**brute-force-attacker:**

**risker och lösningar i lösernordssäkerhet**

IT-Ingenjör

Cybersäkerhet & Digital Infrastruktur

Beruk Gebru

Marc Söder

Raman Rostam

**Program:** IT-Ingenjör: Cybersäkerhet & Digital Infrastruktur

**Svensk titel:** Brute-force-attacker: Risker och lösningar i lösenordssäkerhet

**Engelsk titel:** Brute-force-attacks: Risks and solutions in password security

**Författare:** Beruk Gebru, Marc Söder & Raman Rostam

**Nyckelord:** Lösenord, Brute-force, Hydra, Cybersäkerhet

# Sammanfattning

Den här rapporten undersöker effektiviteten hos olika lösenordsskyddsmekanismer i att motverka brute-force-attacker genom en rad kontrollerade experiment, som genomfördes i en säker labbmiljö. Attackerna utfördes från en Kali Linux-maskin, och de testade inloggningssidorna var placerade på en dedikerad webbserver. Säkerhetsåtgärder som utvärderades inkluderade ingen skyddsåtgärd alls, sessionsbaserade begränsningar, databasbaserade skydd, samt IP-adressbaserad blockering. Dessutom testades säkerheten för hashade lösenord med hjälp av verktyget Hashcat.

Resultaten visade att inloggningssidor utan begränsningar var mycket sårbara för brute-force-attacker, där verktyg som Hydra kunde testa tusentals lösenord på kort tid. Även om både sessionsbaserade och IP-adressbaserade skydd lyckades bromsa attackerna något, kunde angripare ändå kringgå dessa åtgärder genom att använda nya sessioner eller IP-hoppning. De databasbaserade begränsningarna visade sig vara den mest effektiva lösningen och förhindrade alla försök till lösenordsknäckning. Dock medförde även denna metod vissa tillgänglighetsproblem, då legitima användare kunde blockeras efter upprepade felaktiga inloggningsförsök.

Denna studie understryker vikten av att kombinera starka lösenord med robusta säkerhetsmekanismer, såsom skydd på databasnivå multifaktorautentisering (MFA) och CAPTCHA, för att stärka säkerheten utan att äventyra användarens åtkomst. Rapporten lyfter också fram hur dessa resultat kopplar till CIA-triaden (Konfidentialitet, Integritet och Tillgänglighet) och betonar hur bristfälliga säkerhetslösningar kan utsätta företag för allvarliga operativa och ryktesmässiga risker. Ur ett perspektiv av ekonomisk hållbarhet riskerar företag som misslyckas med att implementera tillräckliga säkerhetsåtgärder stora ekonomiska förluster till följd av komprometterade konton, skador på sitt varumärke och störningar i affärsverksamheten.

# Abstract

This report explores the effectiveness of various password protection mechanisms in defending against brute-force attacks. A series of controlled experiments were conducted in a safe lab environment, where the attacks were launched from a Kali Linux machine, and the login pages were hosted on a dedicated web server. The security implementations tested included no protection, session-based limitations, database-based restrictions, and IP-based blocking. Additionally, the security of hashed passwords was tested using the Hashcat tool.

The experiments demonstrated that login pages without limitations are highly vulnerable, allowing brute-force tools like Hydra to test thousands of passwords in a short time. While session-based and IP-based protections slowed down attacks, they were not entirely effective, as attackers could bypass them using new sessions or IP-hopping techniques. Database-based restrictions proved to be the most effective, preventing any successful brute-force attempts. However, these methods also introduced availability issues, potentially blocking legitimate users after several failed login attempts.

The findings underline the importance of combining solid passwords with robust security mechanisms, such as database-level protection, multi-factor authentication (MFA) and CAPTCHA, to improve security without compromising user access. This report also highlights the relevance of these results to the CIA triad (Confidentiality, Integrity, and Availability), emphasizing how poorly implemented security can expose companies to severe reputational and operational risks. From an economic sustainability perspective, companies failing to implement adequate security measures face potential financial losses due to compromised accounts, damage to their reputation, and disrupted business operations.

Innehållsförteckning

[Sammanfattning i](#_Toc180415807)

[Abstract ii](#_Toc180415808)

[1. INLEDNING 1](#_Toc180415809)

[1.1 Bakgrund 1](#_Toc180415810)

[1.2 Forskningsfrågor 2](#_Toc180415811)

[1.3 Syfte 2](#_Toc180415812)

[1.4 Avgränsningar 2](#_Toc180415813)

[2. Litteraturgenomgång / TEORI 3](#_Toc180415814)

[2.1 Brute-Force. 3](#_Toc180415815)

[2.2 Teorier från CIA-triangeln och dess tillämpningar 5](#_Toc180415816)

[2.3 Ekonomisk Hållbarhet 5](#_Toc180415817)

[3. METOD 7](#_Toc180415818)

[3.1 Förberedelse av labbmiljö 7](#_Toc180415819)

[3.2 Genomförande av brute-force-attack 8](#_Toc180415820)

[3.3 Datainsamling och Analys 8](#_Toc180415821)

[4. RESULTAT 10](#_Toc180415822)

[4.1 Empiri på utfört experiment 10](#_Toc180415823)

[4.1.1 Sessionsbaserad inloggningssida utan lösenordsbegränsningar 10](#_Toc180415824)

[4.1.2 Sessionsbaserad med lösenordsbegränsningar. 11](#_Toc180415825)

[4.1.3 Databas-baserade inloggningsbegränsningar 11](#_Toc180415826)

[4.1.4 IP-adressbaserade inloggningsbegränsningar 12](#_Toc180415827)

[4.1.5 Hashade lösenord 14](#_Toc180415828)

[4.2 Att mitigera säkerhetsbrister 15](#_Toc180415829)

[4.2.1 Kombination av säkerhetsåtgärder 15](#_Toc180415830)

[4.3 Ekonomisk hållbarhet 16](#_Toc180415831)

[5. Analys 18](#_Toc180415832)

[6. REFERENSER 19](#_Toc180415833)

**Figurförteckning**

[Figur 2.1.1 Processen för hashing 5](#_Toc181190726)

[Figur 4.1.1 Hydra mot oskyddad inloggninssida 11](#_Toc181190727)

[Figur 4.1.2 Hydra mot sessionsbaserad säkerhetsåtgärd 12](#_Toc181190728)

[Figur 4.1.3 Hydra mot databasbaserad säkerhetsåtgärd 13](#_Toc181190729)

[Figur 4.1.4 Lösenordslista mot IP-adressbaserad skyddsåtgärd 14](#_Toc181190730)

[Figur 4.1.5 Lyckad attack mot IP-adressbaserad skyddsåtgärd 14](#_Toc181190731)

[Figur 4.1.6 Hashcat mot Shein 15](#_Toc181190732)

[Figur 4.1.7 Knäckta hashade lösenord 15](#_Toc181190733)

[Figur 4.1.8 Hashcat med fel hashalgoritm 15](#_Toc181190734)

[Figur 4.1.9 Hashcat mot RockYou 16](#_Toc181190735)

[Figur 4.2.1 Användarnamn och lösenord för inloggningssidan 17](#_Toc181190736)

**Tabellförteckning**

[Tabell 1 Olika login och säkerhetsåtgärder 9](#_Toc181190737)

# INLEDNING

## Bakgrund

Genom tiderna så har det alltid funnits lösenord, lösenord som används för att hålla obehöriga utanför. Det kan vara allt ifrån ett lösenord som förhindrar någon att gå in på en klubb till lösenord på ett bankkonto. Det första digitala lösenordet kommer från datavetenskapsprofessorn Fernando Corbato som 1961 skapade det första digitala lösenordet. Han gjorde detta för att han byggde en tidsdelningsdator som flera personer skulle använda, så varje användare fick ett eget lösenord (Dashlane u. å.).

Efter en tid insåg man att åtkomst till en användares lösenord även medförde åtkomst till deras konto. Inledningsvis var detta mer av ett gissningsspel, där man helt enkelt prövade ett stort antal lösenord tills man hittade det rätta. Detta utgör den första formen av en brute-force-attack, där man systematiskt testar olika lösenordskombinationer tills rätt kombination identifieras.

Problematiken kring lösenord ligger inte i själva lösenorden, utan hos människan bakom dem. Människor har en begränsad förmåga att komma ihåg ett stort antal lösenord, och när lösenorden blir många och komplexa kopplas de till personliga referenser. När lösenord är baserade på personlig information blir de dock enklare för angripare att gissa. Detta medför en ökad säkerhetsrisk, då en angripare som får tillgång till ett lösenord, exempelvis genom en e-postadress, kan använda samma lösenord för att få åtkomst till andra tjänster. Sannolikheten är då hög att angriparen kan kompromettera flera konton.

När lösenord kopplas till personliga tillhörigheter blir det lättare för angripare att lista ut dem, eftersom de baseras på något privat. Detta skapar ett problem då man ofta väljer enkla lösenord för att lättare komma ihåg dem, och det är vanligt att samma lösenord används på flera olika webbplatser. Detta ökar risken, eftersom om en angripare får tag på en användares e-postadress och knäcker ett lösenord, kan de enkelt testa det på andra sidor. Sannolikheten är då stor att hackern får tillgång till flera tjänster.

År 2009 blev företaget RockYou utsatt för en hackerattack där angriparna fick tillgång till 32 miljoner konton varav 14 miljoner lösenord som sparats. Problemet var att lösenorden var sparade i klartext, alltså okrypterade, vilket gjorde det enkelt för angriparna att skapa en ordlista baserad på dessa lösenord, som de sparade under namnet RockYou.txt. Idag är RockYou.txt en standardordlista som finns förinstallerad på Linux och används av både whitehat- och blackhat-hackare (Lurey 2023).

Behovet av hashning uppstod när det blev uppenbart att lagra lösenord i klartext var alltför riskabelt. När databaser blev utsatta för intrång kunde angripare enkelt läsa användares lösenord och orsaka allvarliga säkerhetsincidenter. Hashning introducerades som en lösning, där lösenord omvandlas till en unik sträng av siffror och bokstäver, kallad en hashsumma, som inte kan återställas till det ursprungliga lösenordet. Även om en angripare får tillgång till en databas med hashade lösenord, kan de inte enkelt dechiffrera dem. Hashning blev därmed ett centralt verktyg för att skydda lösenord och minska risken för dataintrång (Open Web Application Security Project [OWASP] u.å).

I Linux-operativsystemet finns program utformade för att hjälpa till att knäcka lösenord. Om man får tag på ett hashat lösenord kan man använda RockYou.txt tillsammans med program som Hydra eller Hashcat. Dessa program fungerar genom att ta det första lösenordet från ordlistan i RockYou.txt, kryptera det och sedan jämföra det med den hashade text man har. Detta upprepas tills en match hittas eller tills alla 14 miljoner lösenord har testats. Processen kan ta tid, vilket är anledningen till att många webbplatser kräver användning av specialtecken och liknande för att försvåra lösenordsknäckning (Hashcat u.å.).

## Forskningsfrågor

* Hur utförs en brute-force-attack mot en hemsida och vilka tekniska faktorer påverkar dess effektivitet?
* Hur kan starka lösenord kombineras med andra skyddsmekanismer för optimal säkerhet?
* Hur successiva brute-force-attacker leda till avtagande ekonomisk hållbarhet?

## Syfte

Syftet med denna rapport är att undersöka hur brute-force-attacker fungerar, belysa hur lösenord kan vara sårbara för dessa typer av attacker, samt identifiera och utvärdera effektiva metoder för att säkra lösenord. Genom att analysera de tekniker som används för att knäcka lösenord och utforska skyddsåtgärder. Rapporten syftar till att få en djupare förståelse av strategier för att minimera riskerna för lösenordsbaserade attacker.

Vidare syftar rapporten till att utvärdera hur komplexiteten i lösenord och implementeringen av säkerhetsrutiner kan främja för långsiktig cybersäkerhet och ekonomisk hållbarhet. I en tid där cybersäkerhet utgör en kritisk komponent för företags framgång, är det av yttersta vikt att identifiera och implementera effektiva skyddsåtgärder. Detta inte bara för att skydda mot potentiella cyberattacker, som kan resultera i betydande ekonomiska förluster, utan också för att säkerställa en hållbar affärsmiljö där företag kan verka utan att riskera sin integritet.

## Avgränsningar

Rapporten kommer att fokusera på den ekonomiska hållbarheten kopplad till brute-force-attacker. Aspekter som rör social eller ekologisk hållbarhet kommer inte att behandlas, då fokus ligger på att analysera de ekonomiska konsekvenserna och riskerna relaterade till sådana säkerhetshot.

Dessutom kommer rapporten inte att undersöka sårbarheter hos molntjänster i relation till brute-force-attacker. Även om molntjänster är relevanta för företag när det gäller dessa attacker och deras ekonomiska hållbarhet, så utgör detta ämne en större komplexitet som skulle kräva en djupare analys.

# Litteraturgenomgång / TEORI

## Brute-Force.

Brute-force-attacker kan genomföras och påverkas på olika sätt, beroende på hur robusta de säkerhetsåtgärder är som vidtagits för att motverka attacken, samt vilka metoder angriparen väljer för att maximera sin chans att lyckas. Det är av stor betydelse att förstå vilka faktorer som påverkar en brute-force-attacks effektivitet. Utan en sådan förståelse blir det svårt att utveckla effektiva skydd mot dessa typer av angrepp. Om man däremot vet hur attacken fungerar, så går det också att förutse angriparens handlingsmönster och identifiera de sårbarheter som är mest kritiska. Detta är avgörande för att kunna formulera effektiva försvarsstrategier. Sun Tzu uttryckte en liknande tanke i The Art of War: "If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle." (Sun Tzu, 2005, s. 43).

Brute force-attacker kan utföras med hjälp av flera verktyg, inklusive Hydra, Medusa och Ncrack, som är särskilt utformade för att knäcka lösenordskyddade system genom att testa olika kombinationer av användarnamn och lösenord. Hydra är ett verktyg för att genomföra brute force-attacker, på grund av dess flexibilitet och effektivitet vid stöd av olika autentiseringsprotokoll (VarshaGrover, Gagandeep, 2020).

För att förhindra brute-force-attacker och andra former av obehörig autentisering finns det ett antal centrala säkerhetsåtgärder som rekommenderas. En av de mest effektiva åtgärderna är att implementera multifaktorsautentisering (MFA), som kraftigt minskar risken för att en angripare lyckas genom att kräva mer än bara ett lösenord för att logga in. Genom att använda en extra autentiseringsmetod, som en SMS-kod eller en autentiseringsapp, kan legitima användare bekräfta sin identitet, vilket gör det betydligt svårare för angripare att få obehörig åtkomst (Vugdelija, Nedeljković, Kojić, Lukić & Vesić 2021).

Lösenordspolicys spelar också en viktig roll i att förhindra att användare väljer svaga eller lätt gissade lösenord. Ett sätt att stärka detta är genom att använda en lista över de 10,000 vanligaste lösenorden och aktivt blockera användning av dessa, samtidigt som minsta lösenordslängd säkerställs. Dessutom kan användningen av lösenordsgeneratorer uppmuntras för att hjälpa användare att skapa starkare lösenord. Vidare rekommenderas det att begränsa eller sakta ner antalet felaktiga inloggningsförsök och logga dem alla för att kunna upptäcka och förhindra systematiska attacker (Vugdelija et al. 2021).

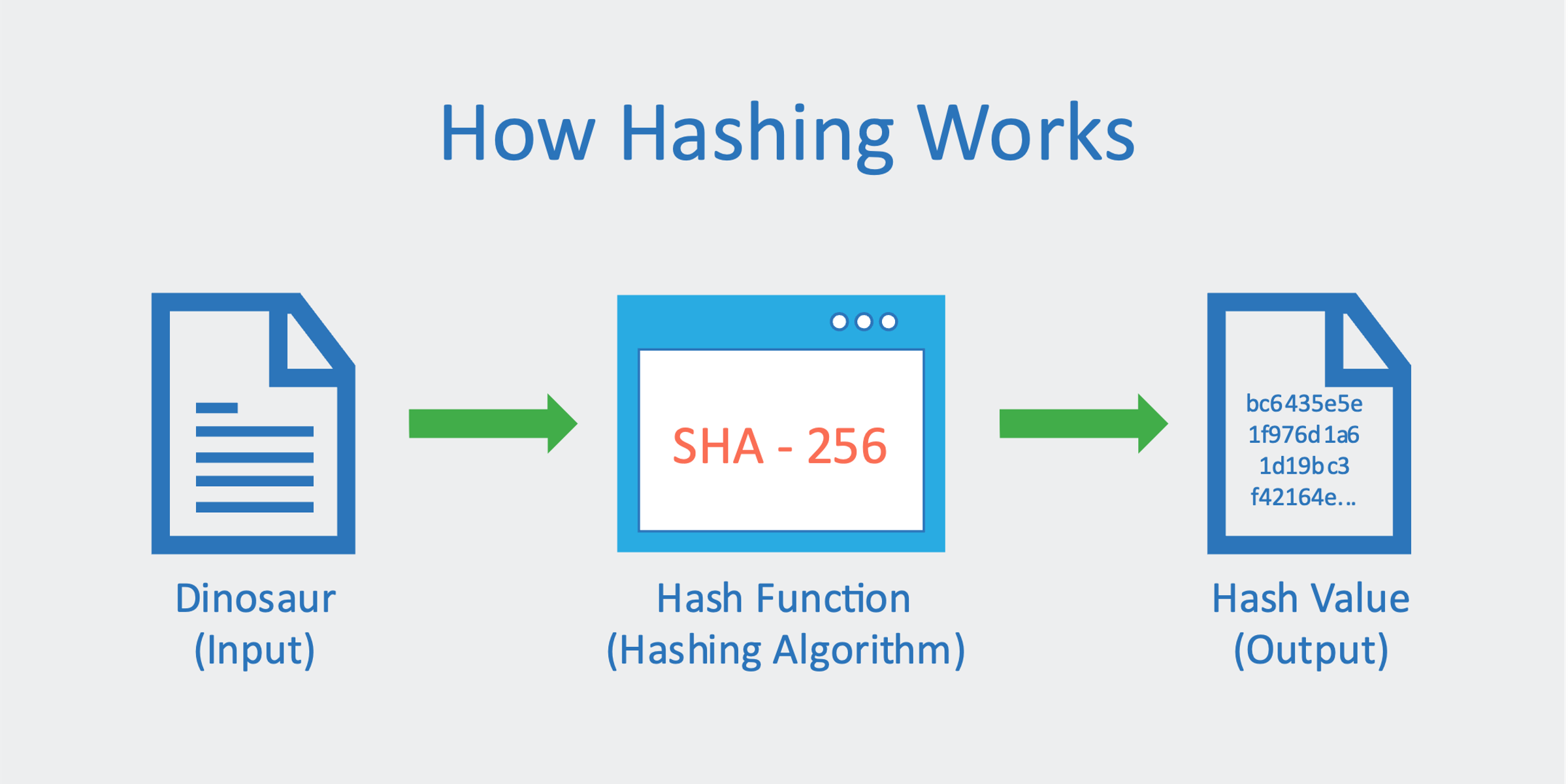
Kontolåsning efter ett antal felaktiga lösenordsförsök är en vanlig metod för att skydda mot brute-force-attacker. Denna åtgärd kan dock missbrukas av angripare som medvetet låser användarkonton, vilket kan leda till en Denial of Service (DoS)-attack där legitima användare hindras från att logga in. Dessutom är kontolåsning ineffektiv mot långsamma attacker som testar lösenord över längre tidsintervall eller attacker som riktar sig mot flera användarnamn med ett enstaka lösenord. Allmänt så är detta en bra metod för att försvara mot brute-force-attacker, speciellt i en miljö där prioriteringen av intrång är betydligt högre än att skydda sig mot DoS attacker (Vugdelija et al. 2021).

CAPTCHA står för "Completely Automated Public Turing test to tell Computers and Humans Apart" och är en säkerhetsmetod som används för att skilja mellan mänskliga användare och automatiserade system, eller bots. En bot är ett datorprogram som automatiskt utför uppgifter, oftast utan mänsklig inblandning. En CAPTCHA består vanligtvis av en uppgift som är lätt för en människa att lösa men svår eller omöjlig för en dator, till exempel att identifiera förvrängda bokstäver eller bilder. Vid brute-force-attacker, där en angripare automatiskt försöker gissa inloggningsuppgifter genom att testa många olika kombinationer, kan CAPTCHA spela en viktig roll i att förhindra framgångsrika attacker. Eftersom en dator inte kan lösa CAPTCHA-uppgifter lika effektivt som en människa, hindras attacker från att automatiseras i samma omfattning (Vugdelija et al. 2021).

En ytterligare metod är att implementera säkerhetsfrågor, där användare måste svara på en eller flera frågor för att återfå åtkomst till sina konton. Dessa frågor bör vara svåra för andra att gissa men lätta för användaren att komma ihåg. Detta kan fungera som ett extra skydd vid sidan om kontolåsning och MFA.

Att blockera inloggningsförsök baserat på IP-adresser är en effektiv metod för att förhindra en angripare från att fortsätta sina attacker genom att neka åtkomst från den specifika IP-adressen. Denna strategi kan också kombineras med geolocation-teknik för att begränsa åtkomst baserat på användarens fysiska plats, vilket ytterligare minskar risken för obehörig åtkomst. Trots dess effektivitet kräver denna metod en viss teknisk kompetens, vilket kan utgöra en utmaning för vissa användare eller administratörer. En nackdel är risken för överblockering, särskilt i miljöer där många användare delar en och samma publika IP-adress, exempelvis i stora företag eller nätverk (OWASP u.å). Detta kan leda till att legitima användare oavsiktligt stängs ute från systemet. En ytterligare nackdel är att angriparen kan kringgå detta skydd genom att använda en VPN, vilket möjliggör byte av IP-adress. Trots detta kan en ökad aktivitet från flera olika IP-adresser som försöker få tillgång till samma konto ändå vara en indikator på en potentiell brute-force-attack (Vugdelija et al. 2021).

Ytterligare har hashing algoritmer använts för att kryptera ens lösenord genom att använda hashfunktioner som omvandlar lösenord oavsett storlek till en fast längd av sträng (se Figur 2.1.1). I figuren visas hur det klartextiga ordet "Dinosaur" omvandlas av hashfunktionen med SHA-256 algoritmen till ett hashvärde som knappt går att läsa av. Denna hashing processen som ofta används inom datorsäkerhet och lösenordsammanhang är enkelriktade process, vilket innebär att det inte går få tillbaka original datan från hashen. Det finns många olika hashingsalgoritmer och varje algoritm som exempelvis MD5 eller SHA-256 har sin egen struktur och längd på konverterade lösenord som ofta kallas för Hash (Code Signing u.å.).



Figur . Processen för hashing

Inom hasning så finns det även en process som kallas för ”Hashing scheme” som involverar tre centrala komponenter: en hashfunktion, antalet iterationer, samt en salt. Salt är en säkerhetsåtgärd som används vid hashning av lösenord för att ytterligare skydda mot brute-force-attacker. Genom att införa en slumpmässig sträng, kallad salt, tillsammans med lösenordet innan hashning, blir det betydligt svårare för angripare att använda förberäknade tabeller (rainbow tables) för att knäcka lösenord. Om samma lösenord används av flera användare kommer varje användares hashvärde att vara unikt på grund av det olika saltet. Detta gör att även om en angripare lyckas få tag på en hash av ett lösenord, måste de fortfarande känna till det specifika saltet för att kunna återfå det ursprungliga lösenordet, vilket skyddar mot attacker där flera lösenord kan knäckas på en gång (Ntantogian, Stefanos & Xenakis 2019).

Hashcat kommer att användas för att få fram lösenord från hashad text, detta på grund av att Hashcat är det snabbaste verktyget för detta ändamål. Program som Hashcat är utvecklat för att underlätta återställning av lösenord som har lagrats i hashad form. Detta uppnås vanligtvis genom användning av en ordlista innehållande tusentals ”vanligt använda” lösenord, vilka sedan hashas med samma algoritm som det hashade lösenordet. Ett korrekt gissat lösenord att resultera i ett identiskt hashvärde, vilket möjliggör identifiering av det ursprungliga lösenordet (Hashcat u.å.).

## Teorier från CIA-triangeln och dess tillämpningar

Inom cybersäkerhet används ofta CIA-triangeln (Confidentiality, Integrity, Availability) som ett ramverk för att förstå hur system och data bör skyddas. Konfidentialitet innebär att endast behöriga användare har tillgång till information, integritet säkerställer att informationen är korrekt och inte har manipulerats, och tillgänglighet innebär att system och data alltid ska vara åtkomliga för de som har rätt att använda dem (Covert, Francis, Steinhagen & Streff 2020).

NIST, eller National Institute of Standards and Technology, är en amerikansk federal myndighet som utvecklar standarder, riktlinjer och bästa praxis inom olika områden, inklusive cybersäkerhet. NIST använder CIA-triangeln som en grundläggande komponent i sitt säkerhetsramverk. Deras huvudsakliga mål är att säkerställa att system uppfyller de tre viktiga säkerhetsprinciperna: konfidentialitet, integritet och tillgänglighet. NIST ramverk är praktiskt orienterat och fokuserar på att implementera både tekniska och administrativa åtgärder för att hantera hot, som till exempel brute-force-attacker. Detta kan omfatta åtgärder som att begränsa antalet inloggningsförsök, använda robusta algoritmer för lösenordshantering, samt övervaka och logga misstänkta aktiviteter för att skydda systemets konfidentialitet och säkerställa att det förblir tillgängligt för legitima användare. Genom dessa strategier kan säkerheten i systemet stärkas och risken för obehörig åtkomst, såsom brute-force-attacker, minskas (Covert et al. 2020).

ULD (Unabhängiges Landeszentrum für Datenschutz) å andra sidan, integrerar CIA-principerna i sitt utvidgade säkerhetskoncept, känt som "Six-Pointed Star"-modellen. Denna modell går utöver de traditionella målen om konfidentialitet, integritet och tillgänglighet genom att lägga större fokus på användares rättigheter och dataskyddsaspekter. ULD ramverk prioriterar dessutom transparens, möjligheten för användare att ingripa i hanteringen av sina egna data (intervenability) och skyddet mot att data kan kopplas till andra sammanhang utan användarens medgivande (unlinkability). Detta innebär att ULD strävar efter att ge individer mer kontroll över hur deras personliga information används och skyddas, vilket skiljer sig från NIST mer praktiska angreppssätt (Covert et al. 2020).

## Ekonomisk Hållbarhet

Förmågan att skydda sig mot brute-force-attacker är en avgörande aspekt för att säkerställa ekonomisk hållbarhet inom en organisation. Konsekvenserna av sådana attacker kan vara betydligt mer omfattande än vad som ofta antas, eftersom det inte enbart är enskilda individer som drabbas. Medan brute-force-attacker ofta ses som försök att få åtkomst till personliga konton, utgör de även ett stort hot mot organisationer och företag. Om en angripare till exempel får åtkomst till ett administratörskonto, kan de potentiellt orsaka betydande skador genom att radera kritisk data, manipulera systeminställningar eller läcka känslig information. Dessa typer av incidenter kan resultera i betydande ekonomiska förluster för organisationen, både direkt genom förlorade tillgångar och indirekt genom kostnader relaterade till återställning av system, skadat anseende och potentiella rättsliga påföljder. Därför är ett starkt skydd mot brute-force-attacker en viktig investering för att undvika kostsamma avbrott och långsiktiga ekonomiska förluster.

Cybersäkerhet är således en avgörande komponent för ekonomisk hållbarhet, där cyberattacker, såsom brute-force-attacker, medför betydande ekonomiska förluster för både individer och organisationer. Dessa incidenter underminerar förtroendet för de drabbade organisationerna, vilket i sin tur påverkar kundernas tillit negativt. När kunder blir påverkade av sådana attacker till följd av bristande säkerhetsåtgärder hos företaget, riskerar de att tappa förtroendet för organisationens förmåga att skydda deras data och integritet. När kunder påverkas av dessa attacker till följd av bristande säkerhetsåtgärder inom företaget, minskar sannolikheten för att de fortsätter som kunder, vilket ytterligare förvärrar de ekonomiska förlusterna i takt med att fler kunder lämnar organisationen (Vasiu, I. & Vasiu, L. 2018).

I dagens samhälle sker kundköp inte längre enbart i fysiska miljöer, utan e-handel har blivit en central del av den globala ekonomin. E-handel har visat sig vara en viktig drivkraft för ekonomisk tillväxt genom att skapa möjligheter för såväl konsumenter som företag och länder. Genom e-handel kan varor beställas, produkter exporteras och stora inköp göras snabbare och effektivare än någonsin tidigare. Denna utveckling har särskilt gynnat små och medelstora företag (SME), som får tillgång till nya marknader och tillväxtmöjligheter (Vasiu, I. & Vasiu, L. 2018).

Samtidigt medför denna digitala tillväxt också ökade risker, särskilt när det gäller cyberattacker som brute-force-attacker. SME är ofta mer utsatta, eftersom de inte har samma resurser som större företag att investera i robust cybersäkerhet för sin e-handelsverksamhet. Denna sårbarhet ökar risken för betydande ekonomiska förluster och höga kostnader för att återhämta sig från cyberincidenter, vilket negativt påverkar den ekonomiska hållbarheten i längden (Vasiu, I. & Vasiu, L. 2018).

# METOD

## Förberedelse av labbmiljö

Labbmiljön bestod av två datorer där en av dem innehåller en Kali Linux-maskin. Datorerna och servern konfigurerades enligt följande:

* **Dator specifikationer:**
* **CPU**: Intel Core i5 12500
* **RAM**: 32GB 2394MHz
* **Moderkort**: HP 8957 (U3E1)
* **Skärm**: HP 840 G9 AiO (1920x1080@60Hz)
* **GPU**: 4095MB NVIDIA GeForce RTX 3050 Ti Laptop GPU (HP)
* **SSD**: 476GB SK Hynix BC711
* **Operativ System (OS)**: Windows Server 2019 Datacenter 64-bit

**Webbserver:**

En dedikerad dator kommer att konfigureras som webbserver med hjälp av XAMPP, där flera inloggningssidor med avsiktliga säkerhetsbrister mot brute-force-attacker etableras, inklusive svaga lösenord. XAMPP valdes för sin enkelhet och flexibilitet, vilket möjliggör snabb uppsättning av en lokal servermiljö. Med stöd för MySQL förenklar det skapandet av databaser, vilket gör det till ett idealiskt verktyg för att testa och analysera olika säkerhetsmekanismer utan att behöva hantera komplex serveradministration (XAMMP u.å.).

1. Den första inloggningssidan kommer inte att implementera några skydd mot brute-force-attacker. Inga timeout-mekanismer eller spärrar kommer att finnas, vilket möjliggör obegränsade inloggningsförsök. Detta för att bevisa konsekvenser om säkerhetsåtgärder ignoreras.
2. Den andra inloggningssidan kommer att ha ett skydd genom en 5-minuters lockout på sessionsnivå efter tre misslyckade inloggningsförsök. Detta skydd kommer att vara per användarsession och blockera endast inloggningar från samma webbläsarsession. På grund av det populära alternativet av sessionsbaserade åtgärder, så är det värt att testa denna metod.
3. Den tredje inloggningssidan kommer att använda en 5-minuters lockout på databasnivå, där varje felaktigt inloggningsförsök kommer att loggas i databasen. Efter tre misslyckade försök kommer vidare inloggningsförsök för den användaren att blockeras från alla enheter och sessioner. För att testa den impopulära säkerhetsåtgärden på grund av DOS nackdelarna (OWASP u.å).
4. Den fjärde inloggningssidan är väldigt lik den tredje men istället för att blockera användaren från alla enheter och sessioner så kommer IP-adressen att blockeras, denna skyddsåtgärd håller också i 5-minuter. Att testa denna metod kan vara värdefullt för att jämföra resultaten med den sessionsbaserade mekanismen och den databasbaserade lösningen. Genom denna jämförelse kan man bedöma om den IP-adressbaserade lösningen är mer effektiv (OWASP u.å).

**Kali Linux-maskin:**

För att skapa en virtuell maskin (VM) med Kali Linux används ett program som heter Oracle VirtualBox, som måste installeras på en dator. En virtuell maskin fungerar som en separat, virtuell dator som kan användas för olika ändamål. I detta sammanhang skapas en Kali Linux-maskin specifikt för att genomföra brute-force-attacker. Den fysiska datorn fungerar som en värd (host) för alla virtuella maskiner som skapas.

Genom att använda en virtuell maskin istället för en fysisk dator med Kali Linux kan man ta ögonblicksbilder (snapshots) av systemet. Detta innebär att om ett misstag skulle ske under testningen kan man enkelt återgå till en tidigare snapshot. Denna funktion skapar en säker labbmiljö där man kan utforska och experimentera utan risk. Med tanke på laborationens syfte är en virtuell maskin det optimala valet. Anledningen till att Kali Linux används istället för andra operativsystem som Windows är att Kali Linux är speciellt utformat för penetrationstestning och säkerhetsforskning. I detta sammanhang är det därför en perfekt lösning (Kali u.å.).

## Genomförande av brute-force-attack

Experimentet kommer att inledas med att Hydra används för att systematiskt testa lösenord mot de olika inloggningssidorna. Hydra kommer att användas på grund av den användarvänliga strukturen Hydra är uppbyggt på samt effektiviteten och populariteten med Hydra (VarshaGrover, Gagandeep, 2020).

På den första inloggningssidan kommer Shein.txt-ordlistan (som innehåller 17 301 lösenord) att användas utan restriktioner i antalet försök. På de andra två sidorna kommer attacker att genomföras både före och efter att skyddsåtgärder såsom sessionslåsningsmekanismer och databasspärrar aktiveras. Brute-force-attacken ska utföras på alla inloggningssidor för att kunna se vilka ut av dessa säkerhetsåtgärder skyddar mot attacken bäst. Hemsidorna innehåller sju olika användarnamn som ska testas vid varje attack.

I den här laborationen används VM för att utföra brute-force-attacker med hjälp av olika verktyg som finns tillgängliga i Kali Linux, där ett av dessa verktyg är Hydra. Hydra kommer att användas för att testa säkerheten på inloggning hos webbservern och utvärdera hur sårbara de är samt Hashcat kommer att användas i sammanhang med hashade lösenord.

Tabell Olika login och säkerhetsåtgärder

|  |  |  |
| --- | --- | --- |
|  | Implementerade Säkerhetsåtgärder | Attack Metod |
| Login 1 | Inga | Hydra i kombination med användarnamn-ordlista och Shein-lösenordlistan. |
| Login 2 | Sessions-Baserad Timeout Mekanism efter 3 felaktiga försök per användare | Hydra i kombination med användarnamn-ordlista och Shein-lösenordlistan. |
| Login 3 | Databas-baserade Timeout Mekanism efter 3 felaktiga försök per användare | Hydra i kombination med användarnamn-ordlista och Shein-lösenordlistan. |
| Login 4 | IP-adress baserad blockeringsmekanism efter 3 felaktiga försök per användare. | Hydra i kombination med användarnamn-ordlista och Shein-lösenordlistan tillsammans med skript där IP-adress ändras varje 3 hydra försök. |
| Hashade lösenord | Inga | Hashcat i kombination med användarnamn-ordlista och Shein-lösenordlistan. |

## Datainsamling och Analys

Under experimentet kommer följande data att samlas in:

* Tidsåtgång för varje brute-force-attack.
* Totalt antal försök och totalt antal lyckade försök.

Datan kommer sedan att analyseras för att identifiera de faktorer som påverkar brute-force-attackerna mest samt för att utvärdera vilka säkerhetsåtgärder som visar sig vara mest effektiva för att motverka dessa attacker.

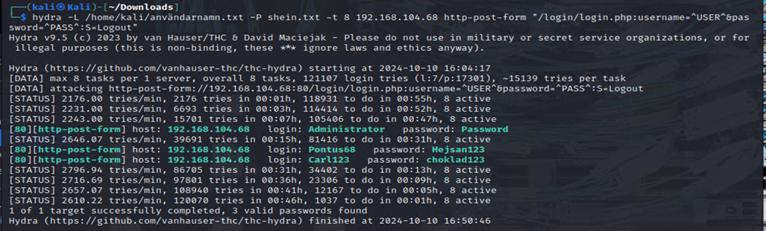
# RESULTAT

## Empiri på utfört experiment

### Sessionsbaserad inloggningssida utan lösenordsbegränsningar

Den första inloggningssidan var helt oskyddad, vilket innebar att den tillät obegränsade inloggningsförsök utan några säkerhetsåtgärder som session timeout eller kontolåsning. Hydra-verktyget användes för att genomföra brute-force-attacken, och eftersom det inte fanns några restriktioner kunde alla lösenord i ordlistan testas fritt.

Hydra gick systematiskt igenom hela Shein-ordlistan, som består av cirka 17 000 lösenord, och prövade dem mot varje användarnamn på sidan. Denna process resulterade i att alla användarnamn kunde attackeras utan avbrott, vilket ökade chanserna att knäcka lösenorden. Samtliga sju användarnamn lades i en separat ordlista och testades en efter en, vilket säkerställde att varje lösenord i Shein-ordlistan prövades mot varje användarnamn. I figur 4.1.1 så visas hela översikten av attacken som utfördes i Kali-Linux terminalen, all text som är markerad med grön text är de försök som lyckats.



Figur . Hydra mot oskyddad inloggninssida

**Resultat:**

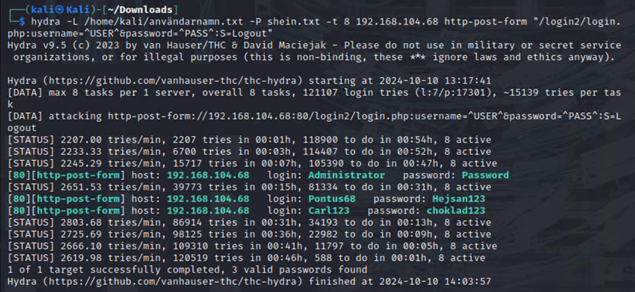
* Totalt försök per användarnamn: Hydra kunde genomföra alla 17 301 försök per användarnamn
* Total tid för hela attacken: Cirka 46 minuter för att utföra hela attacken mot samtliga användarnamn.
* Antal lyckade försök: 3 användarnamn hade lösenord som knäcktes av Hydra under denna session.

Dessa resultat visar tydligt att frånvaron av säkerhetsåtgärder på denna inloggningssida gjorde den mycket sårbar för brute-force-attacker. Hydra kunde testa ett stort antal lösenord på kort tid och lyckades knäcka flera konton utan att möta några begränsningar.

### Sessionsbaserad med lösenordsbegränsningar.

Den andra inloggningssidan använde en sessionsbaserad timeout-mekanism där kontot låstes i 5 minuter efter tre misslyckade inloggningsförsök per användarsession. Detta skulle i teorin göra brute-force-attacker svårare, eftersom användaren måste vänta efter ett antal misslyckade försök.

Hydra-verktyget användes återigen, och samma metod som i föregående attack applicerades. Figur 4.1.2 visar en översikt av attacken. Hydra kunde dock kringgå den sessionsbaserade låsningen genom att öppna en ny session, till exempel genom att starta en ny webbläsarsession i inkognito-läge. På detta sätt kunde attacken fortsätta utan avbrott.

Figur . Hydra mot sessionsbaserad säkerhetsåtgärd

**Resultat:**

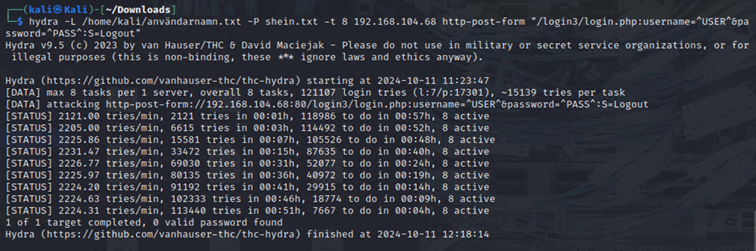
* Totalt försök per användarnamn: Hydra kunde genomföra alla 17 301 försök per användarnamn
* Total tid för hela attacken: Cirka 46 minuter.
* Antal lyckade försök: 3 lösenord knäcktes.

Trots implementeringen av säkerhetsåtgärder visade sig denna attack vara lika effektiv som den första, där ingen säkerhetsåtgärd fanns. Det beror på att den sessionsbaserade låsningen endast gäller för den aktuella webbläsarsessionen. Genom att starta en ny session kunde Hydra fortsätta attacken utan att påverkas av tidsbaserade begränsningar. Detta resultat illustrerar att den valda skyddsåtgärden var otillräcklig för att skydda mot en brute-force-attack med denna metod.

### Databas-baserade inloggningsbegränsningar

Denna inloggningssida bygger på samma principer som beskrivs i avsnitt 4.1.2, men med en avgörande skillnad i timeout-mekanismen. Säkerhetsåtgärden är här implementerad direkt i databasen. Vid varje inloggningsförsök kontrollerar webbservern om användaren har haft tidigare felaktiga försök samt när dessa inträffade. Om antalet felaktiga försök uppgår till tre, blockeras användarens åtkomst under en period av fem minuter.

I samband med detta genomfördes en brute-force-attack med verktyget Hydra, i kombination med Shein-lösenordslistan, för att testa säkerheten hos den databasbaserade inloggningsbegränsningen. Resultaten visar att denna metod kraftigt försvårar brute-force-attacker. Inget lösenord lyckades knäckas under testets gång som syns på figur 4.1.3, vilket understryker åtgärdens effektivitet.



Figur . Hydra mot databasbaserad säkerhetsåtgärd

Resultat:

* Totalt antal försök: Hydra kunde genomföra alla 121 107 försök
* Total tid för hela attacken: Cirka 55 minuter.
* Antal lyckade försök: 0 lösenord knäcktes

Denna säkerhetsåtgärd var mycket effektiv då attacken stoppades helt, och inga lösenord knäcktes eftersom användaren blockeras efter tre misslyckade försök. Nackdelen är att även legitima användare kan låsas ute i fem minuter, vilket påverkar tillgängligheten. En angripare kan också utnyttja detta genom en Denial of Service (DoS)-attack, vilket kan leda till att många användare tillfälligt förlorar åtkomst till sina konton och påverkar användarupplevelsen negativt.

### IP-adressbaserade inloggningsbegränsningar

Den fjärde inloggningssidan använder en säkerhetsmekanism som blockerar IP-adresser efter tre misslyckade inloggningsförsök under en viss tidsperiod. IP-adressen spärras i fem minuter, vilket förhindrar ytterligare försök från samma källa.

Denna säkerhetsåtgärd försvårar brute-force-attacker från samma IP-adress, men en angripare kan kringgå det genom att använda tekniken "IP-hopping," där IP-adressen ändras efter var tredje försök. Ett test med ett skript som automatiskt bytte IP-adress vid varje tredje försök visade att attacken kunde fortsätta trots blockeringen.

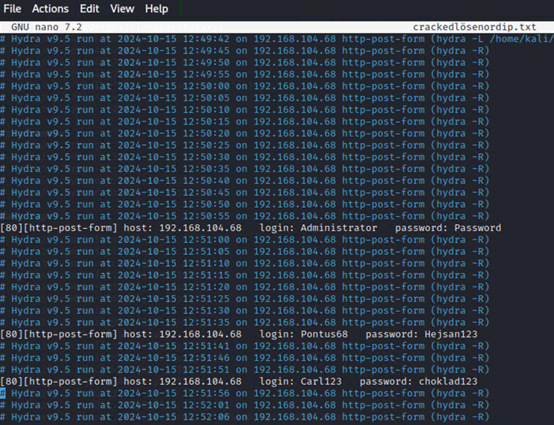
Loggningen av misslyckade försök registrerar IP-adresser, användarnamn, tidpunkt och antalet försök, vilket hjälper till att identifiera angreppsmönster. Testet använde en mindre ordlista (20 ord), vilket resulterade i tre knäckta lösenord på tre minuter (se figur 4.1.4).

A screenshot of a computer

Description automatically generated

Figur . Lösenordslista mot IP-adressbaserad skyddsåtgärd

Figur 4.1.5 illustrerar att samma användarnamn som identifierades i tidigare attacker kunde användas för att genomföra inloggningsförsök, där alla vita rader är dem lyckade försöken, vilket tyder på att säkerhetsåtgärden kring IP-blockering kan kringgås med vissa tekniker. Genom att byta IP-adress efter tre misslyckade försök kunde attacken fortsätta.



Figur . Lyckad attack mot IP-adressbaserad skyddsåtgärd

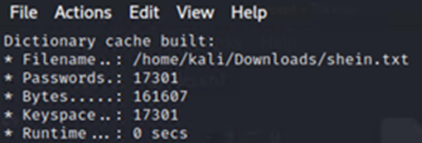
Resultat:

* Totalt antal försök: Hydra kunde genomföra alla försök för alla användarnamn och lösenord.
* Total tid för hela attacken: Cirka 3 minuter.
* Antal lyckade försök: 3 lösenord knäcktes.

Resultaten visar att trots IP-baserade begränsningar kunde attacken fortsätta, vilket indikerar att denna åtgärd inte helt stoppar brute-force-attacker. Inledningsvis testades attacken med Shein-ordlistan, men det visade sig ta för lång tid. Istället användes en mindre ordlista med 20 lösenord, vilket visade att om en större ordlista som Shein (17 000 rader) hade använts, skulle attacken ha varit betydligt mer långsam. Attacken stoppades efter tre minuter, vilket tyder på att åtgärden skulle kunna sakta ner en attack i ett verkligt scenario, men inte helt förhindra den.

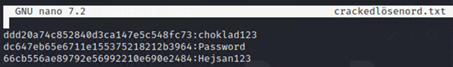
### Hashade lösenord

Hashcat kördes med tre hashade lösenord (”choklad123”, ”Password”, ”Hejsan123”) med SHA1 algoritmen, där ordlistan "Shein" användes. Lösenorden identifierades omedelbart (på 0 sekunder) se figur 4.1.6, och resultatet sparades i en textfil.



Figur . Hashcat mot Shein

I den andra delen användes istället MD5 som hashalgoritm. Även i detta fall kunde programmet utan svårigheter identifiera lösenordet, vilket resulterade i textfilen ”crackedlösenord.txt” har prompt som illustreras på figur 4.1.7.



Figur . Knäckta hashade lösenord

Valet av hashalgoritm är avgörande för resultaten. Om fel algoritm används, som att köra en MD5-hash mot en SHA1-hash, rapporterar Hashcat att inga hashar hittats och att fel hashtyp kan vara specificerad (se figur 4.1.8). Rätt hashformat kan identifieras via enkla sökningar online.

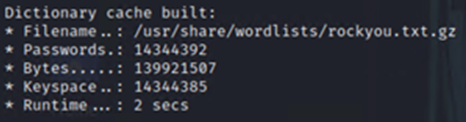
A screenshot of a computer

Description automatically generated

Figur . Hashcat med fel hashalgoritm

Därefter undersöks effekten om man tillägger salt på hashen, om salt värdet är känt har programmet inga större svårigheter att lösa hashen, likt tidigare resultat. Är salt värdet okänt kan inte Hashcat återställa lösenordet.

Genom hastigheten vid Hashcat mot shein.txt som innehåller 17 301 ord så testas också hashen för ”faith813” med MD5 kryptering mot ordlistan rockyou.txt som innehåller 14 344 392 olika ord. Efter endast 2 sekunder så har alla 14 miljoner ord matchas mot hashen för ”faith813” och lösenordet hittades (se figur 4.1.9).



Figur . Hashcat mot RockYou

## Att mitigera säkerhetsbrister

För att optimera säkerheten på inloggningssidor bör starka lösenord kombineras med andra åtgärder för att skapa flerlagerskydd. Laborationerna visade att kombinationen av starka lösenord och ytterligare säkerhetsåtgärder markant förbättrar säkerheten.

Begränsning av inloggningsförsök visade sig vara effektiv mot brute-force-attacker. Genom att blockera användaren eller IP-adressen efter ett visst antal misslyckade försök minskar risken för att en angripare ska testa tusentals lösenord. Experimentet visade att attackerna saktades ned med en tidsbaserad blockering efter tre misslyckade försök.

Även om IP-baserad blockering inte helt stoppade attackerna, sänkte den deras hastighet, vilket underlättade upptäckt av angrepp i realtid. Databasbaserad begränsning stoppade däremot attackerna helt men kan medföra att legitima användare tillfälligt förlorar åtkomst till sina konton, vilket riskerar att utnyttjas i en Denial of Service (DoS)-attack.

Denna metod är särskilt effektiv för att skydda kritiska konton, som administratörskonton, där risken för större skada är högre. Genom att fokusera på högriskkonton kan säkerheten balanseras mot tillgänglighet, vilket minskar risken för allvarliga angrepp utan att försämra användarupplevelsen för vanliga konton.

### Kombination av säkerhetsåtgärder

Multifaktorautentisering (MFA) är en viktig säkerhetsmekanism som lägger till ett extra lager skydd genom att kräva en andra verifiering, till exempel en SMS-kod. Även om MFA inte testades i experimentet, visar forskning enligt (Ahmad et al. 2023) och (Intelligent Technical Solutions 2022) att brute-force-attacker blir ineffektiva om MFA är aktiverat, eftersom angriparen inte kan logga in utan att först verifiera sig igen.

Resultaten visade att ingen enskild åtgärd helt kan stoppa brute-force-attacker, men kombinationer ger ett starkare skydd. IP-blockering bromsade attackerna men kunde inte stoppa dem helt, och genom att lägga till MFA skulle angriparen ändå behöva passera ytterligare säkerhetslager. CAPTCHA skulle ytterligare komplicera situationen genom att blockera automatiserade attacker.

Enligt OWASP kan en människa klara en korrekt utformad CAPTCHA nästan 100% av försöken men den kan effektivt stoppa bots, och tillsammans med MFA och IP-blockering blir brute-force-attacker i princip omöjliga.

Att använda hashade lösenord i stället för klartext ger ett starkt skydd. I laborationen visade sig dock hashade lösenord vara möjliga att dekryptera med verktyg som Hashcat, men genom att lägga till salt blev dekrypteringen betydligt svårare. I verkliga scenarier är saltet hemligt, vilket försvårar attacken ytterligare.

Genom att kombinera IP-blockering, CAPTCHA, MFA och hashning med salt skapas ett robust skydd mot brute-force-attacker. Dock kvarstår en svaghet i lösenordens styrka. Enkla lösenord som "Hejsan123" knäcktes lätt under testerna, medan komplexa lösenord som "pd7$Ke9kJUP" (se figur 4.2.1) inte knäcktes alls. Nackdelen med sådana komplexa lösenord är dock att de är svåra att komma ihåg för de flesta människor.

A screenshot of a computer

Description automatically generated

Figur . Användarnamn och lösenord för inloggningssidan

Lösenordshanterare kan hjälpa användare att generera och hantera säkra lösenord, vilket minskar risken att samma lösenord används för flera tjänster. Webbplatser kan också bidra genom att införa regler som kräver starka lösenord som måste innehålla specialtecken, stora och små bokstäver samt siffror och säkerhetsmätare som indikerar lösenordens styrka.

En intressant observation från testerna var att ”hejsan123” faktiskt finns i den populära ordlistan Rockyou, som innehåller cirka 14 miljoner lösenord, medan ”Hejsan123” inte alls finns där. Skillnaden mellan dessa två är endast att den första bokstaven i det ena lösenordet är stor. Även om "Hejsan123" i sig fortfarande är ett relativt svagt lösenord, visar detta hur enkel förändring, som en stor bokstav, kan göra ett lösenord mycket svårare att brute-forca och undvika kända lösenordslistor.

## Ekonomisk hållbarhet

Ett centralt problem är att komprometterade användarkonton underminerar användarnas förtroende för företaget. Om ett företag upplevs som osäkert, kan användare välja att lämna plattformen eller tjänsten, vilket i sin tur leder till minskade intäkter och förlorade marknadsandelar. Den skadade relationen mellan företaget och dess användare kan också påverka företagets rykte negativt, vilket kan få långtgående ekonomiska konsekvenser när befintliga och potentiella kunder tappar tilliten till företagets förmåga att skydda deras personliga uppgifter. Missnöjet sprider sig snabbt, särskilt genom negativa recensioner och omdömen. Om några kunder uttrycker sitt missnöje genom dåliga recensioner eller negativa kommentarer till vänner och familj, kan det snabbt förvärra förtroendekrisen. Detta kan leda till att ännu fler kunder tappar tilliten och väljer att undvika företaget. Följden blir ett skadat rykte som kan vara svårt att reparera, med potentiellt stora ekonomiska förluster (Bergman & Klevsjö, 2020, S.127).

Företag som inte implementerar korrekta säkerhetsåtgärder för att skydda sina användarkonton från brute-force-attacker misslyckas med att upprätthålla flera aspekter av CIA-triangeln: konfidentialitet, integritet och tillgänglighet. När konton blir komprometterade till följd av otillräckliga säkerhetsåtgärder, hotas först och främst konfidentialiteten, eftersom användarnas känsliga data kan hamna i fel händer. Detta påverkar företagets trovärdighet negativt, då det underminerar användarnas förtroende för att deras personliga information skyddas. Brister i konfidentialiteten kan dessutom skada samarbeten med andra företag, som i sin tur kan bli måltavlor för nästkommande attacker. Integriteten hotas om hackade konton används för att ändra eller manipulera information, vilket ytterligare skadar förtroendet för tjänsten.

Tillgängligheten kan påverkas både genom avbrott i tjänster till följd av säkerhetsincidenter och genom att legitima användare blockeras från sina konton. Experimentet (4.1.3 och 4.1.4) visade också att databas- eller IP-baserade låsningar kan utnyttjas för att genomföra Denial of Service (DoS)-attacker. Genom att överbelasta systemet med misslyckade inloggningsförsök kan en angripare blockera legitima användare, vilket resulterar i avbrott i tjänster. Dessa faktorer kan leda till ett skadat rykte, minskade användarsiffror och betydande ekonomiska förluster genom uteblivna intäkter och ökade driftkostnader, vilket påverkar företagets ekonomiska hållbarhet och effektivitet.

Vid återkommande brute-force-attacker kan systemkompromettering leda till allvarliga juridiska konsekvenser. Företag som inte uppfyller säkerhetsstandarder som GDPR eller NIS-direktivet kan åläggas höga böter och sanktioner, vilket skapar ytterligare ekonomiska påfrestningar. Till exempel så fick ägaren av Shein (Zoetop), enligt New York State Attorny General (2022), böter på 1,9 miljoner dollar när databasen hade läckts ut på nätet, där det fanns ca 39 miljoner användarinformation i databasen. Dessa kostnader, i kombination med förlorat förtroende från kunder och partners, påverkar långsiktig lönsamhet och kan skada företagets varumärke, vilket försämrar dess ekonomiska hållbarhet över tid.

# Analys

Syftet med denna laboration var att undersöka effekten av olika säkerhetsåtgärder mot brute-force-attacker och hur dessa kan implementeras för att skydda användarkonton. Genom att analysera resultaten från experimenten och diskussionerna har vi kommit fram till att ingen enskild åtgärd kan helt förhindra brute-force-attacker. Istället är en flerlagers strategi som kombinerar olika säkerhetsåtgärder avgörande för att skapa ett robust skydd.

Resultaten visar att även med grundläggande säkerhetsåtgärder, såsom IP-blockering, kan brute-force-attacker fortsätta, vilket indikerar att angripare kan anpassa sina metoder för att kringgå dessa begränsningar. När vi testade olika hashing-algoritmer och observerade hur snabbt lösenord kunde knäckas, blev det tydligt att valet av hashing-algoritm spelar en avgörande roll. MD5 visade sig vara sårbar, medan SHA1, trots att det är något mer robust, fortfarande kunde knäckas när angriparen hade tillgång till hashade lösenord.

I introduktionen framhölls vikten av att skydda användardata och de potentiella konsekvenserna av brute-force-attacker för både individer och företag. Den ekonomiska hållbarheten påverkas direkt av hur väl företag skyddar sina användares data. Om ett företag upplever dataintrång kan detta leda till förlorad kundtillit och därmed påverka intäkterna och företagets rykte negativt. Vår slutsats bekräftar att investeringar i säkerhetsåtgärder är kritiska för att upprätthålla detta förtroende.

Sammanfattningsvis visar analysen att medan brute-force-attacker är en verklig och allvarlig risk, kan de avsevärt begränsas genom att använda en kombination av starka lösenord, IP-adressblockeringar, hashning med salt, MFA och CAPTCHA. Det är avgörande för företag att inse att säkerhet är en kontinuerlig process, inte en engångsinsats. Att anpassa sig till hotet inom cybersäkerhet är avgörande för att skydda användardata och upprätthålla förtroendet mellan företag och deras kunder.

# REFERENSER

Ahmad, O. Tripathi, G. Siddiqui, F. Alam, M. A. Ahad, M. A. Akhtar, M. M. & Casalino, G. (2023). *A Blockchain-Based Multi-Factor Authentication Mechanism for Securing Smart Cities* https://www.mdpi.com/1424-8220/23/5/2757

[Hämtad: 2024-10-16]

Bergman, B. & Klefsjö, B. (2020). *Kvalitet från behov till användning*. 6 upplagan. Lund: Studentlitteratur.

[Hämtad: 2024-10-20]

Code Signing (u.å.) *What Is a Hashing Algorithm? A Look at Hash Functions.*

https://codesigningstore.com/what-is-hashing-algorithm-how-it-works

[Hämtad: 2024-10-10]

Covert, J., Francis, D., Steinhagen, P. & Streff, K. (2020). *CIA Triad: A Security Framework*.

https://scholarspace.manoa.hawaii.edu/server/api/core/bitstreams/5486a250-cc3c-4227-a752-7d08378afbdf/content

[Hämtad: 2024-10-06]

Dashlane (u. å.). *A Brief History of Passwords.*

https://www.dashlane.com/a-brief-history-of-passwords

[Hämtad: 2024-09-23]

Descalso, S. (2022). How to Prevent Brute Force Attacks in 8 Easy Steps [Updated]. *Intelligent Technical solutions.* 11 januari.

https://www.itsasap.com/blog/how-to-prevent-brute-force-attacks

[Hämtad: 2024-10-16]

Esheridan, Open Web Application Security Project (OWASP) (u.å.). *Blocking Brute Force Attacks.* https://owasp.org/www-community/controls/Blocking\_Brute\_Force\_Attacks

[Hämtad:2024-10-01]

Grover, V. & Gagandeep. (2020). *An Efficient Brute Force Attack Handling Techniques for Server Virtualization* https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=3564447

[Hämtad: 2024-10-08]

Hashcat (u.å.).

https://hashcat.net/hashcat/

[Hämtad: 2024-09-29]

Lurey, C. (2023). Understanding RockYou.txt: A Tool for security and a weapon for hackers.

*Keeper.* [Blogg]*.* 04 augusti.

https://www.keepersecurity.com/blog/2023/08/04/understanding-rockyou-txt-a-tool-for-security-and-a-weapon-for-hackers/

[Hämtad: 2024-09-23]

New York State Attorney General (2022). *Attorney General James Secures $1.9 Million From E-Commerce Companies Shein and Romwe Owner Zoetop for Failing to Protect Consumers’ Data.* [Pressmeddelande], 12 oktober.

https://ag.ny.gov/press-release/2022/attorney-general-james-secures-19-million-e-commerce-shein-and-romwe-owner-zoetop

[Hämtad: 2024-10-18]

Open Web Application Security Project (OWASP) (u. å.). *Password storage cheat sheet*

https://cheatsheetseries.owasp.org/cheatsheets/Password\_Storage\_Cheat\_Sheet.html

[Hämtad: 2024-09-23]

Tzu, S. (2005) The Art of War. Översatt av Thomas Cleary. Shambhala Publications. [Hämtad: 2024-10-04]

Vasiu, I. & Vasiu, L. (2018). *Cybersecurity as an Essential Sustainable Economic Development Factor* https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=3262527

[Hämtad: 2024-10-06]

Vugdelija, N. Nedeljković, N. Kojić, N. Lukić, L. & Vesić, M. (2021). *Review of Brute-Force Attack and Protection Techniques* https://proceedings.ictinnovations.org/attachment/paper/554/review-of-brute-force-attack-and-protection-techniques.pdf

[Hämtad: 2024-10-04]

HB_logo1_K

Besöksadress: Allégatan 1 · Postadress: 501 90 Borås · Tfn: 033-435 40 00 · E-post: registrator@hb.se · Webb: www.hb.se