PCAP1

Statistics

- 219K packets
- Protocols: `smb`, `rpc`, `ssh`, `mysql`, `ldaps`, `http`, `ftp`

Anomalies

Nmap scanning - Overview

- Overview (this)
- Protocol Hierarchy
- Scanning Activity
  - there is evidence of network scanning by `10.16.80.243`
- Specifications
  - Src: `10.16.80.243`
  - Dst: `10.168.27.1/24`
- Implications
  - For the most part there implications are only that someone was able to scan the network. There doesn't seem to have been any successful access attempts. Though a few misconfigurations/vulnerabilities do exist.
- Solutions
  1. Disable anonymous FTP login
  2. Disable HTTP (unencrypted)
  3. Remove DVWA
  4. Convert to FTPS or SFTP, and disable FTP
  5. Setup a firewall & firewall rules
  6. Segment Network Subnets
  7. Use unique SSH keys for every service
- Sources
  - See below

Protocol Hierarchy

!prot-hier * SSH * Duplicate SSH Keys (two services `10.168.27.14`) * Duplicate SSH Keys (two services `10.168.27.132`) * Sources: 1, 2, 3 * MySQL * Evidence of nmap NSE vuln scanning against mysql * Appears to be unsuccessful * LDAPS (`10.168.27.10`) * This appears to be a tcpwrapped service, so can't confirm. * The service is running on the same port as LDAP over SSL (636) * It appears to have been attempted to be accessed multiple times, but it doesnt seem to have been successfully accessed by the suspect address (`10.16.80.243`) - Handshakes seem to complete, but then are immediately reset by the server. * would need TLS keys inorder to be able to verify. * Packets : 14370-213805 * HTTP * Vulnerable by default, in that it is in clear text. * FTP * Anonymous login enabled * Doesn't seem to have been used by the suspect address (`10.16.80.243`)

Nmap Scanning Activity

Nmap Host Scanning : Requests

!nmap-hostscan-rq

Nmap Host Scanning : Responses

!nmap-hostscan-rs

Nmap Port Scan

!nmap-portscan

Artifact of Nmap execution: `niceports.txt.bak`

- Sources: 4, 5

!nmap-artifact

SSH Duplicate Keys

!ssh-dup-1 !ssh-dup-2

MySQL Attempts

!mysql-failed

LDAPS Attempts

!ldaps-failed

FTP Anonymous Login

!ftp-anon

DVWA Application

!dvwa

Nmap scanning - Implications

FTP Anon login

Because an external IP `49.12.121.47` was able to successfully logon to the FTP service on `10.168.27.10` w/o using an authenticated FTP user, they could potentially use the FTP server as a pivot point to access the rest of the network. The minimum implications of anonymous access me the undermining of autheticated access and actions on the `10.168.27.10` host. If an actor is able to access other machines beyond this host the same implications will be replicated on any other host that is subesequently accessed.

DVWA Application

The "Damn Vulnerable Web Application" (DVWA)[6] is an open source tool that is purpose designed to be full of vulnerabilities for users to gain experience with web app penetration testing. Because this application is running on the host, it is full of a number of easy and well known vulnerabilites. If exploited, it will serve as a foothold for a bad actor to be able to pivot into the rest of the network.

Nmap Scanning

Nmap is an open source, network scanning and vulnerability assessment tool. It is commonly used for legitimate purposes, and so can't be immediately called an exploit, nor its ability to run considered a vulnerability. What we can qualify as an anomaly is that it ran w/o authorization. It doesn't seem to have performed any agressive actions against the network, but it could have. The implications of this network traffic having run are that the network doesn't seem to be segmented properly, and that traffic rules don't seem to exist to help prevent unauthorized scanning. If this was run by a bad actor, then it is possible that they are conducting reconnaisance in preparation for further actions.

SSH Duplicate Keys

The ssh daemon uses host keys to uniquely identify itself to connecting clients[2]. If a service is using a duplicate key and being served from a different port, the possiblity exists that an unwitting user could accidentally access a different service and not be made aware. This could allow for a bad actor to carry out an SSH based Man-in-the-Middle (MitM) attack.

Nmap scanning - Solutions to Anomalies

1. Disable Anonymous FTP Logon
2. Disable HTTP, convert to HTTPS only, and redirect all HTTP requests to HTTPS
    - HTTP traffic is sent in plain text, including any logon credentials.
    - Forcing all HTTP requests to redirect to HTTPS will help limit credential spillage by unwitting users.
3. Remove DVWA
    - If this is a necessary tool, it can be securely deployed by deploying it as a container and implementing secure container policies.
4. Convert to FTPS or SFTP, and disable FTP
    - FTP traffic is sent in plain text, including any logon credentials.
5. Implement a network firewall
    - The fact that an external IP `49.12.121.47` indicates that at a minimum an ingress firewall rule doesn't exist, and at the worst, no firewall exists at all.
6. Segment the network
    - The networks should be segmented by subnets, in order to limit infections by bad actors who gain access to the network.
7. Use unique SSH keys for every service
    - Just generate new keys for each service on the host.

Nmap scanning - Sources

1. https://hubbardonnetworking.wordpress.com/2015/03/16/discovering-ssh-host-keys-with-nmap/
2. https://www.digitalocean.com/blog/avoid-duplicate-ssh-host-keys/
3. https://blog.shodan.io/duplicate-ssh-keys-everywhere/
4. https://support.mozilla.org/da/questions/1335631
5. https://www.google.com/search?q=%2Fnice%2520ports%252C%2FTri%256Eity.txt%252ebak
6. https://github.com/digininja/DVWA