Click to prove you're human

Hosted runners for every major OS make it easy to build and test all your projects. Run directly on a VM or inside a container. Use your own VMs, in the cloud or on-prem, with self-hosted runners. Save time with matrix workflows that simultaneously test across multiple operating systems and versions of your runtime. GitHub Actions supports Node.js, Python, Java, Ruby, PHP, Go, Rust, .NET, and more. Build, test, and deploy applications in your language of choice. See your workflow run in realtime with color and emoji. It's one click to copy a link that highlights a specific line number to share a CI/CD failure. Automate your software development practices with workflow files embracing the Git flow by codifying it in your repository. Test your web service and its DB in your workflow by simply adding some docker-compose to your workflow file. You can't perform that action at this time. You can't perform that action at this time. You can't perform that action at this time. Page 2 You can't perform that action at this time. You can't perform that action at this time. Notice 1: We are excited to announce that our current tool has been ported to a PowerShell version. This means that users can now access and use the tool directly from the PowerShell command line, making it even more convenient and efficient to use. We believe that this new version will greatly benefit our users and enhance their experience with the tool. Thank you for your continued support and we hope you enjoy the new PowerShell version: Notice 2: We have recently learned that Microsoft has enabled the account lockdown policy by default in modern and up-to-date versions of Windows. This policy helps to secure the system by locking an account after a certain number of failed login attempts. While this is a beneficial security measure, it renders the proof-of-concept (PoC) inefficient on these systems. Release date: 2020-05-14 Target: Windows XP to Latest Windows 10 Version (1909) Weakness location : LogonUserA, LogonUserW, CreateProcessWithLogonA, CreateProcessWithLogonW WinBruteLogon.exe -u -w type | WinBruteLogon.exe -u - Win Brute Logon is designed to simulate a brute-force attack on a Microsoft account by guessing large numbers of password combinations in a short amount of time. This allows pentesters to test the security posture of their systems and assess their defenses against brute-force attacks. The tool exploits the lack of an account lockout mechanism, which is a common weakness in many systems (before account lockout becomes enabled by default on Windows 11). By attempting to guess the password of an account, the tool can help pentesters identify and address vulnerabilities in their security measures. It should be used responsibly and within the bounds of the law. For this demonstration, we will set up a fresh version of Windows 10 on a virtual or physical machine. Once the machine is set up, log in as an administrator. Next, create two different local accounts: one administrator account and one regular user account. Please note that although we will be using the guest account for the demo, this proof-of-concept (PoC) is not limited to the guest account. It can be used from any account or group, including guest, regular user, and admin user. net user darkcodersc /add net user darkcodersc trousers (trousers is the password) net localgroup administrators darkcodersc /add net user HackMe /add net user HackMe ozlq6qwm (ozlq6qwm is the password) net user GuestUser /add net localgroup users GuestUser /delete net localgroup guests GuestUser /add In my case both trousers and ozlq6qwm are in SecList : To begin the demonstration, log off from the administrator account or restart the machine and log in to the guest account. Then, place the PoC executable in a location where you have access as a guest user. Usage : WinBruteLogon.exe -v -u -w -v is optional, it design the verbose mode. By default, domain name is the value designated by %USERDOMAIN% env var. You can specify a custom name with option -d prompt(guest)>WinBruteLogon.exe -v -u darkcodersc -w 10k-most-common.txt Wait few seconds to see the following result: [..] Load 10k-most-common.txt file in memory... [DONE] 10002 passwords successfully loaded. [INFO] 2 cores are available [..] Create 2 threads... [INFO] New "TWorker" Thread created with id=2260, handle=364 [INFO] New "TWorker" Thread created with id=3712, handle=532 [DONE] Done. [OK] Password for username=[darkcodersc] and domain=[DESKTOP-0885FP1] found = [trousers] [..] Finalize and close worker threads... [INFO] "TWorkers"(id=2260, handle=364) Thread successfully terminated. [INFO] "TWorkers"(id=3712, handle=532) Thread successfully terminated. [DONE] Done. [INFO] Ellapsed Time : 00:00:06 prompt(guest)>WinBruteLogon.exe -v -u HackMe -w 10k-most-common.txt Wait few seconds to see the following result: [..] Load 10k-most-common.txt file in memory... [DONE] 10002 passwords successfully loaded. [INFO] 2 cores are available [..] Create 2 threads... [INFO] New "TWorker" Thread created with id=5748, handle=336 [INFO] New "TWorker" Thread created with id=4948, handle=140 [DONE] Done. [OK] Password for username=[HackMe] and domain=[DESKTOP-0885FP1] found = [ozlq6qwm] [..] Finalize and close worker threads... [INFO] "TWorkers"(id=5748, handle=336) Thread successfully terminated. [INFO] "TWorkers"(id=4948, handle=140) Thread successfully terminated. [DONE] Done. [INFO] Ellapsed Time : 00:00:06 "In a real-world scenario, if an attacker gains access to a low-privileged user account, they may be able to crack the password of a more privileged user and escalate their privileges. To mitigate this risk, there are a few steps that can be taken: If present, disable any guest accounts. Implement application white-listing to restrict the execution of unauthorized software. Follow guidelines for creating and maintaining strong passwords for all users. To implement a security lockout policy (which is not enabled by default), follow these steps: Open the 'secpol.msc' utility. Navigate to 'Account Policies' > 'Account Lockout Policy' Edit the 'Account lockout threshold' value with a desired number of attempts (from 1 to 999). This value represents the number of failed login attempts before the account is locked. Please note that the lockout policy does not apply to the administrator account. In this case, the best protection for the administrator account (if enabled) is to set up a very complex password. A report detailing this weakness has been sent to the Microsoft Security Team. They should consider enabling the account lockout policy by default." (UPDATE 2022) : Account lockout finally enabled by default. image/svg+xmlBotttsPablo Stanley Körner Ollie Janneck BruteForcer is a software tool designed to help users recover lost or forgotten passwords. It uses a brute-force attack to try all possible combinations of characters in order to recover the correct password. It is a powerful tool that can be used to protect users' online accounts and data. image/svg+xmlBotttsPablo Stanley Körner Caleb F. I recently tried the BruteForcer software to help with a project. It was easy to use, with a nice user interface. I found that it had all the options I needed to get the job done in a timely manner. The speed of the program was also quite good. It was able to finish the task quickly. It was also able to work with a variety of file formats. Overall, I found the software to be quite useful. The price was also quite reasonable. It was able to save me a lot of time. The support team was also quite helpful and quick to respond. image/svg+xmlBotttsPablo Stanley Körner Harrison Z. 1. BruteForcer is a powerful tool for testing password strength. 2. It provides a range of options for customizing tests. 3. It was easy to set up and use. 4. The reports were helpful in understanding the security of my passwords. 5. The results gave me a good indication of which passwords needed to be changed. image/svg+xmlBotttsPablo Stanley Körner Cooper P. This software is a password cracking tool that uses brute force algorithms to guess passwords by trying all possible combinations until the correct one is found. It can be used to crack passwords for various applications and services, including email accounts, FTP servers, and social media platforms. The software allows users to customize the brute force attack parameters, such as password length, character set, and dictionary attacks. It also has the ability to pause and resume the attack process and save the results for future use. image/svg+xmlBotttsPablo Stanley Körner Harris Giglio BruteForcer software is a tool used for attempting all possible combinations of passwords to gain unauthorized access to a system or account. image/svg+xmlBotttsPablo Stanley Körner Andrew This tool excels at network-based password cracking with a user-friendly GUI and robust hashing algorithms. image/svg+xmlBotttsPablo Stanley Körner Matthew Easy to understand interface, effective password cracking tool. image/svg+xmlBotttsPablo Stanley Körner Logan Efficient, exhaustive, straightforward password cracking. This password cracking software is easy to use and very comprehensive. It has been designed to help the user protect their confidential data. It supports a wide range of algorithms, including AES, SHA-1, SHA-2, MD5 and much more, ensuring that your data remains protected against the most advanced hacking techniques. It automatically generates passwords of varying lengths and complexities to ensure optimal data protection. Compatible with a range of operating systems, including Windows, Mac, and Linux, and capable of running on multiple threads at once, it allows for quick and efficient password cracking. Its settings are highly configurable to meet your specific needs. Hacks passwords. Easy to use. Saves time. Overcomes security. Powerful tool. You can't perform that action at this time. Hackers and penetration testers use brute force attack tools to crack login credentials and encryption keys through systematic trial and error. These tools automatically test various combinations of numbers, letters, and special characters to uncover passwords. Most brute force tools are automated bots that can run between 10,000 and 1 billion combinations per second using powerful machines. Threat actors use brute force tools to guess login credentials and encryption keys to fulfill their malicious purposes, such as taking over an account, stealing data, encrypting data, and stealing money. Conversely, penetration testers and security researchers employ brute force tools to identify login credentials and encryption keys' vulnerabilities. Pentesters also use brute forcing to find hidden subdomains and directories associated with main domains. Gobuster – Best for Brute Forcing Subdomains and DirectoriesBruteX – Best for Brute Forcing ServicesDirsearch – Best for Discovering Web PathTHC-Hydra – THC-Hydra: Best for Port ScannerBurp Suite – Best for Web Security Testing SuitePatator – Multi-Purpose Brute ForcerPydictor – Dictionary BuilderHashcat – GPU-Accelerated Password CrackingJWT Cracker – Best for JWT TokenNettracker – Best for Automated PentestSocialBox – Best for Social Media TestingCMSeek – Best for CMS TestingShow moreShow less Geekflare has researched and compiled a list of the top brute force attack tools based on key features such as protocol support, error handling, and logging capabilities. Gobuster is a leading tool for brute-forcing URLs (directories and files) on websites, DNS subdomains (with wildcard support), open Amazon S3 buckets, virtual host names, TFTP servers, and more. Its speed and efficiency, driven by the Go programming language and dedicated modes for DNS and directory brute-forcing, make it one of the best tools for subdomains and directories. Gobuster also allows customization through options like defining HTTP methods, specifying wordlists, using patterns, and proxy configurations. It is frequently updated with improvements such as range support for status codes, TLS 1.0 and 1.1 support, and advanced DNS enumeration features. The best use cases for Gobuster include discovering hidden web directories, subdomains, unsecured S3 and Google Cloud buckets, virtual hosts, TFTP files, and fuzzing HTTP inputs to enhance security testing. It is available via binaries, Docker, or by building from the source with Go 1.19 or higher. Try Gobuster BruteX is a powerful tool for brute-forcing services on a target. It can target open ports, IP addresses, usernames, passwords, and more. Available on GitHub, It can also be run in a Docker environment for added flexibility. It supports brute-forcing SSH, FTP, and other network services to help identify weak passwords and insecure systems. Get BruteX Dirsearch is a command-line website directory scanner. It has many useful features, such as multi-threading, recursive brute-forcing, HTTP proxy support, detection of invalid web pages, and more. You can install it through various methods, including Git, ZIP, Docker, and PyPi. Dirsearch offers various useful dictionary settings for efficient brute forcing, such as custom wordlists, adding extensions to the end of every wordlist entry, adding custom prefixes to all wordlist entries, and more. It also allows you to pause the scanning progress, save the progress, skip the current target, and more. Try Dirsearch Hydra is a versatile brute-force tool for security testing developed by Van Hauser of THC. It supports a wide range of protocols, including FTP, HTTP, SSH, SMTP, MYSQL, Telnet, POP3, RDP, IMAP, LDAP, and many more. Hydra is super fast and flexible, allowing you to add new modules easily. What's more, it lets you have parallelized connections, enhancing its speed and efficiency. Hydra is compatible with platforms like Linux, MacOS, Solaris, and Windows/Cygwin. For convenience, it can also be deployed using Docker on any operating system. Try Hydra Burp Suite is a powerful web application vulnerability scanner that helps pentsters to find a range of vulnerabilities in an application, including directory reversal, OWASP Top 10, HTTP Desync attacks, and more. Though it is a vulnerability scanner, Burp Suite comes with various features to help you brute force the password of a given user using a dictionary attack and try every permutation of a character set. Burp Enterprise Edition has custom pricing, while Burp Professional Edition costs $449 per user per year. The Burp Community Edition is free, and Burp Scanner offers a full-featured trial for evaluation. Try Burp Suite Patator is the top choice for multi-purpose brute forcing. It can help you brute force logins for FTP, SSH, Telnet, SMTP, RDP, IMAP, Lightweight Directory Access Protocol (LDAP), Oracle, MySQL, and many more. Written in Python, Patator is highly valued for its role in password discovery, vulnerability scanning, and reconnaissance. Available for free on GitHub, it's a powerful tool for cybersecurity professionals looking for efficient and customizable brute-force capabilities. Try Patator Pydictor is a versatile wordlist generator tool developed by LandGrey and available on GitHub. It allows you to create general, custom, or social engineering wordlists to carry out brute-force attacks. Key features of Pydictor include but are not limited to creating custom character wordlist, permutation and combination wordlist, configuration file based wordlist, extending wordlist based on rules, wordlist based on web page keywords, and many more. Pydictor supports Python 2.7 and Python 3.x version, and you can run it on Linux, Windows, and Mac. You can download it from GitHub. Try Pydictor Hashcat is an advanced password-cracking tool that supports five unique modes of attack: dictionary attack, combinator attack, brute force attack, hybrid attack, and association attack. It allows you to attack over 300 highly optimized hashing algorithms and lets you crack multiple hashes simultaneously. Hashcat supports various hardware accelerators on Linux, Windows, and macOS, including central processing units (CPUs), graphic processing units (GPUs), and more. It also allows you to enable distributed password cracking. Hashcat offers various utilities, such as cap2hccapx to generate .hccapx files, cleanup-rule, combinator, and more, for advanced cracking of password hashes. If you are stuck and want some help, the Hashcat forum offers tons of information. You can download Hashcat from GitHub. Try Hashcat JWT Cracker is designed to brute-force JSON Web Tokens (JWT) using the S256, HS384, or HS512 algorithms. It can effectively brute-force tokens with weak secrets. You can install JWT Cracker via Node Package Manager (NPM) with a simple command; it requires Node.js version 16.0.0 or higher. After the installation, you can run JWT Craker from the command line to target specific JWT tokens. Get JWT Craker Nettacker is a powerful tool for automated penetration testing. It offers various modules for information gathering and pen testing. You can conduct vulnerability scanning, brute force, check misconfigurations, and more. It uses various protocols, including ACK, SYN, TCP, ICMP, and more, to identify and bypass IDS/IPS/Firewall services. Nettracker offers three modules: scan modules, vuln modules, and brute modules. Its brute modules allow you to brute force FTP users, HTTP basic auth users, SMTP ports, SSH (port 22) users, WordPress users, and more. You have an option to specify extra users/parameters. If you don't specify, it will use its default parameters. You can install OWASP Nettacker directly on a Linux system. Alternatively, you can run it on any operating system using a Docker image. Get Nettracker SocialBox is a popular attack framework to brute force social media accounts like Facebook, Instagram, and Gmail. It was coded by Belahsan Ouerghi. It allows you to automate the process of attempting access to these platforms by systematically trying different password combinations. You can install SocialBox on various Linux distributions, such as Ubuntu and Kali Linux, using a straightforward command-line installation process. You can install it from GitHub. Using SocialBox to brute force others' social media accounts is illegal. So you should never do this. Use this tool only when you want to assess the security of your social media accounts. Get SocialBox CMSeek is best known for scanning content management systems and brute forcing them using its pre-built or custom-made brute forcing modules. It can detect around 180 different CMS platforms and comes with advanced scanning capabilities for popular CMS, including WordPress, Drupal, and Joomla. When scanning for content management systems, CMSeek offers the flexibility to skip certain CMS platforms and avoid scanning targets whose CMS platforms have already been identified. CMSeek employs various methods for detecting CMS, including Directory check, HTTP headers, generator meta tag, page source code, and robot.txt. It stores CMS scanning results in a JSON file and brute forcing results in a Txt file. CMSeek is an open-source tool built using Python 3. Therefore, you will need Python 3 to run it. At present, it works only on Unix-based operating systems. To enforce auto-update, you need to install Git on your system. Get CMSeek There are six key types of brute force attacks, including simple brute force attacks, dictionary attacks, hybrid brute force attacks, reverse brute force attacks, password spraying attacks, and credential stuffing attacks. Knowing about these different types of attacks can help you better secure company accounts by enforcing strict password policies. Below, we have explained briefly common types of brute force methods. Simple Brute Force Attacks: Simple brute force attacks involve guessing actual passwords using combinations of commonly used, weak passwords like 123456789. Though hackers employ brute attack tools to carry out simple brute force attacks, they can also do it manually if the actual passwords are not long or complex. Dictionary Attacks: Hackers use a premade list of words and phrases like those found in a dictionary to run against a single username. Dictionary attacks leverage known combinations or words rather than random letters to brute-force a password. For example, sunshine. Hyrbrid Attacks: Threat actors combine the elements of both simple brute force attacks and dictionary attacks to carry out hybrid attacks. For example, they can use password12345 or shine098765 to run against a username. Reverse Brute Force Attacks: Hackers run reverse brute force attacks to find the usernames of accounts whose passwords they already know. They often scout around various dark web forums that publish leaked data to collect passwords. Password Spraying Attacks: In password spraying attacks, threat actors attempt to gain access to multiple user accounts by using a small set of commonly used passwords. They hope that some users have chosen these weak passwords, which could allow the attackers to infiltrate the corporate network. A hacker, for example, targets multiple employee accounts using a single password like "Password123." If this doesn't give any success, they will use another password and spray against multiple users. Credential Stuffing Attacks: Hackers use stolen credentials to access other accounts, betting that some victims may reuse the same login details. For example, a hacker may get an email and password from a data breach and try them on bank accounts or social media platforms like X or Facebook, hoping the user uses the same credentials elsewhere. As you can see, using common passwords is the root cause of various types of brute-force attacks. Therefore, you should ensure that everyone in your organization creates strong passwords and stops using common passwords. According to Hive Systems, an 8-character bcrypt hash using numbers, uppercase and lowercase letters, and symbols with 32 iterations can take up to 5 days to crack with an A100 x 10,000 GPU. The speed at which a brute force attack cracks a password depends on various factors, including password length and complexity and the computational power of a brute forcing machine. To make your password immune to brute force attacks, you should create complex, at least 18-character long passwords. To add additional layers of security to your company accounts, you can also implement multifactor authentication and an account lockout policy after certain failed login attempts. We need penetration testing tools to identify vulnerabilities, simulate real-time attacks to evaluate how an application's defense performs, and ensure compliance, as many industries, such as finance, require regular security assessments. As penetration testing tools help you identify vulnerabilities that hackers can exploit, they also help you take proactive steps to fix those. So, they improve security posture. Weak passwords are easy security holes hackers can crack to cause a network breach. Once inside, they can carry out various malicious activities, such as installing malicious software and stealing data to cause a data breach or reputational damage. So, you should take proactive steps to mitigate brute-force attacks. The following are key strategies you can implement to mitigate brute force attacks and other password attacks: Set Account Lockout Policies: Ensure that accounts are locked after a predefined number of failed login attempts to deter repeated guessing. Enforce the Use of Complex Passwords: Instruct users to create at least 18 characters long and strong passwords that combine letters, numbers, and symbols. Enable Two-Factor Authentication (2FA): Add an extra layer of security to your company accounts by implementing a second form of verification, such as a text message code. Monitor Login Attempts: Track failed login attempts and flag accounts that show unusual activity. Introduce Delays: Add random delays between login attempts to slow down automated attacks. Limit Login Attempts from an IP Address: Enforce a temporary block on an IP address after a certain number of failed login attempts. Use CAPTCHA: Implement CAPTCHAs to distinguish between human users and automated bots during the login process. Vary Error Messages: Change error messages for failed login attempts to confuse automated tools. Restrict Access by IP Address: Allow users to log in only from specific, trusted IP addresses for added security. Use Rate Limiting: Limit the number of login requests per user or IP address over a certain time period to mitigate attacks. You should also encourage users to follow best practices for password security, such as not reusing passwords and changing them regularly. Using a team password manager can make password management easier in an organization. What is a brute force username and password tool? A brute force username and password tool is a security tool used by researchers, penetration testers, and hackers to crack username and password combinations through trial-and-error methods. What would block brute force attacks? The most effective way to block brute force attacks is to set an account lockout policy after a certain number of failed login attempts. This prevents hackers from gaining unauthorized access to your company accounts by perpetually brute-forcing passwords. Other methods to significantly lower success rates of brute force attacks include creating complex passwords and enabling multi-factor authentication on all your company accounts. What is the brute force attack tool for RDP? THC Hydra is an effecinet tool to brute force RDP. It is available on Linus, macOS, and Windows/Cygwin. You can also use it on any device deploying it in the Docker environment. However, using Hydra to brute force RDP other than for testing or educational purposes is a criminal offense. I work for a small-medium company doing most of the IT tasks, including systems administration, network administration, and even developing in-house software. Recently, my boss (the CEO) told me that some money had been stolen from the company (electronically, like a wire transfer or something) and there was a strong indication that it was done by one of our employees. The company hired a third party to investigate and one thing they did was look through a number of workstations, including the CEO's. One thing that caught my attention was he told me that the "investigator" wanted to look through his computer, which was either locked or not logged in at all. He offered to type in his password so the guy could do what he needed, but he declined and then plugged in a USB stick and suddenly was logged into the account. I've tried searching online for information about a USB exploit like this, but I mostly came across information about the BadUSB vulnerability, which doesn't really seem like it applies here. Does anybody know of any USB exploits like this? The computer runs Windows 10 64-bit. I work for a small-medium company doing most of the IT tasks, including systems administration, network administration, and even developing in-house software. Recently, my boss (the CEO) told me that some money had been stolen from the company (electronically, like a wire transfer or something) and there was a strong indication that it was done by one of our employees. The company hired a third party to investigate and one thing they did was look through a number of workstations, including the CEO's. One thing that caught my attention was he told me that the "investigator" wanted to look through his computer, which was either locked or not logged in at all. He offered to type in his password so the guy could do what he needed, but he declined and then plugged in a USB stick and suddenly was logged into the account. I've tried searching online for information about a USB exploit like this, but I mostly came across information about the BadUSB vulnerability, which doesn't really seem like it applies here. Does anybody know of any USB exploits like this? The computer runs Windows 10 64-bit. You can't perform that action at this time. I work for a small-medium company doing most of the IT tasks, including systems administration, network administration, and even developing in-house software. Recently, my boss (the CEO) told me that some money had been stolen from the company (electronically, like a wire transfer or something) and there was a strong indication that it was done by one of our employees. The company hired a third party to investigate and one thing they did was look through a number of workstations, including the CEO's. One thing that caught my attention was he told me that the "investigator" wanted to look through his computer, which was either locked or not logged in at all. He offered to type in his password so the guy could do what he needed, but he declined and then plugged in a USB stick and suddenly was logged into the account. I've tried searching online for information about a USB exploit like this, but I mostly came across information about the BadUSB vulnerability, which doesn't really seem like it applies here. Does anybody know of any USB exploits like this? The computer runs Windows 10 64-bit. In this post, we will be introducing a new minimalistic tool for local privilege escalation attacks in Microsoft Windows systems. The tool is called localbrute.ps1 and it is a simple local Windows account brute force tool written in pure PowerShell. It doesn't require any 3rd party modules and it is very small in size, which makes it a viable addition to traditional privilege escalation attacks, applicable to various penetration testing scenarios. Attacking local administrative accounts such as the built-in "Administrator" account or any other account that is member of the "Administrators" local group, can be quite an interesting attack vector mainly because of the lack of account lockout policy. We can literally try as many login attempts as we want. If we succeed, we will have full control over the system and we will be able to do all the juicy things that we like to do as pentesters, for example: Disable any defenses and security controls on the systemExtract plaintext credentials from memory and other places (files, registry etc.)Craft raw network packets and run exploits to attack other systems on the networkAccess protected areas of the system to locate sensitive information and many other things .. All these things can help us with the lateral movement, moving further into the infrastructure and demonstrating impact to the customer. Attacking local Windows accounts is nothing new, but here we DO NOT aim to do it remotely using typical pentesting tools such as Metasploit smb_login scanner, Nmap smb-brute NSE script, CrackMapExec or any other similar tools. The localbrute PowerShell tool presented here does the brute forcing locally on the target system itself and so its usage is quite specific.. This tool can be useful in cases where we have gained a low privileged user access to a Windows machine and we can run commands on it – e.g. via RDP session or via terminal services. We could also use this tool in case when we are testing some kind of a restricted or isolated environment – e.g. a VDI environment where we were provided only a user level access and now we are supposed to do the testing from there with limited access to our favorite pentesting utilities. Another use case would be a simulation of a disgruntled employee. Having access to a sample employee workstation, possibly hardened and protected by various security controls. Could we do something and cause a potential damage to the organization? In all these cases, the localbrute.ps1 tool could help us in escalating our privileges. In a nutshell, the localbrute.ps1 tool performs automated login attempts locally on the system, using built-in Windows functionalities. Here are the main features of the tool: Performs login attacks against any selected local account using a supplied wordlistSmall and minimalistic – can be easily typed out by hand (on the keyboard)Written in pure PowerShell – no additional modules neededNon-malicious – undetected by AV / EDR solutions There are two versions of the localbrute.ps1 tool currently available in the GitHub repository – the extra-mini version and the normal version. The only difference is that the normal version is slightly longer and it has the following additional features: Supports resuming, if interruptedDetects already compromised user accounts The following sections describes how to use the tool and how does it work in detail. 1) First thing we need to do is to identify administrative user accounts on the system. These typically include: Members of the local Administrators groupThe local Administrator account itself Here's how we can find members of the local Administrators group: net localgroup administrators 2) Now to run the localbrute tool, simply do: Import-Module .\localbrute.ps1 # Usage: localbrute [debug] # Example: localbrute Administrator .\rockyou.txt Here's an example: Note that it can take a long time until the password is found. See below.. The tool simply iterates through the supplied wordlist (password list) line by line and tries to authenticate as the specified user account locally on the system. It uses internal Windows DirectoryServices.AccountManagement functionalities in the context of the local machine. In effect, this allows us to test authentication for any local account. Here's a standalone PowerShell code snippet to validate single pair of credentials locally: $u = 'Administrator' $p = 'Pa$$w0rd!' Add-Type -AssemblyName System.DirectoryServices.AccountManagement $t = [DirectoryServices.AccountManagement.ContextType]::Machine $a = [DirectoryServices.AccountManagement.PrincipalContext]::new($t) $a.ValidateCredentials($u,$p) With a little bit of PowerShell scripting, we simply wrapped this code into a loop and that's exactly how the localbrute tool works. The enhanced (longer) version has some additional functionality to improve usability when working with large wordlists. Namely it keeps a state file (localbrute.state) in the current working directory to keep track of the progress. Upon interruption (^C), the tool will record the last password candidate that was reached from the given wordlist for the given username. This allows the tool to continue (resume) the attack once the tool is restarted. The state file also keeps record of already compromised accounts. You can turn on the debug mode to see exactly what the tool is doing. Here's an example: Now when it comes to speed, the tool can do around 100-200 login attempts per second, depending on the system performance. True, this is not particularly breathtaking, but it is still much faster than any SMB login capable tool directed to do remote login attack on local accounts over the network. Here's a better overview on the attack speed based on the runtime duration: Runtime durationNumber of login attempts1 second100 – 2001 minute6k – 12k1 hour360k – 720k1 day8.6M – 17.3M This means that we could for example process the whole rockyou.txt wordlist (14.3M entries) between 19.9 – 39.8 hours (1-2 days). This is not so bad and in the outlined scenarios above, it's definitely realistic to have the brute forcing attack running for a prolonged period of time. Note that if the debug mode is on, the speed is impaired by about 20-30%. BEWARE: Running multiple instances of the localbrute script in parallel will NOT increase the speed. In fact, it will result in the following exception very soon: Exception calling "ValidateCredentials" with "2" argument(s): "Multiple connections to a server or shared resource by the same user, using more than one user name, are not allowed. Disconnect all previous connections to the server or shared resource and try again. So, DO NOT run the script in parallel, because at any point in time, only one instance can be calling the ValidateCredentials() system method. The presented localbrute.ps1 script is a simple login brute force tool that can offer an additional method of privilege escalation attacks on Windows systems. Due to the lack of account lockout policy on local accounts, we can use it to test the password strength of the locally privileged accounts and discover accounts configured with weak passwords. Thanks to its compact size, it can come handy during a variety of penetration tests and offensive simulations, similarly as the other minimalistic tools released earlier: Hope you will find it useful sometimes! If you like our tools and you would like more, please do subscribe to our mailing list and follow us on Twitter, Facebook or Github to not miss any new additions! Source: thesecmaster.comBrute force attacks are one of the top three ways that Windows computers are attacked today. These attacks involve malicious actors trying to guess user passwords by repeatedly trying different password combinations. If successful, the attacker gains access to the compromised account and can further penetrate the system.Windows devices have traditionally been vulnerable to brute force attacks against local administrator accounts. This is because Windows did not allow built-in local Administrator accounts to be locked out, no matter how many failed login attempts occurred. Attackers could essentially launch an unlimited number of password guesses over the network against the administrator account.However, Microsoft has introduced new security capabilities in recent Windows versions to counter brute force password attacks against local administrator accounts. By properly configuring new Group Policy settings, you can now lockout local admin accounts after a specified number of failed login attempts. This significantly raises the bar for attackers trying to breach systems via brute forcing credentials.In this tutorial post, we will see what is a Brute Force attack, how Microsoft addressed this problem, what Group Policies setting that protects a Windows PC from Brute Force Attacks, how to enable Administrator Account Lockout Policies, and ultimately, how you can protect your Windows PC from Brute Force Attacks using Group Policies.Table of contents· What is Brute Force Attack and It's Implications on Windows Local and Domain Accounts?· How Microsoft Addressed this Problem Through its Group Policies? ◦ New Lockout Policies for Local Admin Accounts ◦ Default Security Improvements· How to Enable Administrator Account Lockout Policies? ◦ How to Enable Administrator Account Lockout Policies in a Windows domain environment? ◦ How to Enable Administrator Account Lockout Policies in a Local Windows Computer? ◦ Prerequisites· Windows Editions That Support Account Lockout Policies· Bottom LineWhat is Brute Force Attack and It's Implications on Windows Local and Domain Accounts?A brute force attack is a password-cracking method that tries all possible password combinations until the correct password is found. The attacker tries every possible alpha-numeric and special character combination to gain access to a system.Brute force attacks are carried out both manually and using automated password-cracking tools and scripts. Automated attacks are especially powerful when leveraging the processing power of modern GPUs. An unlimited brute force attack can crack most passwords in just hours or days.The implication of a successful brute force attack on a Windows computer is complete compromise of the breached account. If an attacker can brute force guess the password of a local admin account, they gain full control of the system. Brute forced domain admin credentials can give an attacker broad access to domain resources.Once valid credentials are obtained, attackers often use them in follow-on lateral movement and privilege escalation. The initial brute forced account provides the foothold into the environment. Attackers then try to expand access and move towards high-value targets.Beyond standard passwords, attackers are also having increasing success brute forcing other authentication mechanisms like SSH keys, NTLM hashes, and Kerberos tickets. So brute force risks extend beyond just guessing login passwords.How Microsoft Addressed this Problem Through its Group Policies?Previously, Windows did not apply account lockout policies to local administrator accounts. So there was no limit to the number of failed password guesses an attacker could make against these accounts when attempting remote access.To counter the brute force threat against local Windows administrator accounts, Microsoft has introduced new security policies that allow these accounts to be locked out if there are too many invalid login attempts.Group Policy Settings to enable Brute Force Protection on Windows 11Group Policy Settings to enable Brute Force Protection on Windows 10New Lockout Policies for Local Admin AccountsIn Windows 10 and Windows 11, Microsoft has added new settings under Local Computer Policy > Computer Configuration > Windows Settings > Security Settings > Account Policies > Account Lockout Policies that allow local administrator accounts to be locked out.The following lockout policies can now be configured:Account lockout threshold — Specifies the number of invalid login attempts that will cause a user account to be locked out. For example, setting this to 10 will lock out an account after 10 failed login attempts.Account lockout duration — Determines the number of minutes a locked-out account remains locked out before automatically becoming unlocked. This resets the failed login counter to 0.Reset account lockout counter after — Specifies the number of minutes that must elapse after a failed login attempt before the failed login counter is reset to 0. This only applies if the account is not locked out.By enabling these new account lockout policies for local administrators, Windows machines gain enhanced brute force protection. Attackers can no longer indefinitely keep trying to guess admin passwords remotely.Default Security ImprovementsMicrosoft has also improved the default out-of-the-box security posture for local administrator accounts in the latest Windows 11 and Windows Server 2022 versions:The account lockout policies are now enabled by default during the initial operating system setup.Password complexity requirements are also now enforced by default on local administrator accounts on new Windows 11 and Windows Server 2022 installations.These changes significantly strengthen default brute force protections for fresh installations. However, customers with existing systems will need to manually configure the account lockout policies to gain increased protection.How to Enable Administrator Account Lockout Policies?The administrator account lockout policies are not enabled by default on earlier Windows client and server versions prior to Windows 11/Windows Server 2022. To gain enhanced brute force protection, you need to manually enable the lockout policies via Group Policy. Let's see how to enable administrator account lockout policies in both 'Windows domain environment' and 'Local Windows computer'.How to Enable Administrator Account Lockout Policies in a Windows domain environment?Here are step-by-step instructions to enable administrator account lockout in your Windows domain environment:On your Windows domain controller, launch the Group Policy Management Console (GPMC).Right-click the Group Policy Object (GPO) you wish to configure and click Edit.Navigate to Computer Configuration > Policies > Windows Settings > Security Settings > Account Policies > Account Lockout Policy.Double-click on the Account lockout threshold and set to the desired number of failed logon attempts to lockout accounts. For example, set to 10 invalid logon attempts.Double-click on Account lockout duration and set to the desired lockout period in minutes once the account lockout threshold is reached. For example, set to 10 minutes.Double-click on Reset account lockout counter after and set to the desired time period that must elapse before the failed logon counter is reset to 0. For example, set to 10 minutes.Click OK to apply your lockout policy settings.Link the GPO to your Organizational Unit (OU) containing the Windows computers you want to protect from brute force attacks.After the Group Policy is applied, local administrator accounts on your Windows machines will lock out for 10 minutes after 10 failed login attempts within a 10 minute period. This thwarts brute force guessing of admin account passwords.You can adjust the lockout threshold, duration and reset values as desired for your environment. Just be sure to link the GPO containing the account lockout settings to your target OUs.How to Enable Administrator Account Lockout Policies in a Local Windows Computer?This how-to guide explains how to enable administrator account lockout policies to protect a local Windows computer from brute force password attacks.PrerequisitesAdministrative access to the local Windows computerSupported Windows 10 or Windows 11 operating systemsTime needed: 5 minutesStep-by-step procedure to Enable Administrator Account Lockout Policies in a Local Windows Computer.Launch Local Group Policy Editor1. Press Windows key + R to open the Run dialog box.2. Type gpedit.msc and click OK to launch the Local Group Policy Editor.Launch Local Group Policy Editor2. Navigate to Account Lockout PolicyExpand Computer Configuration > Windows Settings > Security Settings > Account Policies.2. Click on Account Lockout Policy.Navigate to Account Lockout Policy3. Configure Account Lockout ThresholdDouble click on Account lockout threshold and set it to the desired invalid logon attempts before the lockout. For example, set it to 10 attempts.Configure Account Lockout Threshold4. Configure Account Lockout DurationDouble-click on Account lockout duration and set it to the desired lockout duration in minutes once the threshold is reached. For example, set it to 10 minutes.Configure Account Lockout Duration5. Configure Reset Account Lockout CounterDouble click on Reset account lockout counter after and set it to the desired time period before failed logon counter resets to 0. For example, set it to 10 minutes.Configure Reset Account Lockout Counter6. Enable Allow Administrator Account LockoutDouble click on Allow administrator account lockout and set it to Enabled.Enable Allow Administrator Account Lockout7. Apply the New PolicyClose the Local Group Policy Editor. The account lockout policy changes take effect immediately.This covers the key steps to enable administrator account lockout policies on a local Windows computer to protect against brute force attacks. The lockout threshold, duration, and reset time can be adjusted as desired.Windows Editions That Support Account Lockout PoliciesThe account lockout policy security settings are supported on the following Windows client and server editions:Windows 11 Pro and EnterpriseWindows 10 Pro, Enterprise, EducationWindows Server 2022 and 2019Older Windows Pro and Enterprise versions still in supportThe administrator account lockout policies are enabled by default on new installations of:Windows 11 version 22H2Windows Server 2022Windows 10/Windows Server 2019 with Oct 2022 or later cumulative updates applied during initial setupFor existing Windows deployments, the account lockout policies must be manually configured via Group Policy as outlined above. This includes older Windows 10/Windows Server 2019 systems without the Oct 2022+ cumulative updates rolled out.So all supported Windows versions can utilize account lockout policies to deter brute force attacks. But only the very latest OS releases have brute force protections enabled out-of-the-box for new installations.Bottom LineLeft unprotected, Windows administrator accounts are prime targets for brute force password attacks. By leveraging new account lockout policy capabilities in Windows, organizations can significantly improve security against brute force credential guessing.For maximum protection, the account lockout policies should be configured via Group Policy on all compatible Windows versions. The optimal balance of security versus usability will determine the ideal failed attempt thresholds and lockout periods for a given environment.Combining account lockouts with long, complex admin account passwords makes brute forcing Windows systems extremely difficult. Attackers are blocked after just a few failed logon attempts. Account lockout policies greatly raise the bar for successfully guessing passwords via brute force.We hope this post helps you learn what is a Brute Force attack, how Microsoft addressed this problem, what Group Policies setting that protects a Windows PC from Brute Force Attacks, how to enable Administrator Account Lockout Policies, and ultimately, how you can protect your Windows PC from Brute Force Attacks using Group Policies.Visit our website, thesecmaster.com, and social media pages on Facebook, LinkedIn, Twitter, Telegram, Tumblr, & Medium and subscribe to receive updates like this.This post is originally published at thesecmaster.comWe thank everybody who has been supporting our work and request you check out thesecmaster.com for more such articles. Learning how attackers target weak domain account passwords is not enough for Active Directory security. Let's look beyond domain accounts and understand the ways adversaries attack local accounts on Windows servers and desktops. For this post, we will focus on the most important local account: Administrator. The Administrator account is built into every Windows operating system and provides full control over the system, including the ability to compromise domain accounts through Pass the Hash and Pass the Ticket attacks. The Administrator account is vulnerable to password attacks for two reasons: There is no lockout policy for the Administrator account. Microsoft notes that this makes the account "a prime target for brute-force, password-guessing attacks." Administrator accounts often share the same password, so if you can compromise one account, you can often reuse the password across other local accounts in the environment Let's walk through a typical attack against the Administrator account using a popular tool, CrackMapExec. Step 1. Guess the plaintext password using a brute force attack Because the Administrator account has no lockout policy, it is possible to make unlimited guesses of the account's password. Using password lists like the SecList collections, an adversary can craft a custom list of well-known passwords to use to try to log on using the Administrator account. To create a more targeted attack, they can enumerate the password policy on the target systems. This will reveal the minimum password length and password complexity settings, so they can limit their list to viable passwords only. Issuing this command against a member server or workstation will return local policy information. cme smb [hostname or list] –u [username] –p [password] –pass-pol Enumerating the local password policy options of a target server with CrackMapExec Once the adversary has a list of likely passwords, they can use the following command will run a brute-force attack against the local Administrator account, testing each password in turn: cme smb [hostname or list] –u Administrator –d builtin –p [password list] Brute-forcing the Administrator account using CrackMapExec Here you can see I clearly exceeded the local account lockout policy of 10 bad passwords but was still able to compromise the plaintext password of the Administrator account. Step 2. Use the password to spread laterally to other systems. With the password for one Administrator account in hand, adversaries may try using the same password on other systems in the environment. This strategy is often successful because it can be difficult to set and manage different passwords for the Administrator account on each endpoint. Therefore, attackers move laterally from the initial machine to a large number of machines very easily. Defense strategies Fortunately, there are several effective ways to protect against password attacks on local Administrator accounts. One option is to disable the account entirely and create a new administrative account in its place. Another strategy is to use Microsoft's Local Administrator Password Solution (LAPS) to automatically randomize the Administrator passwords across domain-joined computers and store the secrets centrally in Active Directory. This can guarantee that passwords are long and complex, and not reused across computers, which minimizes the risk of successful attacks. A third defense is to use Group Policy to deny network logon for all local Administrator accounts. This will help prevent password replay attacks from succeeding. How Netwrix can help Secure your Active Directory from end to end with the Netwrix Active Directory Security Solution. It will enable you to: Uncover security risks in Active Directory and prioritize your mitigation efforts. Harden security configurations across your IT infrastructure. Promptly detect and contain even advanced threats, such as DCSync , NTDS.dit extraction and Golden Ticket attacks. Respond to known threats instantly with automated response options. Minimize business disruptions with fast Active Directory recovery.