# cat hosts.txt | wc -l
8
```

Q2:
What seems to be the purpose of the individual hosts (as far as you can tell from the services they offer)?

A2:
Based on the *Service Discovery Scan* we can identify the following most interesting systems/services (details file: *script_version_scan.gnmap*):

- Host: 10.0.1.10 / Windows System / Microsoft Terminal Services
- Host: 10.0.1.100 / Windows DC (Domain: winattacklab.local0) / AD DC services
- Host: 10.0.1.102 / Windows System / NetBios, Microsoft Terminal Services
- Host: 10.0.1.103 / Windows Web Server / Microsoft IIS httpd 10.0

Q3:
Did the scan already reveal vulnerabilities/weaknesses?

A3 (updated):
The performed nmap scans creates three output files. Unfortunately not all three files contains the same output. the `XML` file provide more details as the gnmap and nmap files. The `XML` file contains now the following potentional vulnerability/weakness on the Microsoft IIS Server (10.0.1.103).

Part of scan result from host 10.0.1.103:

```
<port protocol="tcp" portid="80"><state state="open" reason="syn-ack"
reason_ttl="128"/><service name="http" product="Microsoft IIS httpd"
version="10.0" ostype="Windows" method="probed" conf="10">
<cpe>cpe:/a:microsoft:iis:10.0</cpe><cpe>cpe:/o:microsoft:windows</cpe>
</service><script id="http-methods" output="&#xa;  Potentially risky
methods: TRACE"><table key="Potentially risky methods">
<elem>TRACE</elem>
</table>
</script><script id="http-server-header" output="Microsoft-IIS/10.0">
<elem>Microsoft-IIS/10.0</elem>
</script><script id="http-title" output="Happy Pony"><elem
key="title">Happy Pony</elem>
</script></port>
```

The system has the `TRACE` method enabled. The HTTP TRACE can be used for Cross Site Tracing (XST) attacks. It can also be combined with cross-domain browser vulnerabilities to read sensitive header information from third-party domains.

References:

- https://www.arridae.com/blogs/trace-method.php

A3 - first answer provided on 08.12.20202
The scan details don't present vulnerabilities/weaknesses information. But based on the scan report we can now go further and look for windos vulnerabilities based on the version information (e.g. Microsoft IIS httpd 10.0).

Q4:
Explain the nmap options that make a difference between the first and the second scan

A4:
First scan (host discovery):

```
nmap -n -sn 10.0.1.0/24 -oA host_discovery --min-rate=20000
```

This scan performs a ping scan without a DNS lookup and writes the output in the three different file formats (xml, nmap, gnmap).

`-n` no DNS resolution
`-sn` Ping Scan - disable port scan
`-oA` Output in all supported formats (xml, nmap, gnmap)
`--min-rate` Send packets no slower than per second

Second scan (service discovery):

```
nmap -n -sC -sV -iL hosts.txt -oA script_version_scan --min-rate=20000
```

With this scan a service discovery scan is performed with the standard NSE scripts. Furthermore a version discovery for the open ports are executed. The scan is only performed on the hosts which are provided with the input file.

`-n` no DNS resolution
`-sC` Performs a script scan using the default set of scripts. It is equivalent to --script=default.
`-sV` Probe open ports to determine service/version info
`-iL` Reads target specifications from input filename.
`-oA` Output in all supported formats (xml, nmap, gnmap)
`--min-rate` Send packets no slower than per second