Курсов Проект

По

Софтуерни Технологии

Валентин Викторов Върбанов

ИСН 76, 471218011

1. Brute Force Attack
2. What is a Brute Force attack?

A brute force attack uses trial-and-error to guess login info, encryption keys, or find a hidden web page. Hackers work through all possible combinations hoping to guess correctly.

This is an old attack method, but it's still effective and popular with hackers. Because depending on the length and complexity of the password, cracking it can take anywhere from a few seconds to many years.

1. What do hackers gain from Brute Force Attacks?

* Profiting from ads or collecting activity data
* Stealing personal data and valuables
* Spreading malware to cause disruptions
* Hijacking your system for malicious activity
* Ruining a website’s reputation

1. Types of Brute Force Attack

* Simple brute force attacks

Hackers attempt to logically guess your credentials — completely unassisted from software tools or other means. These can reveal extremely simple passwords and PINs. For example, a password that is set as “guest12345”.

* Dictionary attacks

In a standard attack, a hacker chooses a target and runs possible passwords against that username. These are known as dictionary attacks. Dictionary attacks are the most basic tool in brute force attacks. Some hackers run through unabridged dictionaries and augment words with special characters and numerals or use special dictionaries of words, but this type of sequential attack is cumbersome.

* Hybrid brute force attacks

These hackers blend outside means with their logical guesses to attempt a break-in. A hybrid attack usually mixes dictionary and brute force attacks. These attacks are used to figure out combo passwords that mix common words with random characters. A brute force attack example of this nature would include passwords such as NewYork1993 or Spike1234.

* Reverse brute force attacks

Just as the name implies, a reverse brute force attack reverses the attack strategy by starting with a known password. Then hackers search millions of usernames until they find a match. Many of these criminals start with leaked passwords that are available online from existing data breaches.

* Credential stuffing

If a hacker has a username-password combo that works for one website, they’ll try it in tons of others as well. Since users have been known to reuse login info across many websites, they are the exclusive targets of an attack like this.

1. How to Protect passwords

* High encryption rates

To make it harder for brute force attacks to succeed, system administrators should ensure that passwords for their systems are encrypted with the highest encryption rates possible, such as 256-bit encryption. The more bits in the encryption scheme, the harder the password is to crack.

* Salt the hash

Administrators should also randomize password hashes by adding a random string of letters and numbers (called salt) to the password itself. This string should be stored in a separate database and retrieved and added to the password before it's hashed. By salting the hash, users with the same password have different hashes.

* Two-factor authentication (2FA)

Additionally, administrators can require two-step authentication and install an intrusion detection system that detects brute force attacks. This requires users to follow-up a login attempt with a second factor, like a physical USB key or fingerprint biometrics scan.

* Limit number of login re-tries

Limiting the number of attempts also reduces susceptibility to brute-force attacks. For example, allowing three attempts to enter the correct password before locking out the user for several minutes can cause significant delays and cause hackers to move on to easier targets.

* Account lockdown after excessive login attempts

If a hacker can endlessly keep retrying passwords even after a temporary lockout, they can return to try again. Locking the account and requiring the user to contact IT for an unlock will deter this activity. Short lockout timers are more convenient for users, but convenience can be a vulnerability. To balance this, you might consider using the long-term lockdown if there are excessive failed logins after the short one.

* Throttle rate of repeated logins

You can further slow an attacker’s efforts by creating space between each single login attempt. Once a login fails, a timer can deny login until a short amount of time has passed. This will leave lag-time for your real-time monitoring team to spot and work on stopping this threat. Some hackers might stop trying if the wait is not worth it.

* Required Captcha after repeated login attempts

Manual verification does stop robots from brute-forcing their way into your data. Captcha comes in many types, including retyping the text in an image, checking a checkbox, or identifying objects in pictures. Regardless of what you use, you can use this before the first login and after each failed attempt to protect further.

1. Sniffing, Man in the middle
2. What is packet sniffing?

A packet sniffer — also known as a packet analyzer, protocol analyzer or network analyzer — is a piece of hardware or software used to monitor network traffic. These packets are intended for — and addressed to — specific machines, but using a packet sniffer in "promiscuous mode" allows IT professionals, end users or malicious intruders to examine any packet, regardless of destination. It's possible to configure sniffers in two ways. The first is "unfiltered," meaning they will capture all packets possible and write them to a local hard drive for later examination. Next is "filtered" mode, meaning analyzers will only capture packets that contain specific data elements. Packet sniffers can be used on both wired and wireless networks — their efficacy depends on how much they are able to "see" as a result of network security protocols.

1. Risk Factors

Using a sniffer, it's possible to capture almost any information — for example, which websites that a user visits, what is viewed on the site, the contents and destination of any email along with details about any downloaded files. Protocol analyzers are often used by companies to keep track of network use by employees and are also a part of many reputable antivirus software packages. It's worth noting, however, that these analyzers can also be used for malicious purposes. If a user is convinced to download malware-laden email attachments or infected files from a website, it's possible for an unauthorized packet sniffer to be installed on a corporate network.

1. How to prevent sniffing?

Encrypt all the traffic, meaning use HTTPS. This will ensure that even if the traffic is being sniffed, the attacker will not be able to make sense of it. One thing here to be noted is that security work on defense in depth principle. Encrypting he data does not mean that now everything is safe. The attacker might be able to capture a lot of data and run crypto attacks to get something out of it. Use of secured protocols ensures that the traffic is encrypted and renders security for the traffic. Websites using HTTPS protocol are more secure than the ones that use HTTP – how is that achieved? Encryption.

If required, VPN (Virtual Private Networks) can be used to provide secure access to users.

1. DOS
2. What is a DoS attack?

A “denial of service” or DoS attack is used to tie up a website’s resources so that users who need to access the site cannot do so. Many major companies have been the focus of DoS attacks. Because a DoS attack can be easily engineered from nearly any location, finding those responsible can be extremely difficult.

Attackers include hacktivists (hackers whose activity is aimed at promoting a social or political cause), profit-motivated cybercriminals, and nation states.

1. Types of DoS attacks

* Flooding Attacks

Flooding is the more common form DoS attack. It occurs when the attacked system is overwhelmed by large amounts of traffic that the server is unable to handle. The system eventually stops.

* + ICMP flood

Also known as a ping flood — is a type of DoS attack that sends spoofed packets of information that hit every computer in a targeted network, taking advantage of misconfigured network devices.

* + SYN flood

A variation that exploits a vulnerability in the TCP connection sequence. This is often referred to as the three-way handshake connection with the host and the server. Here’s how it works:

The targeted server receives a request to begin the handshake. But, in a SYN flood, the handshake is never completed. That leaves the connected port as occupied and unavailable to process further requests. Meanwhile, the cybercriminal continues to send more and more requests, overwhelming all open ports and shutting down the server.

* Crash Attacks

Crash attacks occur less often, when cybercriminals transmit bugs that exploit flaws in the targeted system. The result? The system crashes.

* DDos Attack

Distributed denial of service (DDoS) attacks represent the next step in the evolution of DoS attacks as a way of disrupting the Internet. Cybercrimininals began using DDoS attacks around 2000. The attacks use large numbers of compromised computers, as well as other electronic devices — such as webcams and smart televisions that make up the ever-increasing Internet of Things — to force the shutdown of the targeted website, server or network.

* Slowloris

Slowloris is a highly-targeted attack, enabling one web server to take down another server, without affecting other services or ports on the target network. Slowloris does this by holding as many connections to the target web server open for as long as possible. It accomplishes this by creating connections to the target server, but sending only a partial request. Slowloris constantly sends more HTTP headers, but never completes a request. The targeted server keeps each of these false connections open. This eventually overflows the maximum concurrent connection pool, and leads to denial of additional connections from legitimate clients.

* Http Flood

In an HTTP flood DDoS attack, the attacker exploits seemingly-legitimate HTTP GET or POST requests to attack a web server or application. HTTP floods do not use malformed packets, spoofing or reflection techniques, and require less bandwidth than other attacks to bring down the targeted site or server. The attack is most effective when it forces the server or application to allocate the maximum resources possible in response to every single request.

1. How to prevent

* Dynamic IP restrictions

The Dynamic IP Restrictions Extension for IIS provides IT Professionals and Hosters a configurable module that helps mitigate or block Denial of Service Attacks or cracking of passwords through Brute-force by temporarily blocking Internet Protocol (IP) addresses of HTTP clients who follow a pattern that could be conducive to one of such attacks.

* Throttling

Limit a request to once in a set amount of seconds, thus limiting the possibility for DoS attacks. Based on IP from which the request was sent.

1. XSS and Script Upload
2. What is Cross Site Scripting

Cross site scripting is the injection of malicious code in a web application, usually, Javascript but could also be CSS or HTML. When attackers manage to inject code into your web application, this code often gets also saved in a database. This means every user could be affected by this. For example, if an attacker manages to inject Javascript into the product name on Amazon. Every user who opens the infected product would load the malicious code.

1. Consequences of XSS Attacks
   * Attackers could read your cookies and therefore gain access to your private accounts like social media or bank.
   * Users may be redirected to malicious sites.
   * Attackers could modify the layout of the website to lure users into unintentional actions.
   * Users could be annoyed which will lead to damage to your reputation and probably a loss of revenue.
   * Often used in combination with other attacks like cross site request forgery (CSRF).
2. Best Practices against Cross Site Scripting Attacks
   * Validate every user input, either reject or sanitize unknown character, for example, < or > which can be used to create.
   * Test every input from an external source.
   * Use HttpOnly for cookies so it is not readable by Javascript (therefore an attacker can’t use Javascript to read your cookies).
   * Use markdown instead of HTML editors.
3. Cross Site Scripting in ASP .NET Core

ASP .NET Core Is already pretty safe out of the box due to automatically encoding HTML, for example < gets encoded into &lt. As long as you are using the recommended way to access, input and show information, you should be safe.

1. Ransomeware
2. What is Ransomware?

Ransomware is a type of malware from cryptovirology that threatens to publish the victim's data or perpetually block access to it unless a ransom is paid. While some simple ransomware may lock the system so that it is not difficult for a knowledgeable person to reverse, more advanced malware uses a technique called cryptoviral extortion. It encrypts the victim's files, making them inaccessible, and demands a ransom payment to decrypt them. In a properly implemented cryptoviral extortion attack, recovering the files without the decryption key is an intractable problem – and difficult to trace digital currencies such as paysafecard or Bitcoin and other cryptocurrencies are used for the ransoms, making tracing and prosecuting the perpetrators difficult. Ransomware attacks are typically carried out using a Trojan disguised as a legitimate file that the user is tricked into downloading or opening when it arrives as an email attachment. However, one high-profile example, the WannaCry worm, traveled automatically between computers without user interaction.

1. Types of Ransomware

* Encrypting ransomware

The first known malware extortion attack, the "AIDS Trojan" written by Joseph Popp in 1989, had a design failure so severe it was not necessary to pay the extortionist at all. Its payload hid the files on the hard drive and encrypted only their names.

* Non-encrypting Ransomware

In August 2010, Russian authorities arrested nine individuals connected to a ransomware Trojan known as WinLock. WinLock did not use encryption. Instead, WinLock restricted access to the system, asking users to pay, to receive a code that could unlock the machine.

* Exfiltration (Leakware / Doxware)

The converse of ransomware is a cryptovirology attack invented by Adam L. Young that threatens to publish stolen information from the victim's computer system rather than deny the victim access to it. In a leakware attack, malware exfiltrates sensitive host data either to the attacker or alternatively, to remote instances of the malware, and the attacker threatens to publish the victim's data unless a ransom is paid.

* Mobile ransomware

With the increased popularity of ransomware on PC platforms, ransomware targeting mobile operating systems has also proliferated. Typically, mobile ransomware payloads are blockers, as there is little incentive to encrypt data since it can be easily restored via online synchronization.

1. Mitigation of Ransomware

Using software or other security policies to block known payloads from launching will help to prevent infection, but will not protect against all attacks. As such, having a proper backup solution is a critical component to defending against ransomware. Note that, because many ransomware attackers will not only encrypt the victim's live machine but it will also attempt to delete any hot backups stored locally or on accessible over the network on a NAS, it's also critical to maintain "offline" backups of data. Moreover, if using a NAS or Cloud storage, then the computer should have append-only permission to the destination storage, such that it cannot delete or overwrite previous backups. Installing security updates issued by software vendors can mitigate the vulnerabilities leveraged by certain strains to propagate. Also important is keeping critical computers isolated from networks.

1. SQL Injection
2. What is an SQL Injection?

SQL injection is one of the most common web hacking techniques, potentially able to destroy your database. SQL injection is the placement of malicious code in SQL statements, via web page input. SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

Most common SQL Injection types:

* + Incorrectly filtered Data

This form of injection occurs when user input is not filtered for escape characters and is then passed into an SQL statement. This results in the potential manipulation of the statements performed on the database by the end-user of the application.



* + Blind SQL injection

Blind SQL injection is used when a web application is vulnerable to an SQL injection but the results of the injection are not visible to the attacker. The page with the vulnerability may not be one that displays data but will display differently depending on the results of a logical statement injected into the legitimate SQL statement called for that page.



* + Second order SQL injection

Second order SQL injection occurs when submitted values contain malicious commands that are stored rather than executed immediately. In some cases, the application may correctly encode an SQL statement and store it as valid SQL. Then, another part of that application without controls to protect against SQL injection might execute that stored SQL statement.

1. How to protect against SQL Injection

Checking every query made to the back end, before it is run as a SQL query is the best way to defend your app.

Escaping every special and SQL character is one way to defend against Injections.

Another good way is pattern check. Adding pattern check to strings that we expect to be a certain way every time good be a good defense, for example, adding pattern to a date field returning a string, or a name field expecting only a-z and A-Z.

Most common web languages already check for those queries, when used correctly.

For example, ASP.NET has a smart parameterized way of executing SQL queries, checking each parameter to ensure that it is correct for its column and are treated literally, and not as part of the SQL to be executed.

1. Cross site request forgery
2. What is CSRF?

An attack that forces an end user to execute unwanted actions on a web application in which they’re currently authenticated. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker’s choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.

1. Types of CSRF Attacks
   * GET scenario

The exploit URL can be disguised as an ordinary link, encouraging the victim to click it, sending the GET forged by the hacker.

* + POST scenario

The only difference between GET and POST attacks is how the attack is being executed by the victim. Such a request cannot be delivered using standard A or IMG tags, but can be delivered using a FORM tags. This form will require the user to click on the submit button, but this can be also executed automatically using JavaScript

* + Other HTTP methods

Modern web application APIs frequently use other HTTP methods, such as PUT or DELETE. Let’s assume the vulnerable bank uses PUT that takes a JSON block as an argument. Fortunately, this request will not be executed by modern web browsers thanks to same-origin policy restrictions. This restriction is enabled by default unless the target web site explicitly opens up cross-origin requests from the attacker’s (or everyone’s) origin by using CORS with the following header.

1. Defend against CSRF

As said above modern browsers already have some level of defense against CSRF by using CORS policy. Many modern frameworks or languages have implemented other types of defenses.

For example, ASP.NET has implemented its own Anti Forgery Token. This method adds the hidden form field and also sets the cookie token.

Most CSRF attacks are based on social engineering and therefore if the site exposes as little to no data the hacker will have a hard time to replicate the POST or GET requests.

Гитхъб с имплементацията на защити: https://github.com/Vvarbanov/RealEstateManager/tree/project