Global Information Assurance Certification Paper

A Non-technical Perspective: Authentication

AKA: The Idiot's Guide to Passwords

Matthew L. Galin

May 18, 2004

GSEC Practical Assignment 1.4b
Option 1: Research on Topics in Information Security

Abstract

For many companies, their assets rely in the information stored within computer systems. Aside from these corporate assets, personal private information is also stored in many of these same computers. Almost all computer systems protect this data with passwords.

Passwords, if used improperly, present a large exposure in protecting personal and proprietary corporate data. The "Human factor" contributes to the use of bad and easily compromised passwords. People often weigh on the side of usability instead of selecting secure passwords.

It is the intent of this document to explore, in a non-highly technical way, how to use passwords in a more secure fashion. To provide these best practices, background and detail will be provided on how passwords are used and compromised within computer systems. Some alternatives and supplemental technologies will also be provided. Better passwords and technology will not eliminate the risk completely, but reduce it significantly.

Introduction

The dictionary defines password as "something that enables one to pass or gain admission"[1]. Passwords have been used for thousands of years. They are typically given to someone in return for something else, may it be entry or some other object of desire. The modern application of a password would be a private word or phrase used to gain entry to some type of computer system.

Passwords are often the first and last line of defense of protecting electronic information assets. Electronic information assets are defined as anything of value that is stored in an electronic form. These assets can be anything from a social security number in a spreadsheet on someone's personal computer, to patent health records inside a heavily fortified database at a large healthcare organization.

The problem with passwords is that they have to be kept secret to be useful in securing something. For them to be kept secret they should be complex enough to avoid being guessed by either a person or computer. They have to be simple enough so that someone does not write them on a piece of paper next to their computer or under their keyboard. However, they have to be unique enough so that people do not use the same password across every computerized system they come in contact with. Industry has spent a good deal of money toward technology to keep systems secure.

Much of the risk and vulnerably associated with weak passwords can be mitigated by selecting better passwords. In addition to proper selection, there are also technologies in hardware and software that can supplement or replace passwords regardless if they are weak or not.

A good understanding of how passwords are stored and how they can be reverse engineered will help one make better informed decisions when selecting a password. We are going to explore some of weaknesses found in passwords so there is a better understanding of how easily a password can be acquired by even the least technical people. We will then pose some best practices in how to select a good password. There will also be some technology discussed that can be used to replace or supplement passwords. The intent of this document is to give you a better, basic understanding on how people and computers use passwords for authentication. Hopefully with this understanding, you will be alarmed at how poor many users' passwords are, and how easily they can be compromised if care isn't taken.

Weakness of Passwords

Everyone uses passwords for something in their life. It may be a mother's maiden name on the phone from your credit card company. It may be a four digit ATM machine pin number. Passwords almost always have the same enormous

issue; the human factor. "In the security world, that weak link is the human element, and it manifests in the poor management of user passwords."[3] The majority of the time, the passwords are related to something personal about the person using it. If someone knows personal facts or information about the person, passwords can often be guessed. Passwords often weigh to the side of usability instead of security. In the example with an ATM machine, it's most common for people to use birthdays of themselves or their spouses, part of their home phone number, basically something easy to remember. Almost every bank provides the following statement to their customers when they set up an account; this is an excerpt from an actual bank. "When selecting a PIN, avoid numbers and letters that can be easily identified with you. For example, don't use your initials, birthday, telephone or Social Security number."[4] Pin numbers are only one example. Pin numbers are usually only 4 digits long in the United States. Passwords to other systems such as computer systems are often longer than this but they still suffer from the same weakness.

Passwords are most commonly a dictionary word, thus far from random. They are often a set of letters from the user's language. In our case, we'll assume English letters, such as "A to Z", sometimes upper or lower case, without spaces. Some examples are "Zebra", "Green", "summer", "Friday" or "July".

Figure 1 illustrates a sampling of a password audit, done on a production Windows NT domain. Here are some general statistics:

- 46 contained a typical persons first name
- 41 were the word "password"
- Several contained a state, city, a month or day of the week.

BUILDSERVER	BLOOMFIELD	DIAMOND1	GROUNDHOG	LACROSSE	NCC1701O	PASSWORD	POSSIBLE	SPRINGTIME
PARKING1	BRISTOL1	DISCOVERY	HALLOWEEN	LANCELOT	NEWBORN2	PASSWORD	PRESCOTT	SPRINGTIME
12345678	BROTHER0	DONALD	HAWKEYE3	LAVENDER	NEWPASS	PASSWORD	PRINCESS	SPRINGTIME
1234567A	BRYANT	DRUMMING	HEATHER9	LENINGRAD	NEWPASS2	PASSWORD	PRINCETON	STARFISH
ACCIDENT	BUTTERFLY	EGGPLANT	HERITAGE	LETMEIN1	NEWPASS	PASSWORD	PROBLEMS	SEPTEMBER
AMBITION	CAITLIN1	ELIZABE1	HOLIDAYS	LITURGICAL	NEWPORT7	PASSWORD	PROFESSIONAL	SEPTEMBER
AMERICAN	CAITLIN5	EMERALD5	HOLIDAY4	MANAGER1	NEWYORKKNICKS	PASSWORD	PROJECTS	SEPTEMBER
AMERICAN	CANDACEC	EQUINOX9	HOLIDAY8	MANAGER1	NEWYORKGIANTS	PASSWORD	PROPHETS	SEPTEMBER
ANDERSON	CHARLES3	FEATHER1	HONEYBEE	MARRIOTT	NEWYORK2	PASSWORD	RAINBOW5	SEPTEMBER
ANDOVER1	CHARLIE1234	FEBRUARY	ILOVEYOU	MATTHEW7	NICHOLAS	PASSWORD	REBECCA2	STEAMBOAT
ANGELA12	CHARLIE8	FLANAGAN	ILOVEYOU	MAYFLOWER	AMERICANO	PASSWORD	REBECCA1	STELLARL
ANTHONY8	CHARLIE4	FLORENCE	IMPORTANT	MECHANICAL	NOVEMBER	PASSWORD	RECRUIT777	STRAWBERRY
ARKANSAS	CHELSEA1	FLORENCE	INTERNET	MELISSA7	GOFORITNOW	PASSWORD	REDEMPTION	SUMMER
AHHHOLE	CHEVALIER	FLORIDA5	ISTANBUL	MELISSA2	PALISADO	PASSWORD	REDSKIN1	SUNFLOWER
ATLANTA1	CHICKEN2	FLORIDA1	ANNETTEJ	MELISSA3	PANDORA2	PASSWORD	RESPECTFUL	SUNFLOWER
ATTORNEY	CHIPPER1	FLORIDAST	JACKPOT1	MICHAEL8	PARSIFAL	PASSWORD	RICHARD3	SWANSONY
BARBARA1	CHRISTOPHER	FLORIDA7	JEFFREY4	MICHAEL7	PASSWORD	PASSWORD	RICHARD4	SWEETIEPIE
BARSTOW0	CHRYSLER	FLORIDA4	JEFFREY4	MICHAEL1	PASSWORD	PASSWORD	SAMANTHA	SWEETPEA
BASEBALL	COLLEEN5	FRANCIS7	JENNIFER	MICHAEL0	PASSWORD	PASSWORD	SAVANNAH	THANKSGIVING
BASEBALL	COLUMBIA	FRANKIE9	JENNIFER	MICHAELD	PASSWORD	PASSWORD	SEPTEMBER	TIMOTHY9
BATAVIA1	COMPANIES	ACHIEVEMORE	JENNIFER	MICHAEL1	PASSWORD	PASSWORD	SEPTEMBER	TIMOTHY8
1234567U	COMPUTER	CHIQUITOS	JESSICA4	MICHAEL8	PASSWORD	PASSWORD	SEPTEMBER	TIMOTHY9
BEVERLY1	CONNECT	GENTILE1	JESSICA2	MICHAEL3	PASSWORD	PASSWORD	SERVICES	CHARLESTOWN
BIRTHDAY	CONSOLE1	GINSENG4	JONATHAN	MICHELE1	PASSWORD	PASSWORD	SERVICE1	TRINITY9
BIRTHDAY	CORVETTE	GLENDALE	JUPITER3	MINNESOTA	PASSWORD	PASSWORD	SEYMOURB	TUESDAY1
BIRTHDAY	COTTAGE8	GOFORIT1	JUPITER2	MONARCH2	PASSWORD	PASSWORD	SHANNON2	TUESDAY1
BISMARK1	COWBOYS5	GOLIATH1	JUSTINE1	MONDAY	PASSWORD	PASTURE7	SIMPLICITY	TUESDAY9
BISMARK5	DAFFODIL	GOODBYE2	KENNETHS	MONIQUE1	PASSWORD	PERSONAL	SNICKERS	TUESDAY9
BLOOMFIELD	DECEMBER	GRANDMA2	KNOWLEDGE	MUSICAL1	PASSWORD	PLAYHOUSE	SOUTHERN	VACATION
BLOOMFIELD	DESTINY5	GRIFFINS	KRISHNAA	MUSTANG8	PASSWORD	PLEASE	SPECIAL2	VACATION

Figure 1 – NT Domain Sample Password list

These examples show that in many cases passwords are easy to guess. Figure 1 also shows another very common behavior pattern: the appending of a numeric number to the end a password. Knowing that most people do this makes passwords easy to guess. On its own this isn't necessarily weak, however when people change their passwords, many often just increment this number. So if you happen to know an old password of a particular user and the frequency of required password changes, it would be easy to guess their current password. When you add numbers to the beginning or end of a password, they call this a "hybrid" password. Appending or pre-pending of numbers to dictionary words is considered quite weak. There are also some methods of choosing a hybrid password with replacing some characters with numbers or other characters. This is called character substitution, and is much stronger. This will be discussed later.

To fully understand how having some hybrid passwords or dictionary word passwords are weak; we need to discuss how passwords are stored on computers. We will go into what password hash is and how it relates to one-way and reversible encryption.

Passwords are almost never stored in plain text on a computer system. The likelihood of being able to look at clear text file and it being compromised is high. Instead of having clear text passwords stored, passwords are hashed, which is technically a form of encryption and stored. A definition of encryption pertaining to computers is: "To alter (a file, for example) using a secret code so as to be unintelligible to unauthorized parties"[2]. In simple terms, passwords (any mix of numbers, words or special characters) are processed with some sort of algorithm (a mathematical calculation), to get some sort of result. This result is called a password hash. Unlike encrypting files, where you want to return them to original state to be read, password hashing is a one way process. You take the input, process it with some methodology and get a result. This is often stored in a somewhat secure location, sometimes without any additional encryption. In reality you don't care about highly securing a hash, since it's impossible to mathematically calculate the original password. The real use of password hashing comes into play when you want to gain re-entry to a system, in which you put in your password. The computer takes the same proven algorithm that was used to make the hash and run what you just entered against the same process. The computer then compares the results of what you entered to what is stored to see if they match. Whether they match or not, the system will permit or deny access. This is how to protect access with passwords, without having to store actual passwords on a computer. However, this process can be reverse engineered through what is called password cracking.

How does one password crack? Assuming you read above and understand that every combination of words, letters, characters has a unique hash value, can be an easy feat. Or even better, search the internet using a

search engine, such as Google, for password cracking utilities. Basically there are hundreds of applications out there to crack passwords, in our case; we'll look at Windows security, more specifically LAN Manager. This is a good place to start seeing that LAN manager passwords are probably the most common implementation of computerized passwords in use today.

How do these applications work? They take advantage of a computer to process several calculations very quickly. These are much faster than a human. Ultimately, what they all do is take the lists of hashed passwords, take files that contain pre-defined lists of every known dictionary words, then using the same algorithm used to make hashes, they try every combination until they get a match. Some applications can even automatically prepend, append or do common number/character substitution. These methods are known as dictionary or hybrid password attacks. It usually does not take long to go through all possible iterations of dictionary words with or without numbers at the beginning or ends. After dictionaries are exhausted, brute force is the next method.

Brute force tries every single combination of letters, numbers or characters. This can take a significant amount of time. Brute force password cracking boils down to how fast the computer you are using is, and how much time you have available to you. Any password can be cracked with this methodology. The real issue is, if it takes X amount of days to crack a password, can it be cracked if before a goes through a password expiry process and changes it. Some experts apply the ability of password cracking to Moore's law, the faster computers get over time and generations, the easier brute force cracking is.

For reference, some very easily obtained password cracking utilizes are available on the internet, such as:

LC4 by @stake technologies. www.atstake.com
Cain http://www.oxid.it/
John The Ripper http://www.openwall.com/john/

Each of these applications has methods of obtaining password hashes either through password sniffing, or file/registry access to the device. If passwords aren't sniffed on the network, some elevated rights are required to obtain the hashes. Most of these tools use brute force, dictionary and hybrid password attacks to obtain passwords. There is one notable exception to this. This is known as a "rainbow" crack. In the 1980's, a crypto-analyst named Martin Hellman[5], published a paper called "A cryptanalytic time-memory trade off". To paraphrase his work, he theorized that in cryptographic applications, it's faster to pre-compute results of several mathematical equations ahead of time, store the results and re-use repetitively later. Another analyst named Philippe Oecshlin[6] several years later wrote a paper called "Making a Faster Cryptanalytic Time-Memory Trade off" and applied Hellman's[5] work to password hashes. He

described ways that one could pre-compute all the possible plain text and cipher text pairs, and store them in tables. Instead of cracking each password, he used these tables to discover the passwords. It takes a very long time to pre-compute these tables, but this a one time effort. Once they are finished and stored, these tables can be reused for future needs. This is known to be significantly faster.

You may desire to implement policy or technology that forces more complex passwords, which is rarely possible. In businesses, you are often tied to legacy applications and dated standards. It is generally not feasible to re-write every business application in-house to accommodate changes in best practices of password policy. Technology, standards and best practices change over time. For example, in the mid 90's, at birth of the modern internet and the web browser, 40 bit encryption was enough to protect sensitive data. Approaching the mid 2000's, 256 bit types of encryption are de-facto standard. In the days of mainframes and minicomputers, 6-8 letters or numbers was enough. Today, in the home environment, modern software is often used and they rarely have these older password limitations, such as maximum length or different types of characters. Unfortunately, in small business and enterprises, we still live in a world of legacy systems and mixed environments. Most companies still have mainframes that are 25 years old. This is ancient in pc years. Most large companies commonly tie fancy modern graphical front ends and multiple tiered applications with these mainframe back ends. Most companies have no short or medium term plans to migrate off and retire these very old systems. This is relevant to passwords because, for interoperability reasons, they often have to dumb down password security so that applications can communicate with these older devices and databases. When mainframes were designed 25 or more years ago, it was pretty rare to require more than 8 characters, with only letters or numbers allowed. 10 years ago, in the days of DOS and LAN manager networking (mid 80's) 14 character maximum passwords were good enough. Both of these systems had limitations with password length and permitted characters.

With modern and cutting edge computer systems, even though there are rarely limitations to length or character complexity, the average consumer and computer user still likes simplicity. We use passwords of our wives, kids and favorite team. In many cases we use these same passwords reiteratively across many different computer systems. This is an enormous vulnerability. Disregarding the likelihood of easy to guess passwords, if one password gets compromised, you have access to every system they used the same password on. Here is an example for demonstration purposes. Joe User, an average account representative works for a large fortune 500 company. Below is a sample list of places Joe uses passwords for within a 1 week timeframe. We will split of Joe's passwords into personal and business uses.

Personal:	Business:
ATM Machine pin	Network Password
Home Answering Machine pass code	Mainframe Password

Cell phone voicemail pass code
ISP for internet service at home
Personal web based email
Sports scores portal
Personal banking
Credit Cards statements
Email Password
HR website
Voicemail Passcode

This is an example that shows 13 potential passwords. The average user can easily have twice or three times as many. We will assume that that Joe commits these to memory and does not write these down. We estimate that Joe only has only 3-4 different passwords among this set, even though 13 systems were listed. Much of the differentiations are probably different iterations of similar or past passwords.

Here is an actual sample that supports this argument. We took 10 real life random people that are not directly related to the information security field. We polled them on how many systems they used passwords with. We also asked how many they used with home and work computer systems. We documented how many of the same password was used across 3 or more places. Below is a table of actual data. See figure 2.

User	Work	Home	Shared 3 or More
1	5	10	5
2	8	30	29
3	6	6	3
4	8	26	7
5	3	5	5
6	8	36	4
7	8	10	5
8	5	13	6
9	7	20	8
10	15	50	10
AVG	7.3	20.6	8.2
Rounded	7	21	8

Figure 2 – Number of Passwords in Use

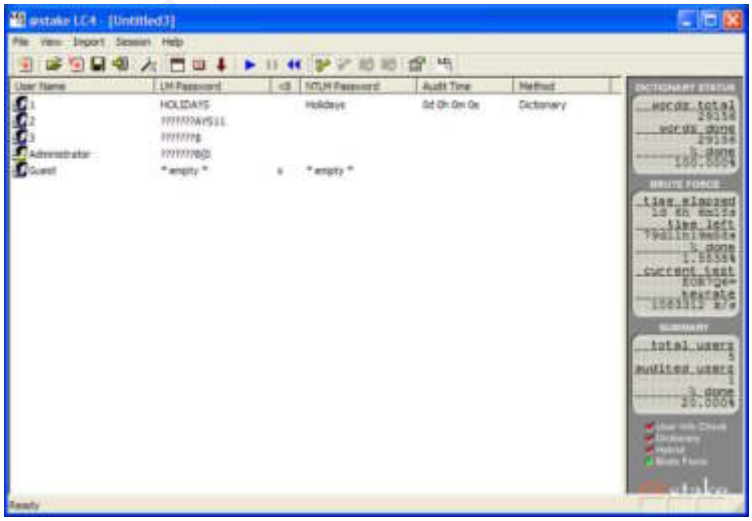
The results are not astonishing. The average of this sample set used 7 passwords for work related systems. They used 21 different passwords on home related systems. Eight of these passwords were common to 8 or more systems. This is commonplace and should concern you. If one password that contains sensitive data is compromised, all systems that use the same password are wide open. Why hack into computers when you can go right in the front door with a password?

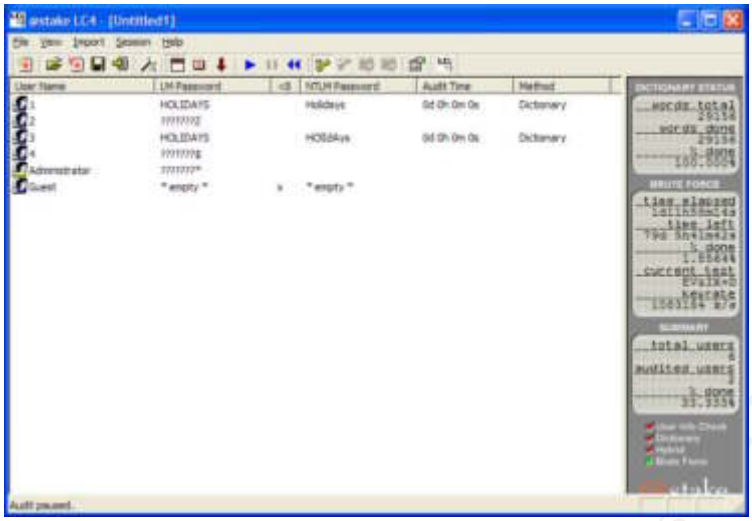
Best Practices: Password Selection

Since passwords are often your front line of defense when securing data, we'll spend some time describing some best practices in how to select a good password. Not all suggestions presented are necessarily feasible to use in every environment. There are often technical limitations within systems which may prevent using such passwords.

One methodology is using pass phrases or sentences instead of dictionary words. Here is an example: "My dogs name is spot". Instead of a single dictionary word, we take several dictionary words and concatenate them together. This is more difficult to hack. Another alternative is a phrase without spacing, such as "Mydogsnameisspot". This is also much more difficult to guess. These are still a little bit weak, since they only use text characters, but are pretty good about not being cracked by dictionary and hybrid attacks. Capitalization can also be done which can deter guessers, but doesn't stand up to hybrid attacks (which attempt combinations of dictionary words). Another alternative is to take each of the vowels out of the password, like "Mydgsnmsspt". This is not even close to similar to a dictionary word and can only be brute force attacked (every letter combination). That last one is pretty strong also since it is longer than average number of characters long. Another very strong password is where one substitutes characters with other characters, special characters or numbers. An example of character substitution is "wdnsday", "h0lidayz"where we have substituted $ for e's, zero's for letter o, and z for s. These are almost impossible to guess, and extremely difficult to crack on a computer. Below is an actual attempt at two attempts to crack passwords in l0phtcrack (LC4). The actual product interface with relative time required to crack the passwords on a Pentium III-733MHZ PC.

The passwords tested were various iterations of the word "Holidays", such as "H0lida$11", "H0liday$" and an administrator password that had the actual password of "!2#4%6&8(0". A dictionary word was used as a reference sample. Passwords were not acquired other than the reference sample, within a 30 hour period.





These examples show ways to use a password that still has meaning to you, but still maintains a very high level of security. Given that there are often limitations in systems to implement password complexity, there are alternatives.

Alternatives

Sometimes complex passwords aren't feasible to be used within your environment. This leaves you two choices. They are alternatives or supplements to existing weak passwords.

SANS courseware describes the approach of *Defense in Depth* applied to authentication as "something you know, something you have and something that you are". Classically, passwords are often something you know, such as a name or word. Something that you "have" or "are" are key alternatives to strengthening passwords in your environment.

Let's initially investigate the idea of "something that you have". You are at work and your office neighbor over the wall hears you one day, providing your credit card company with your social security number. Another day, he overhears you telling your friend your birthday and where you were born. In many companies, to reset passwords, you need to know your birth city, year of birth and the last 5 digits of your social security number. What will stop your evil office buddy calling up the helpdesk and resetting your password? He will now know how to gain access to everything you have access to. The answer is nothing. Remove the human factor. When you input a password into a system, the computer doesn't know it is you. It only knows the fact that you've entered a word that is processed and has the same value as something on file, thus letting you in.

One way around this exposure, are tokens. Some examples of tokens in the security world are SecurID keys, RSA keys or smartcards. SecurID and

RSA keys are basically small electronic devices that go on a keychain. These devices have a small display with numbers on an LCD screen. These numbers are not randomly generated, but change based upon some secret proprietary algorithm that is a combination of time, serial number of the key, as well as other unpublished factors. This number changes with a certain frequency, about every 30 or 60 seconds. On the server side, if the system is programmed with the serial number of the key, is time synchronized and it can be predicted what they key should be on the user end. This is often used by a person when he enters his password. At this time, it really doesn't matter too much if it's a very weak password. He will then get prompted for another password, which contains the number that was generated from the keychain device. This is usually a predetermined user-selected PIN + number that is displayed the key. This means if you do not have the RSA or SecurID key on your person and know the PIN, you cannot gain entry to the system. Smartcards is another similar token. Smartcards do not have a display, but they contain a microprocessor that can be used to store information, that is very difficult to forge or replicate.

The next isn't a token, but along the lines as SANS courseware states: "What you are". It is something that is unique to your person. This isn't something in your mind, but often some physical attribute. These are biometrics. In this case, this can be fingerprint, your eye (retina), your voice or brainwaves. Some biometric authentication methodologies are easily defeated; voice recognition is one of these. High fidelity tape recorders can easily reproduce someone's voice, if the speech patterns required ahead of time are known. Fingerprinting can also be defeated with very expensive equipment, along with graphite powder and adhesive tape. Retina scanners are very difficult to beat, but are extremely expensive to implement. Each of these items has pros and cons, as far as cost, and ability to be compromised. In most cases, along with a pass phrase, biometric methodologies do serve their purpose well as a means of secondary authentication.

Do we necessarily have to implement any hardware to have a more secure environment? The answer is: No. Use two factor authentication and layered approaches to passwords. Instead of moving towards single sign on, move away from it. Have different passwords for different systems. Require multiple logons onto a system. Perhaps have 1 password to access a timekeeping/expense reporting system and have another password for actual approvals. For a human resources website, use pass-through authentication to get to a portal, but if you access any private information, or personal confidential data, require additional layers of password protection. Single sign-on is a very big topic for many companies, as it's easier for the end user to use, but in reality, a HUGE security risk goes along with this. Just remember: there are tradeoffs for numbers of passwords and more complex passwords, these tradeoffs relate to time, expense, and support costs.

If you are going to ask your users to keep a high number of complex passwords, there is software to mitigate some of the support calls relating to users forgetting passwords. These are password manager applications. These applications can store several desperate passwords in a highly encrypted database for later retrieval. Some even provide a methodology of auto filling web forms where you may have stored them in the past. This way you only have to remember fewer passwords and can make them as complex as needed. There are several that run on PDA devices etc. Password managers use secure ways to store the data. The encryption used is often state of the art and beyond a level for the most paranoid. In paper "Options for Secure Personal Password Management"[7], Hugh Ranalli put together a discussion explaining these applications, how they work, how to use and how to select one. A couple examples of these applications are:

- Oubliette: (http://www.tranglos.com/free/oubliette.html)
- Passsafe: (http://www.schneier.com/passsafe.html)
- Password Manager (http://www.symantec.com/passwordmanager/)

Costs of Passwords vs Alternative Technology

There are several costs, both realized and unrealized, associated with implementing technology to assist authentication. We will try to give you a brief synopsis to compare between different methodologies.

The first is the technology itself. RSA or SecurId tokens, the small device that goes on a keychain, are the first example of this. The costs are often moderate to high. Servers can cost several thousand dollars themselves to support this solution. There are multiple servers to deal with failover when there is hardware failure. There are software licensing costs along with consulting and maintenance agreements with the software vendors. Here is an example of basic items and contrived figures to show an example. The assumption is you will be implementing this solution for a small company, of 35 employees. You'd like to implement RSA keys along with standard network authentication. A server could cost 3000-6000$ from any vendor. You probably want two, in-case of a failure. The tokens themselves could be 40-50$ a piece per employee. There will be a percentage of damaged or lost keys. There is cost of time pertaining to consulting services for implementation (2 weeks @100$/hr), plus $5,000/yr for a software maintenance agreement. Granted, these numbers are estimates, which may underestimate the true cost. General this added up to ~$25,000 for the first year to implement a solution. This is quite expensive and not necessarily realistic for a small company. There is also intangible support costs associated with any of these technical solutions. The general user base is only partially computer literate. There is cost associated with ramping up and maintaining technically trained helpdesk. This is assuming you can also get your users to carry their token with them at all times that they may need to access the network.

The next alternative is biometric. Biometrics could be fingerprint scanners, retina scanners, voice recognition, hand shape identifiers. These are exponentially more expensive to implement than a token solution. The hardware cost of individual hardware to read a fingerprint, or scan eyes is extremely expensive. This does offer a higher level of security in most cases. Since you cannot share your eye with a colleague, it provides for less false positives or false negatives. These solutions do have their faults. For example, if you use voice recognition, you may have a cold and may not be recognizable enough to permit access. Digital recorders are cheap these days thus potentially your voice can be recorded and re-synthesized on a computer. There are even solutions to forge fingerprints with latex or gel called a "gummy finger"[9].

Mitigation Strategy

Biometrics and token solutions are strong secure alternatives, but they have a large cost associated with them. Most companies are unable to fund these solutions. Training your user base on how to select strong passwords is an inexpesive, effective way to protect your environment. However, this is hard to enforce, so we need to look at a couple ways to mitigate some of the risks associated with the gap though policy and some other inexpensive it implement technical solutions.

The first is password policy. In most computer systems, there is the ability to force users change their passwords at some frequency. Even if passwords are weak, changing with frequency does mitigate some risk so that if someone knows your password, they will not be able to gain entry with your information if they don't know the latest password. Many enterprise organizations use 45-60 day password expiry on their systems that people log in daily. Frequent password changes also help reduce the risk that passwords might be compromised if being brute force cracked. Brute force cracking can take several days or weeks - if there is a password change during this duration, the attack will be cracking the old password and security is still maintained.

Account lockout is also another good way to reduce risk. If a person makes an attempt with a bad password, after a certain number of bad attempts the account will get locked out. Many systems can unlock the account out after a certain amount of time. Have caution: if a hacker attempts to try several user ID's and passwords and the systems are set to lockout without timed un-lockout, there will be a denial of service condition. They may be able to lock out many accounts and impact the ability of administrators to deal with the volume of calls of lockouts.

Conclusion:

Passwords are often the first and last line of defense in maintaining security in a computer system. The problem with passwords is human nature.

For passwords to be easy to remember, people compromise and often use the same passwords across similar systems. These passwords are almost always something of meaning to them. These are often words that relate to family, hobbies, or personal interest. We looked at how crack passwords, so one would have a better understanding on how to use and select stronger passwords. Hopefully, one would realize with a computer, programs from the internet and minimum technical knowledge, they can compromise security of most computer systems. The only sure way to mitigate this risk is user education and setting policy. With a very over-simplified but very relevant discussion on how passwords are stored and used, a better understanding could be had and applied to selecting better passwords. If money is no object, biometrics and tokens are good technical solutions that can mitigate some risk. There are also computer technical solutions like passwords storage applications or applications to force password complexity. The bottom line: train your users to make their passwords hard to guess. “Since many people use a single password for most of their business applications, the password becomes a kind of universal key once it's in the wrong hands”[8]. Don’t use the same passwords across multiple systems. Also try to implement some computer enforced password policies such as account lockout or password complexity checking to remove some of the “human factor”. The bottom line is that a small amount of user training will reduce these risks mentioned significantly.

References

1. Merriam Webster Dictionary
   URL: http://www.merriam-webster.com/cgi-bin/dictionary?book=Dictionary&va=password

2. Dictionary.Com
   URL: http://dictionary.reference.com/search?q=encryption

3. Nash, Randy – The Weak Link
   URL: http://www.net-security.org/article.php?id=13

4. Advancial Credit Union
   URL: https://www.advancial.org/ATM.html

5. Hellman, M. E. A cryptanalytic time-memory trde off. IEEE Transations on Information Theory. IT-26:401-406, 1980.

6. Oechslin, Philippe, Making A Faster Cryptanalytic Time-Memory Trade-Off Laboratoire de Cryptographie, EPFL
   URL: http://lasecwww.epfl.ch/pub/lasec/doc/Oech03.pdf

7. Ranalli, Hugh, Options for Secure Personal Password Management – October 22, 2003
   URL: http://www.sans.org/rr/papers/index.php?id=1287

8. Verisign Austrlia Pty Ltd
   URL: http://www.verisign.com.au/whitepapers/enterprise/trust/trust2.shtml

9. Lemos, Robert, The hacker's got the gummy touch. May 16, 2002
   URL: http://news.com.com/2100-1001-915580.html