Global Information Assurance Certification Paper

Password Insecurity:
Securing a system with multifactor authentication

GSEC Practical Assignment v1.4b
Michael S. Kun
Oct 17 2004

ABSTRACT:

As technology advances, the ubiquitous password is becoming insufficient for controlling access in many environments. This paper will show why passwords are no longer a good form of access control, and discuss other methods of single-factor authentication that provide for greater security.

In addition, this paper will also discuss two-factor authentication and show why this kind of authentication has many advantages over single-factor authentication.

Technologies discussed will be passwords and PINs, tokens and biometric scanners.

INTRODUCTION

Authentication is one of the most important aspects of security. Regardless of how tightly-locked down a system is, the information is useless without some means of controlling who can access that data.

Authentication remains a tricky issue for one primary reason, it must interact with the end user. Systems administrators, programmers and other technologically savvy individuals often understand the issues surrounding authentication and are willing to deal with many of the difficulties inherent in the process, such as memorizing complicated passwords and passphrases and using different passwords for each system. Many less skilled users, however, view authentication as a nuisance and try to ignore or circumvent the issue entirely. Advancing technologies, however, are giving administrators the ability to increase security on their systems even while making the process of authentication easy for the end user.

SINGLE FACTOR AUTHENTICATION: PASSWORDS

The vast majority of today's computer authentication revolves around single factor authentication. The user supplies a credential that only he knows or has and, if it matches that the system expects, the user is granted access. The most ubiquitous of these is the password.

Password have long been an ever-present reminder of security. They are widely used for many reasons; they are portable, easy to use and simple to change. However, passwords, as a single-factor of authentication, are rife with weaknesses that will only grow worse as computer power increases and new advances in processor speed, drive storage and memory make computers more

and more powerful.

It is important to understand the passwords are not only a technological construct. They are closely linked with human behavior, which allows hackers to make certain assumptions and allows new security holes to develop. Papers such as *The Strong Password Dilemma* show how the complexity of passwords and number of restrictions on the content of the passwords have increased to compensate for raw computing power, the theory being that a sufficiently complex password will take too much time to crack and is "safer" then a less complex one. This behavior, although theoretically leading to a strong password policy, is, in fact, detrimental to it. As password complexity requirements rise, the tendency for the average user to forget the password increases.[1] This leads to the user doing something, like writing down the password and placing it an an easily accessible place, that abrogates the policy in its entirety. This kind of behavior makes it easy for even a casual visitor to uncover the password and gain access to the system.

Human beings can also be deceived into giving up their passwords. Social engineering is a favorite hacker technique and can be as simple as a intruder contacting a staff member and, posing as a helpdesk or system administrator, request the username and password of an individual under a false pretense. Although many security administrators are aware of this tendency,and present training to overcome this tendency, it is difficult to overcome. A recent study conducted by *Security Pipeline,* showed that over 70% of individuals on the street would divulge their passwords to a complete stranger for a candy bar.[2] A hacker using social engineering could stake out the target office building and easily collect many passwords in a short amount of time.

The same study shows a more ominous facet of password security. The study showed an alarmingly high percentage of people used easy to guess passwords such as family names, sports teams and pet names. When coupled with cracking technologies this fact can be used to quickly pare down a list of suspected passwords. As explained in *The Strong Password Dilemma,* if we use an example of a four digit numerical password, such as a bank account PIN. Although theoretically there are about 10,000 possible combinations, by condensing the list to a subset that a human is likely to pick, for example a significant date, we can reduce the list to 366 likely combinations.[1] Since it is likely that people with choose passwords that are easy to remember, we can reduce down the number of possible combinations by basing it off a list of words. This is the basis of the dictionary attack, a common brute-force attack wherein only words specified in a database of existing words, English or otherwise, are used, rather than all possible combinations of characters. Many modern password crackers that use dictionary type attacks also allow for other permutations, including common misspelling and substituting other characters for letters, $ for S for example. Obviously, there is trade off in this method, as the more permutations that are allowed, the longer the scan will take, however, with computing power constantly increasing, there is little actual penalty for increasing the number of possible combinations of the characters. This allows a intrudes to use brute-force method with some success against a system.

Brute-force is a method of online cracking wherein an attacker may try to crack an account by trying various username and password combinations under

different protocols, such as POP3, IMAP, and FTP. This kind of attack attempts to mimic how a remote user might access a system and hopes to uncover the proper combination of username/password before the target system notices.

This method of on-line cracking has several disadvantages; primarily, it is very easy to detect. If a system administrator notices that there were suddenly many attempts on a remote account, he can easily conclude that an attacker was trying a brute-force attack. Another disadvantage is centered on the use of account timeouts and lockouts. Many system administrators will enforce durations of lockouts after a certain number of failed attempts. This discourages an on-line attacking for two reasons; first, it raises the profile of the attack when many users contact the systems administrator to have their accounts unlocked, and secondly it slows the attack, as the attacker must wait of the lockout duration to expire before trying again.

Brute force is not the only method of online password cracking. Intruders can also attempt to recover the password as it is transmitted to the server. In many systems, including Microsoft's, authentication occurs at a central server. When a client transmits the data to the server, the username/password combination is sent in cleartext. Only when the information arrives at the server is the password hashed and compared. If an intruder has managed to install a sniffer or a keylogger on the network, he can read the username and password as it is transmitted to the server, circumventing the benefits of the encrypting hash. In a study by PriceWaterhouseCoopers, they showed that within a 15 minute period, a time frame gained by asking to use a phone in an unattended conference room, the testers managed to sniff 10 passwords in cleartext over the network including an administrator password.[3] This kind of test shows that in many networks, especially Windows based networks, passwords are easy to discover. Other systems, such as Novell, are more secure as they encrypt the password before transmitting it to the server, rendering this kind of attack useless.

Another tactic in online cracking is to examine the contents of the pagefile or other caches on the harddrive. A savvy cracker can easily locate a bootable Linux cd online such as Knoppix or Local Area Security (LAS) and use that bypass the native OS entirely. Once on the local system, the attacker can use the tools on the disk to read areas of the harddrive, such a pagefiles and system caches and search for likely password strings. An experienced hacker on a Windows system can even accomplish the same thing with a simple hex editor, small enough to fit onto a floppy or USB drive, simply by booting the computer into safe mode and then scanning the pagefiles on the local drive.[4] Although this technique does require an attacker gaining physical access to a location, the attacker only needs to do this once; once the password is discovered, the network can be infiltrated remotely and other techniques can be used to further compromise the network without the need to worry about circumventing other on-line security and authentication measures.

An intruder can also use techniques do not involve realtime cracking. Offline password cracking can be done if an intruder gains access to the files to stores the passwords on a system. In both Unix and Windows systems passwords are stored in a dedicated hidden file. Within Windows this is the SAM file; in Unix systems it is the passwd file. In both cases the passwords are stored

using non-reversible encryption. The password is encrypted, and, even if the file is recovered, there is no way to reverse the encryption to get the username/password combination in cleartext. In order to authenticate the user, the system encrypts the supplied password with an identical algorithm and compares the result to the stored value. If the value matches, the user is granted access. Unix systems add an additional layer of security by prefixing each password hash with a salt, a random value that must also match. This adds an additional 4096 values that must be checked before a password match can be computed. The Windows SAM file does not use this hash and so is generally quicker to crack.[5 6 7]

Offline password cracking involves gaining access to the passwd or SAM file, either remotely or by penetrating physical site security and then analyzing the file at another location. Once the file is cracked, the attacker can then log on remotely using any account they desire, including root or administrator. The advantage to this method is that once the file has been acquired, the attacker does not have to rush or try to cover his tracks. Using any number of tools, including L0phtcrack or John the Ripper, the attacker can slowly work though random strings of characters, or multiple dictionaries of common passwords until the file is cracked.

A recent development in the area of offline password cracking, rainbow tables, is further eroding the protection of most passwords. Rather than hashing and comparing each value to a given password file, rainbow tables pre-compute millions of possible password hashes and store the tables in memory, allowing millions of possible combinations to be checked in a matter of minutes. By doing the heavy computing an advance, the program only needs to compare the pre-computed hashes with the target file to quickly find a matching password.[8] This technique is based off a technique first postulated by Martin Hellman in the 1980's where he described a process where precompiled password hashes were stored in memory and could be used to reduce the time it took to acquire a username/password match. At the time his paper was published, the memory that was needed to crack a given password was prohibitively large. The limitation is described as a time vs. memory trade-off. The more memory used, the less time the cracking would take. However, in the past two decades, memory has become plentiful and very cheap. As a result this trade-off of time vs. memory has less value and more precomputed hashes can be stored on the average system. A further enhancement was a development by Philippe Oechslin that shortened the calculation process and optimized the tables. This resulted in a rainbow table that contained 99.9% of all alphanumeric Windows NT password hashes in a set of tables that were just over 1.4 GB in size. This tableset was capable of cracking a given password in about 13 seconds. Using this technique, an attacker with a significantly powerful computer could search though and crack any password in seconds once the tables are compiled. This represents a new threat to the old style of access control. System administrators must now be aware that no matter the complexity of the passwords they create, most can be quickly and easily cracked if the attacker can gain access to the password file.

It may be questioned as to why passwords should be used, based on all their shortcomings. The basic password as a method of access control, however, remains a proven method for low-level security and one that has many

advantages when a part of defense in depth. Passwords remain portable, easy to implement and easy to change. They represent a barrier the a casual intruder may avoid, suspecting this attempts to thwart the password scheme may tip off an alert systems administrator that an attack is imminent.

SINGLE FACTOR AUTHENTICATION: BIOMETRICS

Passwords occupy one dimension of authentication, “something you know”. Another dimension is “someone you are”, a way of authentication that proves the user is who he says he is. Biometrics is the most common implementation of this factor

Biometric authentication is a area that is rapidly maturing as an alternative to the basic password. Biometric authentication falls into several categories, including fingerprint scanners, iris scanners, rental scanners, hand geometry and facial geometry. In all biometric systems, the user is enrolled into the system by entering the data, a fingerprint, iris scan, etc, into the system. Once enrolled, the user needs only to scan the same body part in order to be authenticated. Unlike passwords, this enrollment process is much more complicated, requiring someone experienced in enrollment procedures to guide the user so enrollment can be completed correctly. With some biometric systems multiple scans must be completed for enrollment to be successful. This adds an additional level of administrative overhead not needed when using other types of systems, such as enabling a password or issuing a token. It is important to note that the scanner does not examine the whole object, but only selected points to compare to a stored template. By selecting unique and easily distinguished points, factoring in their relationships to other points and storing those measurements as a string of characters, the system can create a passcode that is far longer and more complex that a human being can comfortably remember. However, this method can cause problems if the angle or placement of the finger, hand or eye on the reader is not similar to the position the object was in when the user was enrolled in the system.

Since the biometric system is not based on a word of phrase that needs to be written down, it circumvents the problems of social engineering that plague a simple password setup. There is no way for a user to “give up” the key to his access. Furthermore, biometric systems can better show that the user was the one that authenticated, allowing a greater level of accountability within the system.

Biometric systems are not without weaknesses, however, and some of these weaknesses can be exploited by intruders into the system. The primary weakness of a biometric systems is that if the system is compromised it is difficult to change a users authentication. Tokens and password can be changed, but in biometric-based system there is only a limited number of points of authentication. A user only has two eyes for iris scanners, or ten fingers to fingerprint scanners.

Another weakness is the fact that the system must be based on the software of operating system that runs the computer. If the software is bypassed, by either booting into safe mode or with a bootable CD, the software

the runs the scanner can be bypassed entirely and allow the intruder access to the local system.[4]

Sniffing attacks, like those discussed earlier, are more difficult with biometric scanners. The scanner does not record an image of the body part, but rather maps unique points on the object. These points are then encoded into a series of numbers or ASCII characters and hashed. This technique makes it very difficult for an intruder to recover the original “key” that was used to create the code.[9]

Early scanners suffered a weakness that allowed an intruder to enroll himself into a system. Once the intruder determined the kind of scanner being used, he could obtain an identical scanner and enroll himself into a computer that he owned. Once the hash was established, he could then upload that data into the system he wanted to compromise. Once the data was on the system, the intruder was essentially enrolled in the system and could authenticate to it whenever he wanted. Biometric companies have since taken measures to thwart this kind of attack.

Another flaw with biometric scanners concerns the subset of users the fail to enroll in the system. Each of the methods has a class of people that will not be able to enroll in the system. Fingerprint scanners and hand geometry scanners, for example, are difficult for people who are paralyzed or with limited mobility to use. Iris and retinal scanners are difficult for users who are blind or with sight impairment to use.

Although difficult to circumvent, it is not impossible to fool biometric scanners, indeed, early scanners were laughably easy to spoof. Technology has advanced, however, so has the ability of intruders. Many people are familiar with fingerprint and iris scanners. As a result, these technologies may be easier to circumvent due to the volume of information written about them. Fingerprint scanners have been shown to be vulnerable to “gelatin attacks” as described by Tsutomu Matsumoto in his paper “Impact of Artificial "Gummy" Fingers on Fingerprint Systems”. In this kind of attack an intruder captures a latent fingerprint and uses it to carve a piece of circuit board which is then used to mold a fingertip out of gelatin. Since gelatin has a capacitance similar to human flesh, a properly used fake fingertip can be used to thwart a fingerprint scanner.[10] Further refinements in this technique that use thin latex or other materials may fool even sophisticated sensors the measure moisture, body heat or pulse by allowing the sensors to read the moisture, heat and pulse of the intruder's finger, but scanning the ridges of the fake fingerprint. In answer to these threats some organizations are developing ways to ensure the only a live fingerprint can be used. Techniques that use micro-electrical current or radio-frequencies to examine the fingerprint below the surface of the skin, within the living cells themselves would be far more secure against this kind of attack.

Iris scanners are also a rapidly maturing technology. While early versions were easily fooled by holding a picture of an eye up the the scanner new versions are proof against this kind of spoofing.

Although not impossible to duplicate an individual iris pattern, it cannot to ruled out that an sufficiently sophisticated hacker could capture someone's iris pattern. While early sensors could be fooled by high-quality images held up to the scanner, modern iris scanners not only look at the point around the iris, but

also check for depth of the eye.

Facial recognition scanners are relatively new and remain inaccurate. The idea of checking points on a face, such as the distance between the eyes, or the length of the nose seems sound enough, but these details are easy to forge, even with something as simple as a good photograph or a video capture. Without any way of ensuring that the individual is actually present, facial scanners are a weak implementation of biometric authentication at best. Further devaluing facial recognition is the simple fact that people's faces change. Growing a mustache or gaining weight may be enough to cause the system to reject a user, a flaw not found within the previous two methods. A live test run at Palm Beach airport showed that facial recognition systems failed over 50% of the time.[11]

Likewise, hand geometry, which examines the size and shape of the hands and fingers is vulnerable to changes in the individual. People with joint problems or poor circulation can find that the size and shape of their fingers changes over time. The architect of the system must then decide if the scanner should be set with more forgiving parameters, and risk the possibility that a intruder can spoof the scanner or he must be aware that individuals may have a hard time with the scanner.

SINGLE FACTOR AUTHENTICATION: TOKENS

A third dimension of authentication would be “something you have”. This method requires the user have a device or object before being allowed access to a system. Token devices like these require that either a card or token be scanned before the user can access a system. Some devices include wireless components so that a user carrying a token only need to be in proximity to the computer for the token to be recognized.

Token systems like these, however, have many weaknesses. Token only systems are no better than the password they are intended to supplant. Although a far more complex passcode can be encoded into a token, making it difficult to brute-force such a code, the real weakness of the system is with the token itself. A token can be easily stolen allowing a intruder easy access to a system without any real barriers. It is not even necessary for the intruder to devise a method of tricking the target out of the information. The token can be stolen and the system compromised before the target even knows it is missing. That fact, coupled with the costs and administrative overhead associated with managing physical tokens makes this a difficult system to implement and secure.

TWO FACTOR AUTHENTICATION

Since all these systems have their own flaws, it would make sense that combining them into a layered defense would make sense. This two factor approach is in fact where many security architects have turned to in order to better secure their systems.

Such two-factor authentication has been in place for years. Consider the bank card, a classic example of two-factor authentication. Although the password, the PIN, is usually only four digits, it is useless without the token, the bank card itself. Only by possessing both items, the “something your have” and

"something you know" can a user be authenticated. This provides an added level of redundancy, since neither device alone will allow an unauthorized individual access. A thief might steal the card, but without knowing the PIN cannot gain access and vice versa. The fact that there is a physical token that intruder must posses allows administrators to relax the rules placed on password selection and reduces the likelihood that the password will be written down. An additional benefit to two-factor identification is that is allows simplification of policy through the use of a single sign on. Rather then remember multiple passwords for different systems, a properly executed two-factor infrastructure allows the users to use a single token and associated PIN to gain access to multiple systems throughout the enterprise. This, again, reduces the likelihood there will be passwords written down.

This type of two-factor identification can be linked with either token or biometric technology. Tokens are usually a USB based key that must be attached to a device before the user is allowed to proceed further. There was a fear the tokens of this type could be compromised themselves, allowing an attacker the captured a key to unlock and read the data within. Modern tokens, however use ARM micro controllers to essentially turn off the restricted memory, rendering it undetectable by most software until the micro controller authenticates the user. Once the user is authenticated through the token, usually by use of a PIN, the user is allowed access to the restricted memory and the rest of the system. [4]

This kind of authentication has many advantages over the traditional password structure. It is highly resistant to social engineering, even if a users PIN is discovered, there is still no way for the intruder to gain access to the system without the token. This kind of system retains many of the advantages associated with a traditional password scheme, among them portability, reliability and ease of administration. In addition there is a greater degree of accountability for access made by the user because of the multiple factors the user is required to have. Due to the number of tokens available, this kind of system is easily customized to the needs of the enterprise. Tokens can be smart USB tokens or smart cards .

Unfortunately, most implementations of USB tokens have a critical weakness, ironically one shared by some the password systems they are meant to supplant. In many solutions the weakness is the software the token uses to interface with the client computer. As Tom Bowers writes in *Infosecuritymag,* passwords or PINs should be encrypted prior to being passed to the token. Without this fail-safe, the plaintext password can be found on the hard drive and recovered, a situation identical to the one that traditional password implementations share. In addition, many implementations of token software do not lock out safe mode on Windows systems. If an attacker can boot the machine to Window's safe mode, the token software will not load and allow the attacker unrestricted access to the local machine making it easy to search the pagefile and caches for PIN information.[12]

This weakness would allow an intruder to steal a laptop and a token then, by bypassing the native OS as discussed earlier, locating the PIN on the harddrive and have both factors needed for authentication.

Furthermore, there is another dimension to the weakness of the safe-

mode bypass. Since this method of two factor identification is often employed to lock out laptops in order to preserve confidential company data, a hacker can use this exploit to gain access to that confidential data through safe mode and bypass the token entirely. This weakness highlights the necessity of planning a defense in depth and implementing folder level encryption to keep valuable data secure.[12]

Biometrics can also be used in two-factor authentication. Although the implementations of biometric scanners themselves must be examined with the issues discussed above, biometric two factor identification is a convenient way to both enhance security over passwords and other single factor authentication schemes and also to preserve accountability. The introduction of biometrics, particularly iris scanners makes it easy to show that a user on the systems was the one that logged in. As shown earlier, iris scanners are very difficult to hack and, when combined with a PIN or password as a second factor, it allows for a very secure system and a high degree of certainty that the individual that logged on to the system is the one authorized for it. This same assurance can also be provided by mating a token system with a biometric scanner. In this instance, the token stores the information about the users fingerprint and also uses specific algorithms to generate the passcode. This type of deployment both preserves the two-factor authentication as well as minimizes the possibility that an intruder can bypass the OS. If properly programmed, the token will not allow the system to boot without authentication and, since all operations are performed on the token itself, there is no way for an attacker to discover any data within the cache or pagefile on the system.

Although there are no iris/token devices currently available, the proliferation and increasing complexity and sophistication of small video devices such as cameras and webcams indicate that it is possible to mate a video capture device with a token and achieve a very high level of security.

One thing that is important to note is that when referring to a system we are referring only to local computer and not to a network. With of the combinations of authentication systems that we have discussed previously. All share a single weakness. Although providing a secure logon to the local system credentials still must be sent over the wire and this is the responsibility of the operating system and not the token itself. Some operating systems, as mentioned earlier, send username and password combinations in clear text, resulting in the possibility that the date can be compromised if an intruder has installed a sniffer on the network. Preventing this requires the deployment of PKI , or other kinds of encryption solutions that may not be related to the authentication solutions discussed here.

In addition to two factor authentication, these different authentication options can be combined into three or more factor authentication, a protocol referred to as multifactor authentication. Very high security installations may combine biometric, token and password technologies together into the single cohesive authentication and security solution. As an example the Relativistic Heavy Ion Collider, or RHIC, a particle accelerator located in New York completed a case study showing how incorporating these methods could create a easy to use and hightly secure enviornment. In order to control access to the primary control room, a restricted high security area, a plan was implemented

which used all three of the previously discussed methods of authentication. Personnel at the RHIC are required to authenticate via iris scan as well as possess a token and a password. Only the correct combination of token and password will unlock the biometric iris scanner and permit the user to authenticate with the system. Only once all these elements are correct, will the user be allowed access to the high-security areas of the particle accelerator.[13]

CONCLUSION

Technology is both beneficial and detrimental to maintaining appropriate access controls to any system. Increases in computing power have rendered old-fashioned ways of authenticating and controlling access nearly obsolete. With proper technology an intruder can quickly determine a large number of username and password combinations. If this is the only method in-place to secure a system, that system can be quickly compromised. However the other side of the coin shows that advancements in technology has provided system administrators and security architects with many new tools for controlling access to their systems. Tokens and biometric scanners allow systems to be secured with pass codes that are not only extremely complex and difficult to crack, but through the uses of these tools can be quickly and easily accessed by nearly any user. By combining means with passwords that now can be simplified and made easier to remember by the user and administrator has a very effective method to factor authentication. This type authentication makes it very difficult for intruders to access a secure system, requiring that they both possess some physical device or object and also knowing what that user's pass code is. In the case of biometric authentication, the object the intruder must be able to obtain may be difficult or completely impossible for him to acquire and extremely difficult to mimic or spoof.

In addition, this type of authentication allows a high degree of accountability so that administrators can be assured that they are keeping track of the users in a highly accurate manner. At facilities that require even higher levels of access control, multiple factors can be combined in a layered security approach. This approach makes it even more difficult for an intruder to spoof or circumvent the authentication system and yet remains very easy for users to interact with and authenticate with, requiring only a card or token or a glance at a camera or a touch of a sensor and the entering of a simple PIN. Other emerging technologies can also be integrated into this layered access control approach, including technologies that base their templates on the rhythm of a users typing adding another hidden layer of authentication where, in addition to typing in a password or PIN, the user's unique typing patterns are also incorporated as another level of access control

Although techniques for two and multifactor authentication to require your some foresight and planning in order to implement effectively. Such a deployment is critical to defense in depth, and provides far more security then can the easily stolen or otherwise circumvented single factor password solution.

References

[1] Smith, Richard E. “The Strong Password Dilemma.” Authentication: From Passwords to Public Keys. The Center for Password Sanity. 2002 URL: http://www.smat.us/sanity/pwdilemma.html (17 Oct 2004)

[2] Wagner, Mitch. “Will Trade Passwords For Chocolate”. Security Pipeline. 19 April 2004. URL: http://www.securitypipeline.com/news/18902074 (17 Oct 2004)

[3] Lobel, Mark. “Case for Strong User Authentication.” 2002. URL: http://www.pwcglobal.com/extweb/manissue.nsf/2e7e9636c6b92859852565e00073d2fd/728d168e9e5cce04852566fd00665839/$FILE/case_for_strong(pwc)_wp.pdf (17 Oct 2004)

[4] Bowers, Tom. Telephone Interview. 8 Oct 2004

[5] “The Hack FAQ.” Nomad Mobile Research Center. 2 Aug 2003. URL: http://www.nmrc.org/pub/faq/hackfaq/hackfaq-04.html (17 Oct 2004)

[6] “The Hack FAQ.” Nomad Mobile Research Center. 2 Aug 2003. URL: http://www.nmrc.org/pub/faq/hackfaq/hackfaq-13.html#13.4(17 Oct 2004)

[7] “The Hack FAQ.” Nomad Mobile Research Center. 2 Aug 2003. URL: http://www.nmrc.org/pub/faq/hackfaq/hackfaq-28.html (17 Oct 2004)

[8] Zhu Shuanglei. “Project RainbowCrack.” 19 Sept 2004. URL: http://www.antsight.com/zsl/rainbowcrack/ (17 Oct 2004)

[9] Davis Russ. “Biometric myths: six of the best.” 8 July 2004. URL: http://www.itsecurity.com/papers/isl1.htm (17 Oct 2004)

[10] T. Matsumoto, H. Matsumoto, K. Yamada, S. Hoshino. "Impact of Artificial Gummy Fingers on Fingerprint Systems." Proceedings of SPIE Vol. #4677. Optical Security and Counterfeit Deterrence Techniques IV. 2002.

[11] Greene, Thomas C. “Face recognition kit fails in Fla airport.” The Register. 27 May 2002. URL: http://www.theregister.co.uk/2002/05/27/face_recognition_kit_fails/ (17 Oct 2004)

[12] Bowers, Tom. “Token Effort.” Information Security Magazine. July 2004. URL: http://infosecuritymag.techtarget.com/ss/0,295796,sid6_iss426_art864,00.html (17 Oct 2004)

[13] Williams, N.W. Reich, J. “Using Biometric in an Accellerator Personell safety system” 2003 URL: http://conference.kek.jp/wao2003/papers/11p2-3.pdf (Oct

17 2003)