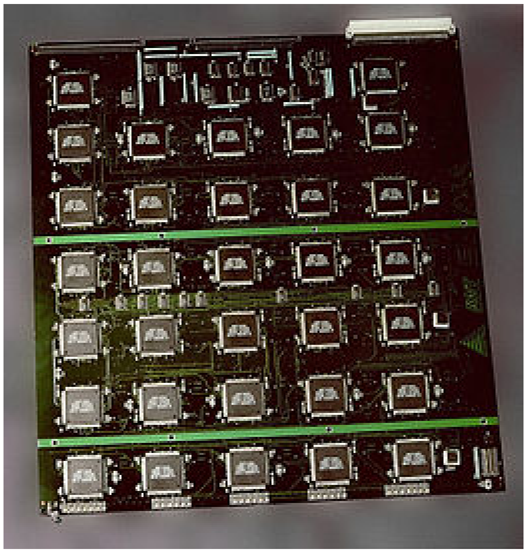
Brute-force attack



The Electronic Frontier Foundation's US$250,000 DES cracking machine contained over 1,800 custom chips *and could brute-force a DES key in a matter of days. The photograph shows a DES Cracker circuit board fitted on both sides with 64 Deep Crack chips.*

In cryptography, a **brute-force attack** consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly. The attacker systematically checks all possible passwords and passphrases until the correct one is found. Alternatively, the attacker can attempt to guess the key which is typically created from the password using a key derivation function. This is known as an **exhaustive key search**.

A brute-force attack is a cryptanalytic attack that can, in theory, be used to attempt to decrypt any encrypted data[1] (except for data encrypted in an information-theoretically secure manner). Such an attack might be used when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier.

When password-guessing, this method is very fast when used to check all short passwords, but for longer passwords other methods such as the dictionary attack are used because a brute-force search takes too long. Longer passwords, passphrases and keys have more possible values, making them exponentially more difficult to crack than shorter ones.

Brute-force attacks can be made less effective by obfuscating the data to be encoded making it more difficult for an attacker to recognize when the code has been cracked or by making the attacker do more work to test each guess. One of the measures of the strength of an encryption system is how long it would theoretically take an attacker to mount a successful brute-force attack against it.

Brute-force attacks are an application of brute-force search, the general problem-solving technique of enumerating all candidates and checking each one.

Basic concept

Brute-force attacks work by calculating every possible combination that could make up a password and testing it to see if it is the correct password. As the password's length increases, the amount of time, on average, to find the correct password increases exponentially.

Theoretical limits

The resources required for a brute-force attack grow exponentially with increasing key size, not linearly. Although U.S. export regulations historically restricted key lengths to 56-bit symmetric keys (e.g. Data Encryption Standard), these restrictions are no longer in place, so modern symmetric algorithms typically use computationally stronger 128- to 256-bit keys.

There is a physical argument that a 128-bit symmetric key is computationally secure against brute-force attack. The so-called Landauer limit implied by the laws of physics sets a lower limit on the energy required to perform a computation of $kT \cdot \ln 2$ per bit erased in a computation, where *T* is the temperature of the computing device in kelvins, *k* is the Boltzmann constant, and the natural logarithm of 2 is about 0.693. No irreversible computing device can use less energy than this, even in principle.[2] Thus, in order to simply flip through the possible values for a 128-bit symmetric key (ignoring doing the actual computing to check it) would, theoretically, require 2^{128} −

1 bit flips on a conventional processor. If it is assumed that the calculation occurs near room temperature (~300 K), the Von Neumann-Landauer Limit can be applied to estimate the energy required as ~10^{18} joules, which is equivalent to consuming 30 gigawatts of power for one year. This is equal to 30×10^{9} W×365×24×3600 s = 9.46×10^{17} J or 262.7 TWh (more than 1% of the world energy production). The full actual computation – checking each key to see if a solution has been found – would consume many times this amount.

Furthermore, this is simply the energy requirement for cycling through the key space; the actual time it takes to flip each bit is not considered, which is certainly greater than 0.

However, this argument assumes that the register values are changed using conventional set and clear operations which inevitably generate entropy. It has been shown that computational hardware can be designed not to encounter this theoretical obstruction (see reversible computing), though no such computers are known to have been constructed.

Modern GPUs are well-suited to the repetitive tasks associated with hardware-based password cracking

As commercial successors of governmental ASIC solutions have become available, also known as custom hardware attacks, two emerging technologies have proven their capability in the brute-force attack of certain ciphers. One is modern graphics processing unit (GPU) technology,[3] the other is the field-programmable gate array (FPGA) technology. GPUs benefit from their wide availability and price-performance benefit, FPGAs from their energy efficiency per cryptographic operation. Both technologies try to transport the benefits of parallel processing to brute-force attacks. In case of GPUs some hundreds, in the case of FPGA some thousand processing units making them much better suited to cracking passwords than conventional processors. Various publications in the fields of cryptographic analysis have proved the energy efficiency of today's FPGA technology, for example, the COPACOBANA FPGA Cluster computer consumes the same energy as a single PC (600 W), but performs like 2,500 PCs for certain algorithms. A number of firms provide hardware-based FPGA cryptographic analysis solutions from a single FPGA PCI Express card up to dedicated FPGA computers. WPA and WPA2 encryption have successfully been brute-force attacked by reducing the workload by a factor of 50 in comparison to conventional CPUs[4][5] and some hundred in case of FPGAs.

A single COPACOBANA board boasting 6 Xilinx Spartans – a cluster is made up of 20 of these

AES permits the use of 256-bit keys. Breaking a symmetric 256-bit key by brute force requires 2^{128} times more computational power than a 128-bit key. Fifty supercomputers that could check a billion billion (10^{18}) AES keys per second (if such a device could ever be made) would, in theory, require about 3×10^{51} years to exhaust the 256-bit key space.

An underlying assumption of a brute-force attack is that the complete keyspace was used to generate keys, something that relies on an effective random number generator, and that there are no defects in the algorithm or its implementation. For example, a number of systems that were originally thought to be impossible to crack by brute force have nevertheless been cracked because the key space to search through was found to be much smaller than originally thought, because of a lack of entropy in their pseudorandom number generators. These include Netscape's implementation of SSL (famously cracked by Ian Goldberg and

David Wagner in 1995[6]) and a Debian/Ubuntu edition of OpenSSL discovered in 2008 to be flawed.[7] A similar lack of implemented entropy led to the breaking of Enigma's code.[8][9]

Credential recycling

Credential recycling refers to the hacking practice of re-using username and password combinations gathered in previous brute-force attacks. A special form of credential recycling is pass the hash, where unsalted hashed credentials are stolen and re-used without first being brute forced.

Unbreakable codes

Certain types of encryption, by their mathematical properties, cannot be defeated by brute force. An example of this is one-time pad cryptography, where every cleartext bit has a corresponding key from a truly random sequence of key bits. A 140 character one-time-pad-encoded string subjected to a brute-force attack would eventually reveal every 140 character string possible, including the correct answer – but of all the answers given, there would be no way of knowing which was the correct one. Defeating such a system, as was done by the Venona project, generally relies not on pure cryptography, but upon mistakes in its implementation: the key pads not being truly random, intercepted keypads, operators making mistakes – or other errors.[10]

Countermeasures

In case of an offline attack where the attacker has access to the encrypted material, one can try key combinations without the risk of discovery or interference. However database and directory administrators can take countermeasures against online attacks, for example by limiting the number of attempts that a password can be tried, by introducing time delays between successive attempts, increasing the answer's complexity (e.g. requiring a CAPTCHA answer or verification code sent via cellphone), and/or locking accounts out after unsuccessful logon attempts.[11] Website administrators may prevent a particular IP address from trying more than a predetermined number of password attempts against any account on the site.[12]

Reverse brute-force attack

In a **reverse brute-force attack**, a single (usually common) password is tested against multiple usernames or encrypted files.[13] The process may be repeated for a select few passwords. In such a strategy, the attacker is generally not targeting a specific user.

Software that performs brute-force attacks

- Aircrack-ng
- Cain and Abel

- Crack
- DaveGrohl
- Hashcat
- John the Ripper
- L0phtCrack
- Ophcrack
- RainbowCrack

See also

- Bitcoin mining
- Cryptographic key length
- Distributed.net

- Key derivation function
- MD5CRK
- Metasploit Express
- Side-channel attack
- TWINKLE and TWIRL
- Unicity distance
- RSA Factoring Challenge
- Secure Shell

Notes

1. *Paar 2010, p. 7.*
2. *Landauer 1961, p. 183-191.*

3. *Graham 2011*.
4. *Kingsley-Hughes 2008*.
5. *Kamerling 2007*.
6. *Viega 2002, p. 18*.
7. *CERT-2008*.
8. *Ellis*.
9. *NSA-2009*.
10. *Reynard 1997, p. 86*.
11. *Burnett 2004*.
12. *Ristic 2010, p. 136*.

13. *"InfoSecPro.com - Computer, network, application and physical security consultants" . www.infosecpro.com. Archived from the original on April 4, 2017. Retrieved May 8, 2018.*

References

- Adleman, Leonard M.; Rothemund, Paul W.K.; Roweis, Sam; Winfree, Erik (June 10–12, 1996). *On Applying Molecular Computation To The Data Encryption Standard*. *Proceedings of the Second Annual Meeting on DNA Based Computers*. Princeton University.

- *Cracking DES – Secrets of Encryption Research, Wiretap Politics & Chip Design* . Electronic Frontier Foundation. ISBN 1-56592-520-3.
- Burnett, Mark; Foster, James C. (2004). *Hacking the Code: ASP.NET Web Application Security* . Syngress. ISBN 1-932266-65-8.
- Diffie, W.; Hellman, M.E. (1977). "Exhaustive Cryptanalysis of the NBS Data Encryption Standard". *Computer*. **10**: 74–84. doi:10.1109/c-m.1977.217750 .
- Graham, Robert David (June 22, 2011). "Password cracking, mining, and GPUs" . erratasec.com. Retrieved August 17, 2011.

- Ellis, Claire. "Exploring the Enigma" . Plus Magazine.
- Kamerling, Erik (November 12, 2007). "Elcomsoft Debuts Graphics Processing Unit (GPU) Password Recovery Advancement" . Symantec.
- Kingsley-Hughes, Adrian (October 12, 2008). "ElcomSoft uses NVIDIA GPUs to Speed up WPA/WPA2 Brute-force Attack" . *ZDNet*.
- Landauer, L (1961). "Irreversibility and Heat Generation in the Computing Process" . *IBM Journal of Research and Development*. **5**. doi:10.1147/rd.53.0183 .

- Paar, Christof; Pelzl, Jan; Preneel, Bart (2010). *Understanding Cryptography: A Textbook for Students and Practitioners* . Springer. ISBN 3-642-04100-0.
- Reynard, Robert (1997). *Secret Code Breaker II: A Cryptanalyst's Handbook* . Jacksonville, FL: Smith & Daniel Marketing. ISBN 1-889668-06-0. Retrieved September 21, 2008.
- Ristic, Ivan (2010). *Modsecurity Handbook* . Feisty Duck. ISBN 1-907117-02-4.
- Viega, John; Messier, Matt; Chandra, Pravir (2002). *Network Security with OpenSSL* . O'Reilly. ISBN 0-596-00270-X. Retrieved November 25, 2008.

- Wiener, Michael J. (1996). "Efficient DES Key Search". *Practical Cryptography for Data Internetworks*. W. Stallings, editor, IEEE Computer Society Press.
- "Technical Cyber Security Alert TA08-137A: Debian/Ubuntu OpenSSL Random Number Generator Vulnerability" . United States Computer Emergency Readiness Team (CERT). May 16, 2008. Retrieved August 10, 2008.
- "NSA's How Mathematicians Helped Win WWII" . National Security Agency. January 15, 2009.

External links

- RSA-sponsored DES-III cracking contest
- Demonstration of a brute-force device designed to guess the passcode of locked iPhones running iOS 10.3.3
- How We Cracked the Code Book Ciphers – Essay by the winning team of the challenge in The Code Book

Retrieved from "https://en.wikipedia.org/w/index.php?title=Brute-force_attack&oldid=933973895"

Last edited 5 days ago by Mike Mounier