Knowledge must have adequate knowledge to be safe in any field. As PowerShell is a scripting language, every command in PowerShell has some or the other purpose to use for n tasks. Whereas AI is used to make work/tasks easier for humans. AI can be useful, but it can’t be secured and reliable as much as scripting done by humans for its purpose. If a task is run by a PowerShell script, a user of it is aware of what purpose this PowerShell script is made for and what and where the changes will be made.

There are a few key points that can be accounted for as counterproductive or dangerous for someone without adequate knowledge for several reasons that are mentioned below:

1. Security: Script generated by AI for a particular task can’t cover all the security concerns for the task use. Whereas manually, when a script is created in PowerShell, security points are taken care-off.
2. Compatibility: The scripts produced by AI might not be optimized for your environment or PowerShell version. Given how dramatically PowerShell scripts can vary between versions and configurations, this may cause compatibility problems and unexpected behavior.
3. New learning: For IT professionals, learning scripting languages and PowerShell is a valuable skill. It can be detrimental to skill development and improvement in this area to rely only on AI-generated scripts.
4. Reliability and Dependency: The use of external AI services or tools, which may have constraints, costs, or availability difficulties, can make you dependent on AI for script development. Reliability is compromised when it comes to AI-generated scripts.

In my opinion, the use of AI in scripting can be helpful, but it can’t be secured or reliable to use in critical areas; it can be used for learning purposes.

Using AI-generated PowerShell scripts in the production environment can cause serious issues for many reasons, such as security, data loss, performance, and maintenance issues.

1. It may be difficult for administrators to manage and troubleshoot AI-generated scripts since they lack adequate documentation. Administrators or users may struggle to change AI-generated scripts if they don’t have a proper understanding.
2. AI-generated scripts could include harmful code, which could result in security breaches or data leaks. It can leak sensitive data.
3. AI-generated scripts can be more poorly scripted than self-generated scripts of PowerShell.
4. Self-generated scripts can perform faster than AI-generated scripts. Sometimes AI-generated scripts may lead to script failure and may lead to major loss in production/critical areas.

**PowerShell Script Security**

The Configuration Manager scripts feature lets you visually review and approve scripts. Another administrator can request that their script is allowed. Administrators should be aware PowerShell scripts can have obfuscated scripts. An obfuscated script could be malicious and difficult to detect with visual inspection during the script approval process. Visually review PowerShell scripts and use inspection tools to help detect suspicious script issues. These tools can't always determine the PowerShell author's intent, so it can bring attention to a suspicious script. However, the tools will require the administrator to judge if it's malicious or intentional script syntax. (Gowdhamankarthikeyan, 2022, para. 2)

AI-generated scripts can be useful for automating processes, but they must always be utilized carefully and in addition to a deep knowledge of PowerShell and as per the requirements of the enterprise context.

After reviewing Conestoga's Academic Integrity Modules, if we take module 2 “Maintaining Academic Integrity to Achieve Success.” For example, to make something function with the help of an AI-generated script we don’t know from where and from whose context that script is generated, which can challenge the integrity point. In Contrast to AI, if the script is generated by an administrator or by an IT professional, they can avoid integrity issues. Without mastery in PowerShell scripting with the help of AI can lead to a lot more issues and can have many different impacts on an enterprise and its users it in various ways. Suppose a company wants to maintain the daily attendance of employees in the office and it generates an AI-generated PowerShell script, there are chances that might give a copy of another company script used for attendance which is an integrity breach.

Reference:

Gowdhamankarthikeyan. (2022, October 4). *Learn more about PowerShell script security - Configuration Manager*. Microsoft Learn. https://learn.microsoft.com/en-us/mem/configmgr/apps/deploy-use/learn-script-security#powershell-script-security