Five Steps to Improve Internal Network Security

Since 2009, Verizon has released a yearly Data Breach Investigations Report (DBIR), which summarizes the data breaches from the year before. In each of these reports, the vast majority of attacks have originated from external threat actors. If this is the case, why do we need to worry about internal network security? The answer lies in the attacker's motivation, which is to access sensitive data. Often, the desired data does not reside on the compromised machine, which means the attacker must move through the internal network to get to the machines with sensitive data.

> *After infecting the victim's systems (usually through a phishing e-mail), attackers utilize the backdoor and C2 features to download additional malware and move towards the goal of getting domain-level access. This can be accomplished via keyloggers, capturing credentials stored on end-user systems, or dumping password hashes from the domain controller. Throughout this process, attackers promulgate across the systems within the network, hiding their activities within system processes, searching for and capturing the desired data, and then exporting it out of the victim's environment.*
>
> ***2013 Data Breach Investigations Report 30-31***

The focus of this talk is five issues I feel should be addressed to help prevent this lateral movement through the network. I routinely see these issues while conducting internal penetration tests and each can lead to partial or complete compromise of the internal network.

Eliminate LanMan Password Hashes

LanMan (LM) is a weak password hashing algorithm that has been in use since before Windows NT. LM hashes were kept in newer versions of Windows for backward compatibility but are now turned off by default in Windows Vista, Server 2008 and newer operating systems. The weaknesses in the LM hashing algorithm makes these hashes particularly easy to crack and there are a number of tools, including Ophcrack, rcracki_mt, and Hashcat, that are used to crack these hashes quickly.

LM password hashes can be disabled on the local machine or across the domain using Group Policy[1]. Unfortunately, disabling LM hashes only prevents them from being stored on the disk. Windows still stores them in memory and tools such as Windows Credential Editor (WCE) can still recover them[2]. The only sure way to remove LM hashes from the network is to use passwords with more than 14 characters.

Remove Shared Local Administrator Passwords

In most organizations, every workstation and server share the same local administrator password. So, after compromising one machine and recovering the local administrator password, the attacker then has administrative access to either all or a large portion of the internal network. Using this administrative access, the attacker can then escalate privileges to get domain administrative access.

[1] http://support.microsoft.com/kb/299656

[2] http://computer-forensics.sans.org/blog/2012/02/29/protecting-privileged-domain-accounts-lm-hashes-the-good-the-bad-and-the-ugly/

One solution is to prevent the local administrator from logging in via the network[3], the other is to use a unique local administrator password on each machine[4]. Using tools such as Metasploit, Nmap, or SMBExec, it is possible to identify shared admin passwords on the network.

Lockdown Open File Shares

Open file shares, including SMB, NFS, and anonymous FTP servers, contain a wealth of information for an attacker or malicious insider. Often these shares contain database configuration files, private SSH keys, and protected health information (PHI). Attackers will exfiltrate any sensitive data and use configuration and authentication data to access other servers and databases. Open file shares are often found on network attached storage devices, particularly consumer grade devices, and printers.

Use Nessus, Nmap, or Metasploit to scan the network for open file shares. On Windows machines, disable Simple File Sharing[5], which creates open file shares by defaultand manage share permissions with NTFS[6]. Access to NFS shares should be restricted by IP address and username[7].

Replace Default/Blank Passwords

Default or blank passwords are extremely common througout most enterprise networks and in some cases, can lead to the complete compromise of a server. Default credentials on an Apache Tomcat server paired with one or two of these other issues has lead to the complete compromise of a number of internal networks. Default passwords are often found on web interfaces for printers, UPS devices, NAS devices, and management consoles such as Dell Remote Access Controller.

There is no easy way to find services with default or blank passwords, it requires a combination of scanning and manual password checking. Make it a habit to identify all web services, using tools like Nmap, Nessus, or Metaspoit, and manually check them for blank, default, or weak passwords.

Lockdown Remote Desktop Protocol

Remote Desktop Protocol (RDP) is used throughout many organizations to remotely administer internal machines and is typically configured with no restrictions other than username and password. The Morto worm propagated by bruteforcing the administrator password on RDP servers[8]. In addition, MS12-020 patched a flaw that could allow a remote attacker to execute arbitrary code on RDP servers without authentication. At this point, there is no publicly available exploit for this vulnerability but similar vulnerabilities could be found and exploited in the future.

Access to RDP should be disabled whenever possible. If it is not possible to disable RDP, access should be restricted to particlar IP addresses and user accounts[9]. Use Nessus, Nmap, or Metasploit to identify RDP servers on the network.

[3] http://technet.microsoft.com/en-us/library/cc758316%28v=ws.10%29.aspx

[4] http://jeffmcjunkin.com/2012/06/28/step-by-step-implementation-of-local-administrator-password-randomization-script/

[5] http://support.microsoft.com/kb/304040

[6] http://support.microsoft.com/kb/324267

[7] http://nfs.sourceforge.net/nfs-howto/ar01s03.html

[8] http://www.f-secure.com/weblog/archives/00002227.html

[9] https://security.berkeley.edu/node/94?destination=node/94