Password Security: KeePass Password Safe

There are a number of recommendations that should be followed in order to use passwords securely. First of all a password needs to be sufficiently long and complex (i.e. drawn from a large set of characters) to make a brute force attack infeasible. Especially in the absence of rate-limiting or account blocking after failed attempts; such as when using poorly developed applications or if there is a possibility of an offline attack [1]. Secondly, dictionary words, common or previously compromised passwords, context related information (e.g. the website name), or easily obtained personal information (e.g. birthdays, phone numbers) should not be used to create a password. Otherwise the password would be vulnerable to dictionary attacks (a type of brute force attack where only likely possibilities are attempted). Finally the password should be kept private, never stored in a plain text file or written down [2].

These requirements are further complicated by the importance of using unique passwords for different services. If one of these services were attacked and the password exposed, the attacker could gain access to any other services using the same password. There are many reasons why this could be possible, some examples include: An application that fails to properly implement password hashing; perhaps using a weak algorithm, or by not using salt, making the passwords vulnerable to rainbow table attacks (a table of precomputed hash chains) [3]. Or misconfigured server logging may capture and store passwords as plain text [4, 5].

The more services that a person uses it becomes increasingly difficult for them to remember good unique passwords for each one. Research by Experian plc found that 'people have on average up to 26 online accounts protected by only five different passwords' [6]. In my personal experience I found that I was using the same password for countless different websites, all using the same email address. A possible solution to this problem was to use a password manager [7], this is a utility that removes the need to remember numerous passwords by storing them in an encrypted database accessible using a single master password[1].

I elected to use KeePass Password Safe [8], it has the advantage of being free and open source, although compared to premium services (such as LastPass) it does require the user to be entirely responsible for storing and synchronising the database file. Fortunately in my situation this was not a problem as I was already was self hosting a Seafile file server. This setup adds a further layer of security because not only is the KeePass database file encrypted itself with its own password, no external parties can gain access to the file because it is only stored on my own encrypted disks and it is synchronised between them over HTTPS.

When using KeePass on a desktop environment[2] I have installed KeePassHttp, a plugin that exposes the database through a HTTP server, making entries accessible to browser plugins such as chromeIPass[3]. Not only does browser integration improve usability it potentially adds protection against phishing: The browser plugin will only extract entries from KeePass when the present URL matches the database value, therefore the chance of being fooled by a phishing site is reduced because I would notice that the password fields are not being populated [7].

There is unfortunately one criticism that can be made against password managers in that they exacerbate the risk of a compromised device[4]. In the absence of a password manager the attacker would have to wait until I signed in on that device (if ever) to each service they are targeting. Whereas as soon as I sign in to KeePass they would gain immediate access to all credentials, including those which may never have been entered on that device [9]. Regardless both scenarios highlight the importance of maintaining good security practices on all of my devices; at the point they are totally compromised, having a strong password becomes irreverent.

[1]Password managers may also combine the password with other forms of authentication such as a key file.

[2]If you use a clipboard manager (I am using Ditto for example), an exclusion should be added to prevent capturing KeePass entries, which would otherwise get saved as plaintext.

[3]The server is bound exclusively to the localhost / loopback interface, and a preshared AES key stored in the database is used to encrypt the traffic between the two applications.

[4]Such as where an attacker has sufficient privileges to install a key logger or read the memory of another process.

References

[1] P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, J. P. Richer, N. B. Lefkovitz, J. M. Danker, Y.-Y. Choong, K. K. Greene, and M. F. Theofanos, "Digital identity guidelines: Authentication and lifecycle management," National Institute of Standards and Technology, Tech. Rep., Jun. 2017, Section 5.1.1.2 & Appendix A. DOI: 10.6028/nist.sp.800-63b. [Online]. Available: https://doi.org/10.6028/nist.sp.800-63b.

[2] CERN Computer Security Information, *Password recommendations*. [Online]. Available: https://security.web.cern.ch/security/recommendations/en/passwords.shtml.

[3] P.-H. Kamp, "LinkedIn password leak: Salt their hide," *ACM Queue*, vol. 10, no. 6, p. 20, Jun. 2012. DOI: 10.1145/2246036.2254400. [Online]. Available: https://doi.org/10.1145/2246036.2254400.

[4] C. Cimpanu, *Github accidentally recorded some plaintext passwords in its internal logs*, May 2018. [Online]. Available: https://www.bleepingcomputer.com/news/security/github-accidentally-recorded-some-plaintext-passwords-in-its-internal-logs/.

[5] Twitter, *Keeping your account secure*, May 2018. [Online]. Available: https://blog.twitter.com/official/en_us/topics/company/2018/keeping-your-account-secure.html.

[6] Experian, *Experian reveals the five key factors that make people & businesses more vulnerable to cyber fraud*, Dec. 2016. [Online]. Available: http://www.experian.com/blogs/news/2016/05/19/experian-reveals-five-key-factors-make-people-businesses-vulnerable-cyber-fraud/.

[7] National Cyber Security Centre, *What does the ncsc think of password managers?* Jan. 2017. [Online]. Available: https://www.ncsc.gov.uk/blog-post/what-does-ncsc-think-password-managers.

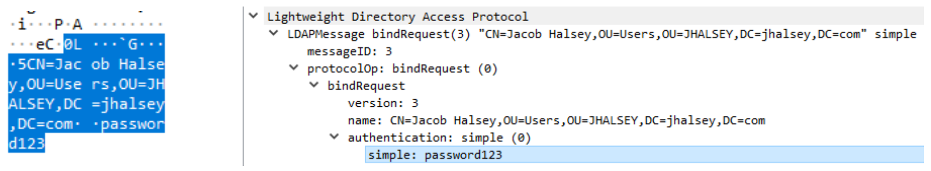
[8] D. Reichl, *Keepass password safe*. [Online]. Available: https://keepass.info/.

[9] R. Lemos, *Malware's new target: Your password manager's password*, Nov. 2014. [Online]. Available: https://arstechnica.com/information-technology/2014/11/citadel-attackers-aim-to-steal-victims-master-passwords/.

Using LDAPS for centralised authentication

I currently have two Ubuntu servers, one running Seafile (a file server) [1], and another running Gogs (a Git service) [2]. Instead of having to maintain multiple sets of credentials, I had configured these servers to centrally authenticate users using LDAP [3] with my Active Directory server. All three servers are connected to the same Ethernet network switch and VLAN. The network does not use any form of access control (such as 802.1X) and it is also accessible via Wi-Fi using a pre-shared key.

I realised that this setup created a potential security weakness because by default LDAP traffic is not encrypted [4, sec 6.1] [5]. For example, figure 1 shows part of the LDAP data that is transmitted as plaintext over the network when I attempted to login to the Gogs server:



```
·i···P·A ········
···eC·0L ···`G···
·5CN=Jac ob Halse
y,OU=Use rs,OU=JH
ALSEY,DC =jhalsey
,DC=com· ·passwor
d123
```

(a) Text

```
⌄ Lightweight Directory Access Protocol
  ⌄ LDAPMessage bindRequest(3) "CN=Jacob Halsey,OU=Users,OU=JHALSEY,DC=jhalsey,DC=com" simple
      messageID: 3
    ⌄ protocolOp: bindRequest (0)
      ⌄ bindRequest
          version: 3
          name: CN=Jacob Halsey,OU=Users,OU=JHALSEY,DC=jhalsey,DC=com
        ⌄ authentication: simple (0)
            simple: password123
```

(b) Details

Figure 1: Wireshark capture of an LDAP bind request

Although during normal Ethernet operation a frame sent from one node to another node will not be seen by other nodes on the network (aside from during the initial MAC learning process)[1], there are at least three types of attack that could be used to manipulate the flow of Ethernet traffic between the servers[2], allowing an attacker to intercept the unencrypted credentials. These include: MAC address spoofing, CAM table overflows, and ARP spoofing[3] [6]. While there are various methods to counter these attacks, including various port security tools provided on some Ethernet switches [7, p. 193]. None of these are ideal solutions for my network because the Ethernet switches do not have consistent support for these features, and it would add significant management overhead to maintain their configuration.

To solve this problem I decided to enable Secure LDAP (LDAPS), which uses TLS to authenticate and encrypt communication between the servers. The following steps were required to do so [5, 8]:

1. Creating a root certificate (and key pair), and importing it into the trusted store of the AD (Active Directory) server.
2. Creating a client certificate (and key pair) signed using the root certificate, for the purpose of 'Server Authentication', with the AD server's FQDN as the subject. This certificate is then imported into the AD server's personal store for LDAPS usage.
3. The root certificate then needs to be installed on each of the Ubuntu servers so that they can establish trust of the AD server's certificate [9].
4. The Ubuntu servers should now be configured to connect securely to LDAPS on port 636 using the AD server's FQDN.

Whilst the communication between servers is now secured, I am no longer sure if a centralised authentication setup is the overall best approach. Now that I am using a password manager having to maintain multiple sets of credentials is of little concern to me. I am instead concerned that the attack surface has been increased because the same set of credentials are moving through software products provided by numerous different vendors, if just one of these were to be exploited, then the attacker would be able to gain a set of credentials capable of accessing all the other services [10, 11].

[1]And also when using an Ethernet hub; the network functions as a bus, however these are rarely used today.

[2]There are other types of layer 2 attacks (e.g. an STP root bridge attack), but they are not really relevant to my network configuration.

[3]In actual fact the servers are configured to use IPv6 which does not use ARP, but rather NDP, however the concept is the same.

References

[1] Seafile Ltd, *Seafile - open source file sync and share software.* [Online]. Available: `https://www.seafile.com/en/home/`.

[2] *Gogs: A painless self-hosted git service.* [Online]. Available: `https://gogs.io/`.

[3] K. Zeilenga, "Lightweight directory access protocol (ldap): Technical specification road map," RFC Editor, RFC 4510, Jun. 2006. [Online]. Available: `http://www.rfc-editor.org/rfc/rfc4510.txt`.

[4] R. Harrison, "Lightweight directory access protocol (ldap): Authentication methods and security mechanisms," RFC Editor, RFC 4513, Jun. 2006. [Online]. Available: `http://www.rfc-editor.org/rfc/rfc4513.txt`.

[5] Microsoft Support, *How to enable ldap over ssl with a third-party certification authority.* [Online]. Available: `https://support.microsoft.com/en-us/help/321051/how-to-enable-ldap-over-ssl-with-a-third-party-certification-authority`.

[6] Y. Bhaiji, *Understanding, preventing, and defending against layer 2 attacks*, 2007. [Online]. Available: `https://www.cisco.com/c/dam/global/en_ae/assets/exposaudi2009/assets/docs/layer2-attacks-and-mitigation-t.pdf`.

[7] W. Odom, *Cisco CCENT/CCNA ICND1 100-101 Official Cert Guide, Academic Edition*, ser. Official Cert Guide. Cisco Press, 2013, ISBN: 9781587144851.

[8] P. Mescalchin, *Enable ldap over ssl (ldaps) for microsoft active directory servers.* [Online]. Available: `https://gist.github.com/magnetikonline/0ccdabfec58eb1929c997d22e7341e45`.

[9] Canonical, *Ubuntu manpage: Update-ca-certificates.* [Online]. Available: `http://manpages.ubuntu.com/manpages/bionic/man8/update-ca-certificates.8.html`.

[10] S. Huque, *Ldap weaknesses as a central authentication system*, Oct. 2007. [Online]. Available: `https://www.huque.com/talks/2007-10-LDAP-Authn.html`.

[11] R. Newington, *The ldap 'authentication' anti-pattern*, Mar. 2018. [Online]. Available: `https://blog.lithiumblue.com/2018/03/the-ldap-authentication-anti-pattern.html`.