CIT Ebikon Consulting

State of the Art aus einer Hand

Windows Server Hardening

Security Hardening Windows Server

10 Essential Steps to Configuring a Server

Whether you're deploying hundreds of Windows servers into the cloud through code, or handbuilding physical servers for a small business, having a proper method to ensure a secure, reliable environment is crucial to success. Everyone knows that an out-of-the-box Windows server may not have all the necessary security measures in place to go right into production, although Microsoft has been improving the default configuration in every server version. We presents this ten step checklist to ensure that your Windows servers have been sufficiently hardened against most attacks.

1. User Configuration

Modern Windows Server editions force you to do this, but make sure the password for the local Administrator account is reset to something secure. Furthermore, disable the local administrator whenever possible. There are very few scenarios where this account is required and because it's a popular target for attack, it should be disabled altogether to prevent it from being exploited.

With that account out of the way, you need to set up an admin account to use. You can either add an appropriate domain account, if your server is a member of an Active Directory (AD), or create a new local account and put it in the administrators group. Either way, you may want to consider using a non-administrator account to handle your business whenever possible, requesting elevation using Windows sudo equivalent, "Run As" and entering the password for the administrator account when prompted.

Verify that the local guest account is disabled where applicable. None of the built-in accounts are secure, guest perhaps least of all, so just close that door. Double check your security groups to make sure everyone is where they are supposed to be (adding domain accounts to the remote desktop users group, for example.)

Don't forget to protect your passwords. Use a password policy to make sure accounts on the server can't be compromised. If your server is a member of AD, the password policy will be set at the domain level in the Default Domain Policy. Stand alone servers can be set in the local policy editor. Either way, a good password policy will at least establish the following:

- Complexity and length requirements - how strong the password must be
- Password expiration - how long the password is valid
- Password history - how long until previous passwords can be reused
- Account lockout - how many failed password attempts before the account is suspended

2. Network Configuration

Production servers should have a static IP so clients can reliably find them. This IP should be in a protected segment, behind a firewall. Configure at least two DNS servers for redundancy and double check name resolution using nslookup from the command prompt. Ensure the server has a valid A record in DNS with the name you want, as well as a PTR record for reverse lookups. Note that it may take several hours for DNS changes to propagate across the internet, so production addresses should be established well before a go live window. Finally, disable any network services the server won't be using, such as IPv6. This depends on your environment and any changes here should be well-tested before going into production.

3. Windows Features and Roles Configuration

Microsoft uses roles and features to manage OS packages. Roles are basically a collection of features designed for a specific purpose, so generally roles can be chosen if the server fits one, and then the features can be customized from there. Two equally important things to do are 1) make sure everything you need is installed. This might be a .NET framework version or IIS, but without the right pieces your applications won't work. 2) Uninstall everything you don't need. Extraneous packages unnecessarily extend the attack surface of the server and should be removed whenever possible. This is equally true for default applications installed on the server that won't be used. Servers should be designed with necessity in mind and stripped lean to make the necessary parts function as smoothly and quickly as possible.

4. Update Installation

This may seem to go without saying, but the best way to keep your server secure is to keep it up to date. This doesn't necessarily mean living on the cutting edge and applying updates as soon as they are released with little to no testing, but simply having a process to ensure updates do get applied within a reasonable window. Most exploited vulnerabilities are over a year old, though critical updates should be applied as soon as possible in testing and then in production if there are no problems.

There are different kinds of updates: patches tend to address a single vulnerability; roll-ups are a group of packages that address several, perhaps related vulnerability, and service packs are updates to a wide range of vulnerabilities, comprised of dozens or hundreds of individual patches. Be sure to peek into the many Microsoft user forums after an update is released to find out what kind of experience other people are having with it. Keep in mind that the version of the OS is a type of update too, and using years-old server versions puts you well behind the security curve.

If your production schedule allows it, you should configure automatic updates on your server. Unfortunately, the manpower to review and test every patch is lacking from many IT shops and this can lead to stagnation when it comes to installing updates. It's much more dangerous, however, to leave a production system unpatched than to automatically update it, at least for critical patches. If at all possible, the updates should be staggered so test environments receive them a week or so earlier, giving teams a chance to observe their behavior. Optional updates can be done manually, as they usually address minor issues.

Other MS software updates through Windows Update as well, so make sure to turn on updates for other products if you're running Exchange, SQL or another MS server technology. Each application should be updated regularly and with testing.

5. NTP Configuration

A time difference of merely 5 minutes will completely break Windows logons and various other functions that rely on kerberos security. Servers that are domain members will automatically have their time synched with a domain controller upon joining the domain, but stand alone servers need to have NTP set up to sync to an external source so the clock remains accurate. Domain controllers should also have their time synched to a time server, ensuring the entire domain remains within operational range of actual time.

6. Firewall Configuration

If you're building a web server, for example, you're only going to want web ports (80 and 443) open to that server from the internet. If anonymous internet clients can talk to the server on other ports, that opens a huge and unnecessary security risk. If the server has other functions such as remote desktop (RDP) for management, they should only be available over a VPN connection, ensuring that unauthorized people can't exploit the port at will from the net.

The Windows firewall is a decent built-in software firewall that allows configuration of port-based traffic from within the OS. On a stand alone server, or any server without a hardware firewall in front of it, the Windows firewall will at least provide some protection against network based attacks by limiting the attack surface to the allowed ports. That said, a hardware firewall is always a better choice because it offloads the traffic to another device and offers more options on handling that traffic, leaving the server to perform its main duty. Whichever method you use, the key point is to restrict traffic to only necessary pathways.

7. Remote Access Configuration

As mentioned above, if you use RDP, be sure it is only accessible via VPN if at all possible. Leaving it open to the internet doesn't guarantee you'll get hacked, but it does offer potential hackers another inroad into your server.

Make sure RDP is only accessible by authorized users. By default, all administrators can use RDP once it is enabled on the server. Additional people can join the Remote Desktop Users group for access without becoming administrators.

In addition to RDP, various other remote access mechanisms such as Powershell and SSH should be carefully locked down if used and made accessible only within a VPN environment. Telnet should never be used at all, as it passes information in plain text and is woefully insecure in several ways. Same goes for FTP. Use SFTP or SSH (from a VPN) whenever possible and avoid any unencrypted communications altogether.

8. Service Configuration

Windows server has a set of default services that start automatically and run in the background. Many of these are required for the OS to function, but some are not and should be disabled if not in use. Following the same logic as the firewall, we want to minimize the attack surface of the server by disabling everything other than primary functionality. Older versions of MS server have more unneeded services than newer, so carefully check any 2012 or 2008 (!) servers.

Important services should be set to start automatically so that the server can recover without human interaction after failure. For more complex applications, take advantage of the Automatic (Delayed Start) option to give other services a chance to get going before launching intensive application services. You can also set up service dependencies in which a service will wait for another service or set of services to successfully start before starting. Dependencies also allow you to stop and start an entire chain at once, which can be helpful when timing is important.

Finally, every service runs in the security context of a specific user. For default Windows services, this is often as the Local System, Local Service or Network Service accounts. This configuration may work most of the time, but for application and user services, best practice dictates setting up service specific accounts, either locally or in AD, to handle these services with the minimum amount of access necessary. This keeps malicious actors who have compromised an application from extending that compromise into other areas of the server or domain.

9. Further Hardening

Microsoft provides best practices analyzers based on role and server version that can help you further harden your systems by scanning and making recommendations.

Although User Account Control (UAC) can get annoying, it serves the important purpose of abstracting executables from the security context of the logged in user. This means that even when you're logged in as an admin, UAC will prevent applications from running as you without your consent. This prevents malware from running in the background and malicious websites from launching installers or other code. Leave UAC on whenever possible.

The tips in this guide help secure the Windows operating system, but every application you run should be hardened as well. Common Microsoft server applications such as MSSQL and Exchange have specific security mechanisms that can help protect them, be sure to research and tweak each application for maximum resilience. If you're building a web server, you can also follow our hardening guide to improve its internet facing security.

10. Logging and Monitoring

Finally, you need to make sure that your logs and monitoring are configured and capturing the data you want so that in the event of a problem, you can quickly find what you need and remediate it. Logging works differently depending on whether your server is part of a domain. Domain logons are processed by domain controllers, and as such, they have the audit logs for that activity, not the local system. Stand alone servers will have security audits available and can be configured to show passes and/or failures.

Check the max size of your logs and scope them to an appropriate size. Log defaults are almost always far too small to monitor complex production applications. As such, disk space should be allocated during server builds for logging, especially for applications like MS Exchange. Logs should be backed up according to your organization's retention policies and then cleared to make room for more current events.

Consider a centralized log management solution if handling logs individually on servers gets overwhelming. Like a syslog server in the Linux world, a centralized event viewer for Windows servers can help speed up troubleshooting and remediation times for medium to large environments.

Establish a performance baseline and set up notification thresholds for important metrics. Whether you use the built-in Windows performance monitor, or a third party solution that uses a client or SNMP to gather data, you need to be gathering performance info on every server. Things like available disk space, processor and memory use, network activity and even temperature should be constantly analyzed and recorded so anomalies can be easily identified and dealt with. This step is often skipped over due to the hectic nature of production schedules, but in the long run it will pay dividends because troubleshooting without established baselines is basically shooting in the dark.

Wrap-Up

Defining your ideal state is an important first step for server management. Building new servers to meet that ideal takes it a step further. But creating a reliable and scalable server management process requires continuous testing of actual state against the expected ideal. This is because configurations drift over time: updates, changes made by IT, integration of new software-- the causes are endless.