usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

LanDscAPe: Exploring LDAP Weaknesses and Data Leaks at Internet Scale

Jonas Kaspereit and Gurur Öndarö, *Münster University of Applied Sciences;* Gustavo Luvizotto Cesar, *University of Twente;* Simon Ebbers, *Münster University of Applied Sciences;* Fabian Ising, *Fraunhofer SIT and National Research Center for Applied Cybersecurity ATHENE;* Christoph Saatjohann, *Münster University of Applied Sciences, Fraunhofer SIT, and National Research Center for Applied Cybersecurity ATHENE;* Mattijs Jonker, *University of Twente;* Ralph Holz, *University of Twente and University of Münster;* Sebastian Schinzel, *Münster University of Applied Sciences, Fraunhofer SIT, and National Research Center for Applied Cybersecurity ATHENE*

https://www.usenix.org/conference/usenixsecurity24/presentation/kaspereit

This paper is included in the Proceedings of the
33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

LanDscAPe: Exploring LDAP Weaknesses and Data Leaks at Internet Scale

Jonas Kaspereit[1], Gurur Öndarö[1], Gustavo Luvizotto Cesar[2], Simon Ebbers[1], Fabian Ising[3,4],
Christoph Saatjohann[1,3,4], Mattijs Jonker[2], Ralph Holz[2,5], and Sebastian Schinzel[1,3,4]

[1]*Münster University of Applied Sciences*
[2]*University of Twente*
[3]*Fraunhofer SIT*
[4]*National Research Center for Applied Cybersecurity ATHENE*
[5]*University of Münster*

Abstract

The Lightweight Directory Access Protocol (LDAP) is the standard technology to query information stored in directories. These directories can contain sensitive personal data such as usernames, email addresses, and passwords. LDAP is also used as a central, organization-wide storage of configuration data for other services. Hence, it is important to the security posture of many organizations, not least because it is also at the core of Microsoft's Active Directory, and other identity management and authentication services.

We report on a large-scale security analysis of deployed LDAP servers on the Internet. We developed LanDscAPe, a scanning tool that analyzes security-relevant misconfigurations of LDAP servers and the security of their TLS configurations. Our Internet-wide analysis revealed more than 10k servers that appear susceptible to a range of threats, including insecure configurations, deprecated software with known vulnerabilities, and insecure TLS setups. 4.9k LDAP servers host personal data, and 1.8k even leak passwords. We document, classify, and discuss these and briefly describe our notification campaign to address these concerning issues.

1 Introduction

The Lightweight Directory Access Protocol (LDAP) is the standard technology to query information stored in directories. These directories may contain sensitive personal data such as usernames, email addresses, and passwords. LDAP is also used as a central organization-wide storage of configuration data for other services. LDAP instances manage data within the directory and allow the authentication of individuals and services seeking access to it [48]. LDAP is used in various domains, including email services [6, 43] such as Microsoft Exchange [34], publishing Public Key Infrastructure (PKI) information [29], and enabling Single-Sign-On solutions [24, 56]. LDAP servers are often an integral part of more comprehensive directory services, notably Apple's Open Directory [4], Microsoft's Active Directory (AD) [7, 28], and Red Hat's Directory Server [42]. In some cases, these services are intended for public use and hence meant to be exposed to and accessible over the Internet. More frequently, however, they hold private or sensitive data that should be only accessible to a small group of principals or within an organization's backend network. The secure configuration of these private LDAP servers is of critical importance.

In this paper, we take a closer look at the security configurations and the directories of LDAP servers publicly accessible on the Internet. To the best of our knowledge, despite it being the industry standard, there is no prior study of LDAP at Internet scale besides DDoS amplification analyses using LDAP over UDP [1]. In fact, LDAP has been mostly studied in terms of isolated vulnerabilities rather than as an ecosystem. Consequently, there are no tools dedicated to large-scale evaluation. Existing software to analyze LDAP settings is also usually limited in scope [32, 51, 54].

For our study, we built a custom tool, LanDscAPe, that identifies public LDAP servers on the Internet, assesses their configuration, including TLS setup, and samples the hosted directories. Concretely, the tool scans the IPv4-wide Internet for hosts with common LDAP ports in an open state. It then sends LDAP probes to determine if the service indeed speaks LDAP. LanDscAPe tries all standard-conforming ways of anonymously (without using passwords) binding to servers and, if successful, collects configuration information and requests directory samples. It analyzes the samples for personal data, sensitive configuration values, and possible passwords. Finally, LanDscAPe examines the TLS configurations of LDAP servers. Note that the scanning and analysis methods behind LanDscAPe, in particular those that take and analyze directory samples, are built to minimize harm to affected users. For example, we only take small directory samples, even if the server is willing to return larger samples or even the full directory, and delete all samples after analysis.

LanDscAPe found 3.7 million IPv4 addresses responding on LDAP ports 389 or 636; about 82.1k IPs sent valid LDAP responses. 12,179 IPs (14.83%) respond with personal data, and out of those, only 1,392 (11.4%) use a recommended

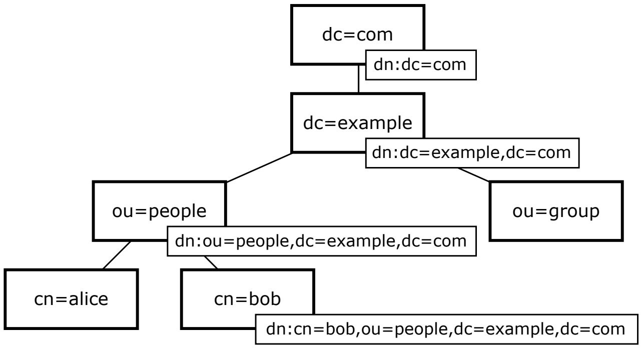

Figure 1: LDAP DIT example.

cipher suite and have a valid certificate chain. This means that the vast majority of LDAP servers exposing personal data insufficiently protect the confidentiality and authenticity of personal data. Furthermore, 616 IPs (0.75%) run an LDAP server version that is linked to at least one CVE and 9,731 IPs (11.85%) leak possibly sensitive internal information. In combination, these findings mean that it is trivial for attackers to obtain valuable information about organizations. Personal data in LDAP directories often resembles contact information of employees; these may even leak the organization's structure, which is valuable reconnaissance information.

Even worse, our sampling revealed that 1,817 IPs (2.21%) leak passwords, either hashed or plaintext. To further minimize harm to users, we develop a method for counting the total numberount of passwords within a directory without actually downloading them. In total, the affected servers leak 3.9 million passwords. For technical, ethical, and legal reasons, we cannot validate whether these passwords are (still) valid, and for many LDAP servers, we do not know which organizations and services the directories are part of. Still, we fear that the sheer amount of leaked passwords implies that a sizable number is used in real-world services.

Contributions. In summary, our contributions are:

- We introduce LanDscAPe, enabling semi-automated and Internet-wide security analysis of public LDAP servers.
- We analyze technical opportunities and limits of the LDAP protocol as well as ethical considerations for Internet-wide LDAP security analysis.
- We uncover thousands of LDAP servers that host personal data and use a subpar security configuration, including TLS ciphers and certificates. We find 1.8k LDAP servers that leak hashed or plaintext passwords, posing a high risk for affected organizations and users.
- We describe our coordinated disclosure campaign and its effectiveness.

2 Background

2.1 The Lightweight Directory Access Protocol (LDAP)

Data access using the Lightweight Directory Access Protocol (LDAP) assumes an X.500 directory structure, which uses a hierarchical tree structure called Directory Information Tree (DIT), as shown in Fig. 1. Every entry within this directory system is identified by a distinct Distinguished Name (DN). This DN consists of the entry's name and the path leading back to the root entry—also known as the directory path. Additionally, entries have attributes with a predefined type and a value. The type determines the syntax, semantics, and characteristics of the value. Attribute types are uniquely identified internationally by their Object Identifier (OID) [25, 26].

The Directory System Agent (DSA)'s DSA-Specific Entry (DSE) is the root node within the DIT [57]. Clients can access this entry through a search operation specifying the desired attributes of the DSE. These attributes hold a unique significance, as they encompass essential information about the server, such as supported controls, extensions, features, LDAP versions, SASL mechanisms, and more [57].

2.2 LDAP Authentication and Authorization

Clients authenticate to an LDAP server by attempting a bind operation. A connection between the client and the server is established if the bind is successful [16, 48]. As part of the bind request, the client chooses an authentication method and, if necessary, supplies authentication credentials. LDAPv3 supports four types of authentication: simple, anonymous, unauthenticated, and SASL authentication [16].

Simple Authentication is performed when the client transmits the fully qualified DN of a user and a cleartext password. This method necessitates an encrypted channel, such as TLS, as transmitting the password in plaintext exposes it to the network [16].

Anonymous Authentication uses empty values for both the username and password or sends a request without a prior bind operation. The client is then authenticated as an anonymous user. This anonymous bind is a mandatory part of the protocol [16].

Unauthenticated Authentication is a simple authentication with a username and an empty password. The client is then bound as an unauthenticated user. It's worth noting that RFC 4513 [16] recommends disabling this method by default for clients and servers. This is due to the unexpected nature of users being able to bind without supplying a password, which may cause security issues.

SASL. SASL supports a wide range of authentication methods, such as username/password, Kerberos, and digital certificates. Anonymous and plaintext SASL authentication is not commonly used in LDAP [17]. While some are notably robust, others have been identified as insecure [58].

Authorization, or access control, is not documented within the LDAP Standards [48]. Consequently, LDAP server vendors often autonomously develop their own access control mechanisms. OpenLDAP uses Access Control Lists (ACLs) [33, 37] for this purpose [15].

LDAP as an Authentication Provider. LDAP is frequently used as a central component of single sign-on systems. For this purpose, most implementations follow a specific workflow: Initially, a search operation is initiated for the provided username to retrieve the DN of the user's LDAP entry. For this, LDAP servers must either allow unauthenticated searches or the application must use an authorized service account. Without the initial search, user sign-in becomes cumbersome because the user needs their Distinguished Name (DN), such as `CN=John Doe, OU=Users, DC=example, DC=com`. After the search, the system proceeds to bind using the DN and the provided password.

2.3 Data Retrieval from LDAP Servers

The search operation is a fundamental feature of LDAP, enabling the retrieval of entries from an LDAP server. The search is performed relative to a base object entry, which the client specifies using a DN. This can also be the root entry of the LDAP directory. Additionally, the scope of the search can be restricted. This allows limiting the search to the specified entry, to the immediate subordinate level of the specified entry, or to its entire subtree.

The search criteria are determined by specifying search filters, based on which the search is conducted. For example, the filter `sn=Doe` searches for an entry where the attribute `sn` (surname) contains the value Doe. Multiple filters can be combined using logical operators such as AND, OR, or NOT. Additionally, searching for any value in an attribute is permissible by specifying an asterisk (*e.g.,* `sn=*`). Furthermore, the LDAP protocol allows for the specification of attributes in the search request that should be exclusively returned in the event of a match. It is possible to determine whether only the attribute names or both the attribute names and values should be returned.

Another option that can be specified in the search request is the size limit. This allows the client to determine the maximum number of results to be returned by the server in the event of a match. In addition to client-side size limits, various LDAP servers allow for server-side size limits to be configured to restrict the number of entries returned in a search request. Different LDAP server manufacturers establish different default values for size limits. For example, the default size in OpenLDAP is set at 500 entries, while Apple's Open Directory Server sets it at 11,000 entries.

3 Methodology for Analyzing LDAP Security

3.1 Threat Model

We assume that the attackers access publicly available LDAP servers on common ports, *e.g.,* by connecting to the IP address of a public LDAP server with a standard LDAP browser. The attackers send standard-conforming packets to retrieve data from LDAP servers and do not attempt to circumvent deployed access controls. Further, we assume they take educated guesses about default configurations—including administrator usernames and attribute names commonly observed.

When analyzing the security of TLS configurations, we assume that network attackers act as active Meddler-in-the-Middle (MitM) between the LDAP client and server. These attackers aim to exploit weak TLS configurations or downgrade the connection to plaintext to steal the LDAP client's credentials and other sensitive data.

3.2 LanDscAPe Analysis Pipeline Overview

LanDscAPe uses a set of modules for the analysis illustrated in Fig. 2. This section gives a brief overview of the analysis pipeline. The *scanning module* performs Internet-wide port scans, resulting in a dataset of IPs responding on well-known LDAP ports. It also collects information on TLS configurations on LDAP servers. Upon detecting an open LDAP port on either of the targeted ports, LanDscAPe's *confirmation module* sends LDAPv2, LDAPv3, or unspecified LDAPv4 messages to provoke an LDAP response. The *sampling module* then attempts to connect to identified LDAP servers and performs information requests through standard-conforming connection and anonymous authentication methods. If a connection succeeds, the *sampling module* requests configuration settings and metadata. Additionally, it requests a subset of LDAP entries from the server for further analysis and stores the data on an internal analysis server.

The two final stages analyze the collected data. The *data classification module* analyzes the configuration details and collected samples for exposed personal data, sensitive data shortcomings, and leakage of credentials. The last stage is the *network analysis module*, which analyzes the security of LDAP servers' TLS configurations. The system examines the supported cipher suites and the trust chain. The pipeline was executed in January 2024, and we base our evaluation on the data gathered from this run.

3.3 Scanning Module

Our *scanning module* begins by scanning the entire routable IPv4 space on ports TCP/389 and TCP/636 with ZMap [9]. The former port is used for plain LDAP and for the in-band upgrade with StartTLS, the latter port for LDAP over TLS. This gives us a list of IPv4 addresses where these ports are

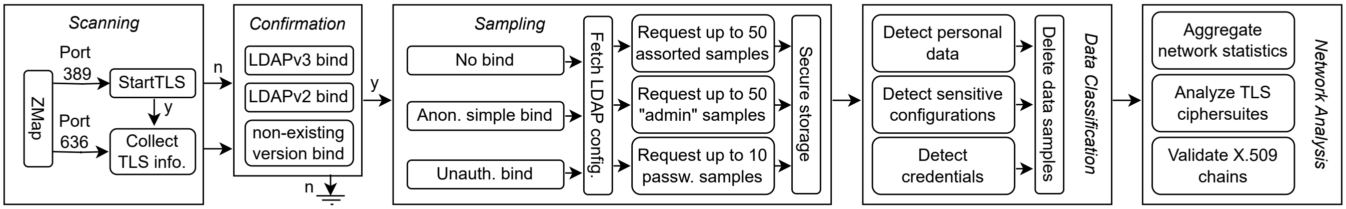

Figure 2: LanDscAPe analysis pipeline.

open. Previous work [27, 44] shows that many middleboxes will make a port seem open, meaning that the true number of LDAP servers can only be determined by following through with a higher-layer connection, *i.e.,* plain LDAP, LDAP and an in-band upgrade to TLS via StartTLS, or a TLS connection over which an LDAP connection runs. These are implemented as submodules that execute within 24 hours of the scan. This means that IP churn can lead to us missing an LDAP host that was still visible in the ZMap scan. We tested the churn and found that it is at most between 1-5% per day; *e.g.,* our TLS scans always cover at least 95% of all IP and port combinations (IP:port) for which we also ran successful LDAP queries. To enable StartTLS for LDAP on port 389, we extend Goscanner [11] with this form of in-band upgrade to collect connection properties and certificates. On port 636, we connect directly with Goscanner and do a TLS handshake. Goscanner is configured to offer TLS versions from 1.3 down to 1.0, in that order. A server that supports a more modern version of TLS should hence select it from our scanner's offer. Similarly, we configured the cipher suites we offer according to the best current practices [50], ordered from highly secure to less secure. However, we do not attempt multiple connections to determine all cipher suites that LDAP servers support to avoid undue load on them. Note that some servers support both TLS-protected and plain LDAP sessions. In our analysis, we treat this case as two distinct ways of accessing a server.

3.4 Confirmation Module

The *confirmation module* ties directly into the scanning module, but we describe it separately here. It verifies that the candidate IP:port combinations collected in the previous steps actually serve the LDAP protocol. To determine this, we send three anonymous simple bind requests with varying versions to each server. First, a normal LDAPv3 bind to check for general LDAP support. Second, an LDAPv2 bind, testing for backward compatibility and outdated servers. Finally, a bind with a non-existing version (v4) to provoke a “BindResponse where the resultCode is set to protocolError” [48].

If a server does not return a valid LDAP response within five seconds, we abort the attempt. Conversely, if the server replies with an RFC 4513-conforming LDAP message [17], we classify the service as an LDAP server. If no protocol error is encountered, we assume that the server supports the respective LDAP version.

After this step, for each IP:port tuple, we know if they are standard LDAP servers, which LDAP versions and connection security they support, and which TLS certificates are used.

3.5 Sampling Module

The *sampling module* gathers information in a three-step process: connecting to an LDAP server, fetching configuration data and metadata, and collecting sample entries.

Connecting to Servers. For each server with each supported connection type (port 389 with and without StartTLS, and port 636 with implicit TLS), we initially request information without bind, leading to an anonymous authorization identity, according to Section 4 of [17]. Subsequently, it executes a simple bind request with both username and password fields left empty, aligning with the specifications in Section 5.1.1 of [17]. The third approach involves an unauthenticated bind, as outlined in Section 5.1.2 of [17], using an arbitrary username (we chose `cn=*`) and an empty password field. We do this for all available connection types per IP address as it may serve different LDAP services or configurations on each.

Fetching Configurations. Next, we collect configuration information and metadata from the server. We first request the root DSE that contains information on supported LDAP controls, extensions, features, and versions. Additionally, it contains the server's naming contexts. Based on this, we request the server's schema, size limit, and password policies.

Collecting Samples. To assess the data the server publicly exposes, we collect a specific number of entries from the server. To retrieve entries from an LDAP server, the client sends an LDAP `SearchRequest` and gets the result via the server's `SearchResultEntry` response.

For our analysis, we request random, admin, and password samples. We request random samples by sending a query filtering for `objectClass=*` and admin samples by filtering for `cn=*admin*`. To prevent the server from sending excessive data, we limit each search to a maximum of 50 entries. For password samples, we request the following fields: `userpassword`, `password`, `sambantpassword`, sambalmpassword, krbprincipalkey, clearpassword, goimappassword, lmpassword, and ntpassword. Here, we limit each request to a maximum of ten results. For ethical reasons, we do not make further requests and accept a smaller sample size if the server responds with less.

Secure Data Storage. Even though we only sample public LDAP servers, collected data samples may contain sensitive information as critical as user credentials. To protect these samples, this module moves them from the scanning server to an internal server that is only accessible via VPN by the researchers involved in this analysis.

3.6 Data Classification Module

The *data classification module* tries to identify the types of data samples collected in earlier stages. Particularly, it aims to detect personal data, internal information, and credentials within our samples. In its final step, all samples are deleted, and only metadata indicating the type of data found per server is kept to prevent possible harm from unauthorized access.

Detect Personal Data. We analyze our samples for typical personal data attributes such as names, email addresses, or certificates. We follow Art. 4 of the GDPR here that defines personal data as any information relating to an identified or identifiable natural person [12, Art. 4]. To identify personal data, we first aggregate semantically identical attributes of our samples (max. 50 samples per server, less if the server sends less), *e.g.,* attributes such as surname, sn and lastname (see Table 12 for more details). We then group by this aggregation and count the servers containing personal data.

Detect Sensitive Configurations. We try to identify potentially insecure servers and confidential information on servers. Therefore, we first categorize LDAP servers into usage scenarios and common products. This categorization uses three indicators: OIDs, the root DSE vendorName and vendorVersion fields, and random sample attributes with hints such as the msExch prefix referencing an AD. Since some servers communicate their exact version, we look up the corresponding version number on [46]. However, this is only a weak indicator that the vulnerability is exploitable and harms the overall security.

Furthermore, we look at SASL (Simple Authentication and Security Layer) mechanisms supported by servers. While some mechanisms are notably secure, *e.g.,* using salts or providing memory or CPU hardness, others are not. If LDAP servers act as authentication servers, a strong Simple Authentication and Security Layer configuration is important.

Moreover, we consider servers exposing internal information related to LinuxUserManagement, Microsoft Active Directory, DICOM, and passwordPolicies having a sensitive configuration. If this information is publicly accessible (*e.g.,* gidNumber, uidNumber), it might be utilized by attackers.

```
def generic_query(prefix):
    results = []
    for letter in alphabet:
        filter=f"(&(cn={prefix + letter}*)(
            ↪ userPassword=*))"
        try:
            result=ldap_connection.search(filter,
                ↪ attributes=[])
        except SIZELIMIT_EXCEEDED:
            result=generic_query(prefix + letter)
        results.extend(result)
    return results
```

Listing 1: Prefix search to count LDAP entries.

Credentials Detection. To detect user credentials on LDAP servers, we examine our fetched password samples (max. 10 samples per password field, less if the server sends less, see section 3.5) and evaluate their values. We automate this process using tags according to [15] and pattern recognition (*e.g.,* BCRYPT). If we detect encodings such as base64, we try automatically to decode and restart the evaluation process. If the password type of a value was not recognizable (*e.g.,* password was masked with asterisks), we excluded the server from further analysis. To assess the extent of leaked passwords, we proceed by reconnecting to the server and counting the number of entries that contain a non-empty password field without transferring any further sensitive information by restricting the attributes returned by the server to the DN. We do this by setting the requested attribute list to empty as shown in Listing 1. We have taken these steps to transfer minimal sensitive information minimizing the risk for users.

Data Deletion. The module only collects and retains information about the type of data available on any public LDAP server. To minimize risk for organizations and users, we delete all collected samples when this module finishes.

3.7 Network Analysis Module

This module analyzes TLS cipher suites and the X.509 certificate chains in the TLS handshakes; it also provides access to aggregate statistics enriched with third-party data sources, such as IP2Location[1] to determine the geographical location and hosting situation of an LDAP instance. For the purpose of this paper, we present only analyses for LDAP servers with security-relevant properties, *e.g.,* those that support authentication via the SASL framework or host personal data or user credentials. Such servers should use secure TLS configurations: even if the data can be queried anonymously, it may cross other networks and should hence be protected against privacy-invasive monitoring (*e.g.,* queries to a public address

[1] https://www.ip2location.com

Region	Total IPs	Per 1M inhabitants	Leaking Credentials	Personal Data	Sensitive LDAP Configuration		
					SASL Support	CVE Exists	Internal Info.
World	82,129 (100%)	46	1,817 (2.21%)	12,179 (14.83%)	26,464 (32.22%)	616 (0.75%)	9,731 (11.85%)
United States	18,466 (22.48 %)	56	463 (2.51 %)	2,968 (16.07 %)	4,995 (27.05 %)	204 (1.10 %)	1,744 (9.44 %)
Germany	8,123 (9.89 %)	97	77 (0.95 %)	872 (10.73 %)	2,138 (26.32 %)	48 (0.59 %)	671 (8.26 %)
France	5,122 (6.24 %)	78	95 (1.85 %)	888 (17.34 %)	1,833 (35.79 %)	25 (0.49 %)	632 (12.34 %)
Poland	5,040 (6.14 %)	133	27 (0.54 %)	153 (3.04 %)	708 (14.05 %)	8 (0.16 %)	144 (2.86 %)
Russian Federation	4,042 (4.92 %)	28	36 (0.89 %)	406 (10.04 %)	1,112 (27.51 %)	39 (0.96 %)	389 (9.62 %)
China	4,004 (4.88 %)	3	369 (9.22 %)	812 (20.28 %)	1,037 (25.90 %)	33 (0.82 %)	729 (18.21 %)
United Kingdom	2,724 (3.32 %)	40	12 (0.44 %)	230 (8.44 %)	452 (16.59 %)	12 (0.44 %)	139 (5.10 %)
Taiwan	2,717 (3.31 %)	113	24 (0.88 %)	383 (14.10 %)	1,027 (37.80 %)	3 (0.11 %)	203 (7.47 %)
Brazil	2,656 (3.23 %)	12	57 (2.15 %)	232 (8.73 %)	583 (21.95 %)	4 (0.15 %)	269 (10.13 %)
India	2,583 (3.15 %)	2	89 (3.45 %)	356 (13.78 %)	875 (33.88 %)	5 (0.19 %)	388 (15.02 %)
Italy	2,545 (3.10 %)	42	46 (1.81 %)	411 (16.15 %)	672 (26.40 %)	19 (0.75 %)	281 (11.04 %)
Netherlands	2,260 (2.75 %)	132	18 (0.80 %)	187 (8.27 %)	498 (22.04 %)	16 (0.71 %)	162 (7.17 %)
Canada	2,027 (2.47 %)	54	14 (0.69 %)	384 (18.94 %)	769 (37.94 %)	13 (0.64 %)	291 (14.36 %)
Switzerland	1,548 (1.88 %)	179	9 (0.58 %)	116 (7.49 %)	202 (13.05 %)	11 (0.71 %)	74 (4.78 %)
Indonesia	1,479 (1.80 %)	5	17 (1.15 %)	127 (8.59 %)	1,023 (69.17 %)	4 (0.27 %)	482 (32.59 %)
Viet Nam	1,462 (1.78 %)	15	6 (0.41 %)	81 (5.54 %)	545 (37.28 %)	1 (0.07 %)	308 (21.07 %)
Japan	1,338 (1.63 %)	11	93 (6.95 %)	289 (21.60 %)	318 (23.77 %)	27 (2.02 %)	184 (13.75 %)
South Africa	1,200 (1.46 %)	20	1 (0.08 %)	54 (4.50 %)	116 (9.67 %)	5 (0.42 %)	28 (2.33 %)
South Korea	1,179 (1.44 %)	23	43 (3.65 %)	201 (17.05 %)	319 (27.06 %)	5 (0.42 %)	195 (16.54 %)
Australia	1,032 (1.26 %)	40	15 (1.45 %)	104 (10.08 %)	203 (19.67 %)	6 (0.58 %)	85 (8.24 %)
Thailand	1,019 (1.24 %)	15	16 (1.57 %)	73 (7.16 %)	326 (31.99 %)	6 (0.59 %)	183 (17.96 %)
Hong Kong	918 (1.12 %)	122	28 (3.05 %)	109 (11.87 %)	281 (30.61 %)	9 (0.98 %)	161 (17.54 %)
Czech Republic	909 (1.11 %)	85	21 (2.31 %)	149 (16.39 %)	308 (33.88 %)	21 (2.31 %)	114 (12.54 %)
Singapore	837 (1.02 %)	143	22 (2.63 %)	134 (16.01 %)	384 (45.88 %)	7 (0.84 %)	135 (16.13 %)
Spain	821 (1.00 %)	18	9 (1.10 %)	127 (15.47 %)	302 (36.78 %)	3 (0.37 %)	90 (10.96 %)

Table 1: Overview of the results of the Data Analysis Modules grouped by region, sorted by total, top 25 regions. Percentages are based on total servers in the region.

book will reveal the communication behavior of individuals). We define "secure" by referring to the best-practice recommendations for TLS in RFC 9325 [50] for connections and X.509 certificate chains. The server chooses the cipher suite from a list the client offers; we can obtain the chosen cipher suite from the handshake.

The certificates need to be validated against root stores that contain the public keys of recognized certificate issuers, *i.e.,* Certificate Authorities (CAs). For the Web PKI, such certificate stores are well-known and can be downloaded for browsers and operating systems. For LDAP, no defined set of CAs exists. Hence, we check against a "meta root-store" and validate the LDAP certificates against the combined root stores from Apple, Microsoft, Google, CCADB (Mozilla), and Oracle (Java). Together, these are known to cover the majority of TLS user agents for web users [31]. We define a valid chain as one that is valid for at least one root store.

Various problems with certificate chains have been found, initially for the Web [8, 22] and later for other communication protocols, especially email [21]. The use case of LDAP is different from these protocols: both the Web and email servers are, almost by definition, meant for public access. LDAP is commonly used in ways that are internal to an organization. Hence, there are two choices to configure X.509: either by using a certificate from a globally acting CA or by running a custom, "in-house" CA. The former method has the advantage that it is thoroughly documented and certificates are available at no cost from CAs like Let's Encrypt. The latter method also avoids costs, but it comes with more administrative overhead as the local CA must be operated securely by the organization (tooling is available in some environments, *e.g.,* Microsoft Windows servers) and its root certificate deployed to client software. In theory, self-signed certificates could also be used to replace an in-house CA; in this case, one would need to ship the self-signed certificate with the client (and possibly alter the software to accept such certificates).

We call certificate chains *globally invalid* if the root CA is not in the root store. This means that our estimate for the number of valid certificates is a *lower bound* (and an upper bound for the number of invalid chains). Custom CAs can be expected to issue relatively few certificates (as there are few servers per organization). Independently of these considerations, we always call a chain invalid if there is a cryptographic problem, *e.g.,* use of the deprecated SHA1 algorithm for certificate signatures. Our validator uses the standard Go library, which rejects such certificates as invalid.

Note that we cannot validate the names in the certificates, as the domain names that resolve to an IP are unknown.[2]

[2] While some large DNS datasets exist (*e.g.,* OpenINTEL), these do not cover subdomains, which are likely names for LDAP servers.

Version Support	Port 389		Port 636		Any	
Any	80,968	(100.00%)	34,095	(100.00%)	82,129	(100.00%)
v2	48,362	(59.73%)	21,162	(62.07%)	56,798	(69.16%)
Only v2	177	(0.22%)	32	(0.09%)	205	(0.25%)
v3	80,750	(99.73%)	34,043	(99.85%)	81,322	(99.02%)
Only v3	32,565	(40.22%)	12,913	(37.87%)	35,325	(43.01%)
None	41	(0.05%)	20	(0.06%)	55	(0.07%)

Table 2: Identified LDAP servers with supported versions.

4 Results

Using the pipeline described above, we aim to answer two main questions: What sensitive data do publicly accessible LDAP servers expose, and how protected are the connections a client can establish to them?

The structure of this section follows the same analysis pipeline as Section 3 by presenting the results of each module in Fig. 2. An overview of the results is given in Table 1.

4.1 Scanning

Our ZMap scan identified 3,704,816 hosts responding to the SYN scan on port 389 and 3,763,310 hosts on 636. We collected 67,498 distinct TLS certificates from 50,970 hosts on port 636 and 35,707 hosts on 389 via StartTLS, including non-static hosts. In pilot studies, we observed that LDAP servers usually run on static IP addresses, observing a churn rate as it is commonly observed for other protocols as well [35].

4.2 Confirmation

We determined that 80,968 servers on port 389 and 34,095 servers on port 636 (82,129 distinct IP addresses in total) respond to our LDAPv2 bind or LDAPv3 bind with an LDAP result message as defined in Section 4.1.9 in [48]; we classified them as LDAP servers (Table 2). Nearly all LDAP servers support LDAPv3, with slightly over half accepting LDAPv2 binds; however, this does not necessarily indicate that the server actually supports LDAPv2. For instance, OpenLDAP documentation [15] states that OpenLDAP does not support LDAPv2 but accepts LDAPv2 bind requests for legacy reasons. Nevertheless, we observed 177 servers on port 389 and 32 on port 636 that only accept LDAPv2.

During the confirmation process, we observed that a few LDAP servers delivered a `protocolError` either during an LDAPv2 or LDAPv3 bind. However, using the respective other version, our connection experiences a timeout (two seconds) or receives a connection reset. The occurrence of a `protocolError` indicates that the server is an LDAP server; however, it does not support the requested protocol version.

4.3 Sampling

We established a total of 115,063 connections (`IP:port`), as shown in Table 2. We found that many connections were willing to send data without a bind request. For example, we retrieved the root DSE entry 92,588 times: 95.04% without a bind, 4.93% with a simple bind, and 0.03% with an unauthenticated bind. This means that in some cases, we were able to request additional information from the server via simple bind and unauthenticated bind Table 3. However, according to RFC 4513 [17], the server should be in the same state as without binding. This hints at misconfigurations and faulty assumptions regarding the authorization and authentication mechanisms LDAP provides.

Alarmingly, we identified entries[3] with credential attributes on 2,935 connections (`IP:port`), highlighting a significant security concern.

5 Data Analysis

This section describes the results of the data analysis module described in Section 3.6.

5.1 Vendor and Type of LDAP Server

In total, we categorized 10,740 servers using supported controls, vendor information, and attribute combinations.

Using the supported controls (OIDs) during our scans, we were able to categorize 5,938 hosts (of 35,083 hosts that expose them)—predominantly Active Directory servers, Sun Java, and iPlanet directory servers (`OID` in Table 11).

Only a small set of hosts (2,382) expose the `vendorName` and `vendorVersion` attributes, which allows us to identify 23 different LDAP server types (`Version` in Table 11).

We attempt to map all of the 3,668 unique attributes observed in the samples to a server type. As a result, of the 16,218 hosts for which we have samples, 5,577 hosts could be categorized (`Attributes` in Table 11). Predominantly, this allowed identifying servers used for Linux user management and servers with Active Directory support for Apple.

Merging all the described categorization approaches results in the LDAP server types shown in Table 11. Note that servers can be categorized into multiple categories—*e.g.,* a server that uses a specific server software and is used for Linux user management.

5.2 Detected Sensitive Configurations

We found several sensitive server configurations that potentially are vulnerable to security flaws or expose sensitive data.

[3] Note that not each entry contains actual user credentials (cf. Section 5).

Bind Type	Root	Schema	Naming	Passw.	Samples		
	DSE		Context	Policy	Rand.	Admin	Passw.
Total	92,588	35,658	90,265	1,432	17,283	11,715	2,935
No bind	95.04%	87.28%	94.97%	99.93%	84.02%	99.74%	99.18%
Only Simple bind	4.93%	12.69%	5.00%	0.07%	15.98%	0.26%	0.82%
Only Unauth. bind	0.03%	0.03%	0.03%	0.00%	0.00%	0.00%	0.00%

Table 3: Summary of sampling result per IP:port combination.

SASL Mechanism	# IPs	Plaintext only	TLS config	
			w/ issues	w/o issues
Total	26,464	7,962	13,418	5,084
Hash	87.95%	31.67%	47.53%	20.80%
CR	87.68%	32.57%	47.57%	19.86%
GSSAPI	57.52%	14.63%	58.13%	27.24%
Plain	32.63%	3.56%	80.68%	15.76%
External	6.05%	10.74%	68.66%	20.60%
Anonymous	4.61%	8.52%	66.99%	24.49%
Proprietary	0.37%	29.59%	68.37%	2.04%

Table 4: Supported SASL Mechanism Types. Percentages are based on Mechanism Type. Servers commonly support multiple mechanisms.

Supported SASL Mechanism. Table 4 lists the SASL mechanisms supported by servers. In total, 26,464 servers support SASL authentication; however, less than one-fifth of them allow authentication over a secure connection. This is especially glaring for the servers supporting plain authentication, of which more than four-fifths do have issues within the TLS configuration—potentially leaking plaintext credentials to a MitM attacker. The categorization of issues with TLS configurations is explained in Section 6.

We also find it worrying that a large share of servers support challenge-response and hash-based authentication only allow plaintext connections (32.57% and 31.67% respectively). This puts the sole trust in the confidentiality provided by SASL.

CVE Check. We found 616 servers with known vulnerabilities (identified by CVEs). Eight servers use a version of the `UnboundID LDAP SDK for Java` that might be vulnerable to CVE-2018-1000134—an authentication bypass using unauthenticated bind with a CVSS of 9.8. Additionally, we detected two known vulnerabilities in servers identified as `389-Directory Servers`: 608 servers might be vulnerable to CVE-2020-35518—an information disclosure on the directory structure with a CVSS of 5.3—and 413 might be vulnerable to CVE-2018-10935—a denial of service with a CVSS of 6.5. The rather low number of servers with known vulnerabilities does not necessarily indicate that these servers are well maintained. On the contrary, we assume that we could not identify a large share of vulnerable servers due to missing product and version information.

Internal Information. Based on the categorization explained in Section 3.6, we could identify hosts exposing settings or parameters that, if accessed by unauthorized users, can lead to security breaches or data leaks.

As shown in Table 11, we found 5,180 hosts that are used for Linux user management and 4,465 hosts that can be identified as Microsoft Active Directory. Neither of these should be exposed to the Internet.

Additionally, there are 283 hosts that expose password policies. These, for example, contain information about the minimum length of a password and thus provide conclusions about the strength of the passwords, which in turn helps in the creation of dictionary attacks. A notable risk involves the potential for targeted account lockouts [49]. Malicious actors can exploit this by issuing bind requests with incorrect passwords, thereby intentionally locking out individual or multiple accounts. This becomes an issue if user enumeration is possible and lockout policies are in place.

Further sensitive information in the form of hosts with Digital Imaging and Communications in Medicine (DICOM) configurations could be found. The DICOM standard for the storage and transport of medical radiologic images proposes the usage of LDAP for configuration data [36, Part 15, H.1]. DICOM's standardized secure authentication methods are often not supported in commercial products and rarely used in productive systems [10]. One common way of securing access is the usage of the Application Entity Title (AET) as a password in a way that the client must provide the correct AET before the connection is allowed by the server. The DICOM configuration data in the LDAP scheme contains, besides the medical institution name, device name, and software version of the DICOM device, the AET title. We found 145 publicly accessible LDAP servers with DICOM configuration values. The publication of the AET title on a public LDAP server is, to the best of our knowledge, not meaningful, and if the AET is used for authentication, a severe security vulnerability.

Personal Data	# IPs	Plaintext only	TLS config	
			w/ issues	w/o issues
Total	12,179	4,801	5,986	1,392
Last Name	89.86%	39.19%	49.01%	11.80%
Email	65.34%	36.55%	50.09%	13.36%
First Name	64.41%	37.92%	51.84%	10.24%
Full Name	49.50%	27.41%	60.24%	12.36%
Phone No.	26.83%	37.82%	47.92%	14.26%
Public Key	14.90%	23.25%	62.04%	14.71%
Location	10.54%	50.55%	32.01%	17.45%
Job	10.17%	33.28%	45.96%	20.76%
Address	10.14%	44.21%	33.36%	22.43%
Title	9.93%	39.04%	35.65%	25.31%
Country	7.45%	58.77%	29.33%	11.91%
Photo	2.82%	47.23%	35.28%	17.49%
Birthday	1.45%	21.02%	47.73%	31.25%
Gender	0.62%	54.67%	38.67%	6.67%
SSN	0.16%	21.05%	78.95%	0.00%

Table 5: Personal data attributes exposed by LDAP servers. Percentages based on the corresponding number of IPs.

5.3 Detected Personal Data

LDAP servers are commonly used to manage users. These servers provide user authentication or serve as public or internal address books. Therefore, it is plausible that these servers contain personal data such as names or email addresses. The question arises as to how many servers provide us with personal data and what form of data is accessible. The *Sampling Module* collected a limited sample set (see Section 3.5) from $16,218$ LDAP servers, which serves as the basis for the investigation in this section.

The samples can be used to determine which server makes which type of personal data publicly available. We divided the attributes we observed in our sample set that potentially hold personal data into groups. We have listed these groups together with the number of servers that provide us with the corresponding personal data in at least one sample in Table 5. A complete mapping of all LDAP attributes to a group can be found in Table 12 in Appendix C.

We identified a total of $12,179$ servers publishing one piece of personal information. We also categorized the number of servers from which the corresponding information can only be transmitted in plain text on port 389. If a server offers TLS, we have listed how many servers have issues with TLS configuration for the transmission. The classification of issues in TLS configurations is explained in Section 6.

In addition to the names and email addresses, LDAP servers also hold public keys. This category includes S/MIME certificates, which usually contain the certificate owner's name. Since LDAP servers are often used within a PKI to distribute S/MIME certificates, the publication of this information is common. More concerning, we found that some LDAP servers return entries with a date of birth, a photo, or even a social security number. Although management of such information in an LDAP directory is plausible, the publication of such information may indicate a misconfigured LDAP server.

In particular, servers that publish attributes such as name, address, city, country, email, or telephone number can only transmit the data in plain text in at least a third of cases. Even the servers that can transmit this information via TLS use TLS configurations with issues in more than a third of cases.

	Passw.	Passw. & Hash
Total #IPs	32	28
Plausible Credentials	18	15
Numerical Credentials	2	-
Sample size 1	6	11
Implausible Credentials	6	2

Table 6: Manual analysis results of servers exposing over 1,000 credentials.

5.4 Detected Credentials

As LDAP can serve authentication services, it is a common practice to store credentials on these servers. In this chapter, we analyze the security implications that arise from misconfigurations by admins, posing a risk to stored credentials. Our analysis reveals that such misconfigurations lead to exposed credentials on $1,817$ (2.21%) of the LDAP servers we examined. RFC 4519 [45] recommends storing passwords in the `userPassword` field as plaintext strings without any form of encryption. We speculate that this recommendation may have contributed to a substantial proportion of servers (0.71%) exposing plaintext passwords.

Verification of Credential Leaks. In our investigation of servers that may leak credentials to unauthorized entities, we faced the challenge of verifying the authenticity of these credentials. A first analysis of the data samples indicated that the exposure of a potentially sensitive attribute does not mean the information is actually sensitive. As a result, we eliminated a proportion of servers (26.7%) where passwords were displayed in a redacted form (*e.g.,* partially masked by asterisks). To reduce potential harm to users as much as possible, we limit credential sampling to 10 entries per password attribute and server, even if a server is willing to return more. This intentionally small sample size made further automated evaluation challenging.

An analysis of the credential distribution indicates that the vast majority of credentials were stored on relatively few servers. This prompted a *manual* examination of our samples of all servers harboring over $1,000$ credentials. Table 6 shows that the majority of servers (18) expose only plausible plain-

Field	**# IPs**	**# of Leaks**
Total	1.8k	3.9m
userPassword	1.4k (78.8%)	3.2m (80.5%)
sambaNTPassword	343 (18.9%)	187k (4.8%)
sambaLMPassword	268 (14.8%)	157k (4.0%)

Table 7: Top three fields (potentially) leaking credentials.

text credentials within our samples, while 15 servers expose plausible plaintext credentials mixed with hash values. We suspect that this is due to the migration of some servers, for example, from plaintext to salted hashes. We also classified some servers as implausible when the sample consisted of very few, repeating identical hash or password values.

The Scope of Credential Exposure. To reduce potential harm for users, we only requested samples of user credentials from LDAP servers with a hard limit of ten entries. That way, we can detect *if* an LDAP server exposes credentials, but not *how many*. However, to determine the risk a credential-leaking server poses, we need to estimate the number of credentials on the server. We accomplished this by performing an enumeration of entries (as outlined in Section 3.5) containing at least one of the identified sensitive attributes (*e.g.,* `userPassword=*`), but only requesting the `DN` of these entries, ensuring minimal data exposure. The returned `DN`s are then counted and discarded.

The outcomes of this enumeration are presented in Table 7. We found that more than 3.9 million credentials were publicly accessible at the time of our data collection in January 2024. Moreover, very few servers expose the majority of credentials.

For ethical reasons, we chose not to download the entire dataset of passwords, and we therefore checked the plausibility of the credential samples. Table 6 summarizes our efforts to check the plausibility and distribution of plaintext and hash values of these passwords.

6 Network Analysis

We analyze data on the network configuration of LDAP instances and the Transport Layer Security (TLS) configuration of LDAP servers. We say an LDAP server's TLS configuration is "without (potential) issues" when it fulfills two conditions: it has a valid X.509 chain, signed by a known root certificate of a Certificate Authority (CA) from a root store, and it follows the respective recommendations from RFC 9325 [50] guidelines for connection security.

Hosting. We use the IP2Location dataset to determine in which countries and networks LDAP instances are deployed and summarize this in Table 1. More than 50% of LDAP servers are hosted in the US, Germany, France, Poland, Russia, or China. As shown in Table 8, nearly half of the instances are located in networks that IP2location classifies as data centers, *i.e.,* a form of hosting. Interestingly, 39.02% are located in network ranges classified as Internet Service Providers (ISPs). This classification *may* indicate self-hosting situations. We note, however, that these categories are rather broad, with blurry boundaries as ISPs often have multiple network ranges for various purposes, which also change over time. Hence, we take this classification with a grain of salt. We also find a sizable number of LDAP servers in commercial settings (7.32%) or education (4.16%). Outside the top four, we also find some deployment in government networks (0.40%) or military networks (0.01%). In all categories, we find servers associated with leaked passwords and personal data—including some in government or military networks.

Network usage type	**Total IPs**	**Leaks**	
		Credentials	**Personal data**
Total	92,109 (100%)	1,817 (100%)	12,179 (100%)
Data Center	48.17%	49.04%	39.63%
ISP	39.02%	26.97%	41.18%
Commercial	7.32%	14.91%	9.24%
Educational	4.16%	6.60%	8.19%

Table 8: Network types where LDAP servers are located.

TLS Cipher Suites. Our Network Analysis Module aggregates all IPs that successfully negotiated a TLS connection with the TLS server certificate, forming a TLS-enabled subset of 48.2k servers (58% of all identified LDAP servers).

TLS defines several cipher suites, indicating the symmetric cipher, encryption mode, hash algorithm, and the key exchange method. We evaluate the established TLS connections against the best practices and recommended cipher suites from RFC 9325 [50]. As described in detail in Appendix D, only 65.92% of the hosts use a recommended TLS cipher suite. This number decreases with servers that leak credentials (36.59%), expose personal data (44.20%), or expose internal information (61.42%).

Servers that leak credentials use recommended cipher suites significantly less frequently (see Table 13 in Appendix D). The same is true for servers for which we find at least one CVE and, to a lesser degree, also for servers that contain personal data. Servers that provide support for Simple Authentication and Security Layer or hold internally relevant, security-critical information, however, tend to fare better, but the situation is still not satisfactory. For example, around 40% of these servers still use other cipher suites than recommended by the RFC. Overall, this paints a picture where weaker cryptography correlates with sensitive data leakages.

TLS 1.3 experienced a fast deployment on the Web, in part

LDAP Client	Total	Rec.	Other	CBC mode
Apache LDAP Studio	38,269	71.82%	28.18%	11.97%
LDAP Administrator	38,545	69.98%	30.02%	13.15%
ldapsearch	37,775	69.86%	30.14%	12.44%
Outlook	38,771	70.69%	29.31%	12.27%
Thunderbird	38,560	66.18%	33.82%	11.89%

Table 9: Simulating the outcome of cipher suite negotiations using the cipher suites preferred by common LDAP clients.

because of the widespread use of cloud hosting [23]. After just a few years, the deployment already exceeded 30% on popular domains. For LDAP today, 36.67% of all our identified TLS-enabled servers support TLS 1.3. Once again, the numbers are significantly lower for servers that leak sensitive information.

X.509 Chain Validation. We find that over 50% of the X.509 chains are *globally invalid* as per our definition. The most common reasons for invalid chains are the use of self-signed certificates, expired or not yet valid certificates, and an unknown root certificate. For the latter case, we note the caveat of in-house CAs—we cannot detect these with our methodology, and hence this number needs to be treated as an upper bound. We note that self-signed certificates can be used in very bespoke situations to bypass the need for an in-house CA, *e.g.,* private servers used by one or very few clients. We summarize our results in Table 13. We find very few cases where chains are invalid for other reasons (less than 1%) (*e.g.,* self-signed certificates used to sign other certificates, too many intermediate certificates in the chain, violating path length constraints [5], or improper use of critical extensions).

Cipher Suites Chosen by LDAP Clients. After our initial analyses, we also ran reconfirmation scans to understand how well our scanning methodology approximates the cipher suite negotiation that common LDAP clients would achieve. We chose the five common LDAP implementations: Apache LDAP studio [3], LDAP Administrator [52], *ldapsearch* [40], Microsoft Outlook, and Thunderbird. We extract the cipher suite string they use in the handshake and use these in our scanner to determine which cipher suites LDAP servers would select. On inspection, we found that our Go library does not support some cipher suites that the clients do, so we do not test them. However, these do not offer a more secure connection than our library offers. However, servers should always choose recommended cipher suites, and we find that the servers do not reject our connection attempts due to a failure to negotiate a cipher suite. Therefore, clients should always be able to negotiate the same connection security our scanner can. Hence, we conclude that our method of simulating common LDAP clients is a good approximation.

We summarize the results of these tests in Table 9, reusing our categorization of recommended ciphers. A more detailed description of the results is presented in Appendix D. We find that the numbers are within a small margin of our previous scans, and in general, the percentage of recommended cipher suites negotiated is actually slightly higher. As these reconfirmation scans took place five months after the original scans, this could simply be due to servers that have received updates meanwhile rather than any difference in the cipher suite negotiation. Our conclusion is that our scanning methodology approximates the behavior of real LDAP clients very well.

7 Discussion

Following, we will discuss the implications of our results for the overall security of the Internet-wide LDAP landscape.

7.1 Personal Data and TLS

The analysis uncovered 12k public LDAP servers exposing personal data. Because of ethical and legal considerations and because of the large number of affected servers, we cannot ultimately determine if this data is supposed to be public. But even if it is supposed to be public, access to it should be encrypted. Take an LDAP server providing access to a public address book as an example. The people within this address book agreed to be listed there, so this information is not confidential. However, people accessing this address book may not want to reveal their searches in the address book because it leaks with whom they are communicating. Moreover, an active MitM could return fake contact results to the client, for example, to intercept the subsequent communication.

We found that out of 12k public LDAP servers exposing personal data, only 1,392 servers (11.4%) run a TLS configuration without issues that LDAP clients can use to authenticate the server and encrypt queries. Given that it is widely accepted that strong transport encryption is mandatory from a security viewpoint and easy to deploy, this is another indicator that LDAP has yet to receive wide attention from security professionals within organizations. Furthermore, the number drops substantially for servers that leak credentials (a mere 10%) or personal data (about 13%). It increases when servers are used for authentication or hold sensitive internal information. Servers that leak credentials or store personal data have particularly often poor configurations, both in terms of cipher suites and certificates. The strong correlation we see between weak cryptography and older TLS versions on the one hand, and data leakages and vulnerable LDAP implementations on the other hand, supports the hypothesis that the more weakly configured servers have not been updated in several years.

7.2 Public Directories vs. Social Engineering

Knowledge about the internal structure of organizations and, in particular, information about employees is an important indicator for the success of social engineering or phishing attacks. Collecting intelligence from open sources is complex and time-consuming because it involves a multitude of techniques and tools.[4] Overall, our analysis found that more than 12k organizations leak their organizational structure, configuration values of other services, or even detailed personal data about employees. These servers provide this information to attackers essentially for free. Even if organizations intentionally expose such information, it is questionable whether they are aware of the possible consequences.

7.3 Fast Track to Admin

Our analysis uncovered 1,817 servers leaking 3.9 million possible user credentials. As an example, the analysis uncovered that a university exposed more than 30,000 credentials, including email addresses and plaintext passwords, via a public LDAP server. This security lapse compromised a wide array of accounts, including those of IT administrators.

Even worse, we found 526 servers leaking passwords where at least one username contained the substring "admin". While it is not necessary for privileged users to have this substring in the username, it is a strong indicator that these servers indeed leak the credentials of privileged users. This is quite dangerous for the affected organizations as many modern ransomware cyberattacks exfiltrate and then encrypt victims' data to demand a ransom from victims. For this, the attackers not only need initial access to the victims' networks, but also privileged user accounts to get read and write access to data or to destroy backups by lateral movement. It is well-known that attackers may give up even if they have network access to a victim if they cannot get privileged user access. Our findings are hence relevant with respect to modern cyber-attackers: there is a substantial risk for organizations if they expose privileged user credentials via LDAP.

7.4 Current Status of Disclosure Efforts

In February 2024, We disclosed the list of credential-leaking servers to a national CERT, which then contacted the operators via email. To check the effectiveness of our disclosure campaign, three months later, we sent probes to the 1,817 server IPs previously found leaking credentials. These probes are limited to finding only one entry containing at least one of the password fields (see section 3.5). As in the previous analysis, we instructed the server not to transfer any data besides the `DN`. Out of the original 1,817 servers, 475 (26.1%) are no longer available. Five servers remain reachable but now require authentication. While we observed a significant reduction in password-leaking LDAP servers, a substantial amount remains online.

8 Ethical Considerations and Disclosure

Our overarching goal was to carry out an Internet-wide security analysis of public LDAP deployments to uncover and responsibly disclose security risks to affected operators. The nature of this work and the characteristics of LDAP led us to carefully consider the balance between improving the overall security of LDAP deployments and introducing potential harm or risk to others in the course of our work. We hence designed our steps and analyses carefully to prevent any unnecessary risks. For example, by interacting with LDAP servers, potentially sensitive data is transferred between our scanners and the scanned targets. In extreme cases, public LDAP servers may even be configured to return the full directory when prompted with a simple, standard LDAP query. To minimize the impact on users, we designed LanDscAPe specifically to only request and process very few samples from each LDAP server—even if the servers were configured to send more. To understand the grave problem of leaked credentials, we developed a method to count such credentials on a server *without* actually transferring them.

A further risk of our method lies in our scanning infrastructure becoming an attractive target itself, as attackers may aim at a cumulative view of our raw analysis results, including a list of LDAP servers leaking credentials. To mitigate this, the processing of the data samples happens on an internal server whose access is restricted by VPN to the authors of this paper. After finishing the classification of a sample, we *delete* it and only keep *metadata* for the disclosure campaign.

As a result, our analysis uncovered more than 1,800 LDAP servers that leak user credentials to anyone. We note that the bar for this is as low as knowing how to use a standard LDAP client. This is a potentially serious risk for well over 3 million users with data on these servers. On balance, we believe our minimally invasive testing is justified as it allows warning affected operators. Downloading a small sample set of credentials was necessary to reduce the rate of unnecessary reports to operators that do not expose real credentials or personal information (see Section 3.6). We conducted a responsible disclosure campaign with a national CERT. Our focus was on notifying those operators responsible for LDAP servers that leak highly critical information such as user credentials or sensitive internal information. Three months after our initial disclosure, more than one-fourth of the credential-leaking servers are no longer exposed to the Internet.

In our measurements, we followed standard best practices, as outlined by Durumeric *et al.* [9]. We sent only standard-conforming TCP and LDAP packets and TLS handshakes, using rate limiting, in randomized order of destination addresses to reduce strain on the network and servers. On the scanning IPs, a website gives project details and contact information for opt-out requests. We honored all such requests. Our ISP, national CERT, and university network administrator cleared all scans.

[4] See the MITRE ATT&CK about the "Reconnaissance" tactic (https://attack.mitre.org/tactics/TA0043/).

9 Related work

Hassler [19], along with Findlay, [13] Sermersheim [48] and Harrison [18], strongly endorsed using TLS to protect LDAP sessions while also emphasizing the importance of considering the possibility of its removal by adversaries. This is a particular issue when using StartTLS. Recent research by Poddebniak *et al.* highlighted vulnerabilities of in-band upgrades to TLS [41]. In this context, we note that security factors can change even during an LDAP session, necessitating additional considerations by servers [13, 18, 48]. Further security risks result from the absence of standardized access control mechanisms in LDAP, requiring each server to implement its own custom approach [13, 19]. In our work, we do not exploit any of these vulnerabilities, nor do we aim to find new ones. Our framework focuses on the analysis of a large set of LDAP servers on the Internet. It analyzes their network security and takes samples (ethically) to assess whether sensitive information is leaked.

Obimbo and Ferrima [39] demonstrated a Denial of Service (DoS) attack on LDAP, highlighting how the protocol can be abused. In 2017, the Project Sonar study by Rapid7 [47] investigated the prevalence of LDAP services on the Internet and fingerprinted them based on the root DSE. The study aimed to identify remote services using the Connectionless LDAP (CLDAP) protocol and also included LDAP scans on TCP ports 389 and 636. They found 300,663 servers on TCP port 389 and 91,842 servers on port 636 speaking LDAP. In our work, we found 80,968 servers on port 389 and 34,095 servers on port 636, indicating significant changes in this area over the past years. Additionally, their analysis assessed the effectiveness of CLDAP in Distributed Reflection Denial of Service (DRDoS) attacks. Srinivasa *et al.* [53] employed honeypots to study malicious intrusion attempts with various purposes, demonstrating that the protocol constitutes an attack surface that is quite commonly targeted by attackers. Our work does not use honeypots nor study DoS attacks; however, we focus on LDAP configurations and publicly accessible data, which may correlate with attack vectors as well, indicating the need for further scrutiny. Furthermore, our methodology employs a broader classification approach, considering additional parameters such as OIDs and LDAP attributes rather than solely relying on the root DSE to categorize LDAP services.

The X.509 PKI has been the subject of numerous publications, with multiple weaknesses identified for the Web as well as other communication protocols, often due to poor cipher suites and invalid certificate chains [2, 8, 21, 22]. Other studies indicated that TLS deployment is difficult even for expert users [30]. More recently, attention has focused on the root stores as well. Ma *et al.* investigated the root stores' operational ecosystem and exposed bad practices of multipurpose root stores and concentration in a root store [31]. Our analysis focuses on the deployment of certificate chains. A standard root store for LDAP does not exist; we hence relied on a combination of the ones available for the Web, which comprises all commonly recognized Certificate Authorities.

Compared to existing LDAP analysis tools, LanDscAPe is primarily designed to detect misconfigurations of LDAP servers at Internet-scale while at the same time transferring as little data as possible for ethical and performance reasons. Tools such as ldapsearch [40] and nmap's LDAP script [38] are designed to simplify communication with LDAP servers; however, they lack server analysis. Windapsearch [14], ldapper [54], and LDAPPER [51] focus on the enumeration of AD, while LanDscAPe minimizes harm by fetching samples instead and is applicable to all LDAP server types. Furthermore, LanDscAPe allows us to analyze servers by only transmitting small data samples using a novel counting algorithm (see section 3.6). Our tool also differs in connection modes. It connects to LDAP servers via anonymous, simple, and unauthenticated bind, while the other tools rely on authentication or switch to anonymous bind. Moreover, LanDscAPe can detect personal data, examine password policies, and find configuration files, whereas previously mentioned tools focus on pen-testing active directories. pbis [20] aims to integrate non-Windows devices into AD to extend security policies, which is outside the scope of our work.

10 Conclusion

We present the first study on public LDAP servers deployed on the Internet. The analysis uncovered more than 80k public LDAP servers, with 12k exposing personal data. Even worse, well over one thousand servers expose hashed or even cleartext passwords of users, resulting in a total of around 3.9 million exposed user credentials. Furthermore, we found that more than 10k servers expose possibly sensitive internal information, and more than 600 servers are linked to some CVE, suggesting they contain security vulnerabilities. We also found that although LDAP supports TLS, either implicit via port 686 or using StartTLS over port 389, the vast majority of LDAP servers supporting TLS have issues in the TLS configuration. This is either because it uses deprecated TLS versions or cipher suites, or because the certificate is self-signed, does not validate against common trust chains, or has expired. Interestingly, the TLS configuration of servers leaking personal data or user credentials is much worse than the TLS configuration of LDAP servers that may be used for user authentication. This indicates that servers leaking personal data or user credentials are not meant to be public.

Overall, the analysis strongly suggests that a surprisingly large number of organizations run LDAP servers leaking highly critical information, running non-optimal configuration values, and subpar TLS configurations. These LDAP servers are a treasure trove for cyberattackers, giving them important insights and possibly even access credentials to privileged users. Affected organizations need to quickly mitigate these risks by securing their LDAP server or removing them from the public Internet. We believe that this analysis sets the ground for future analyses in the security of the LDAP landscape on the Internet.

Acknowledgments We thank the USENIX reviewers and the USENIX Security Research Ethics Committee for their insightful comments on this paper. Furthermore, we thank Samson Umezulike and Marvin Dölzer for designing the database used for the statistical evaluations. We also thank Thomas 'Balu' Walter and Martin Renner for fruitful LDAP and Active Directory discussions. Special thanks to the German BSI CERT for the responsible disclosure efforts. We acknowledge the support by ip2location.com, which provided us with an academic license to use their data.

Jonas Kaspereit was supported by the research project 'Cyber Security Incident Response für KMUs (CySIRK)' (13FH101KB1) of the German Federal Ministry of Education and Research (BMBF) and the nicos AG. Gurur Öndarö was supported by the research project 'North-Rhine Westphalian Experts in Research on Digitalization (NERD II)', sponsored by the state of North Rhine-Westphalia – NERD II 005-2201-0014. Gustavo Luvizotto Cesar was supported by the research project 'CATRIN' (NWA.1215.18.003) as part of the Dutch Research Council's (NWO) National Research Agenda (NWA). This research was supported by the National Research Center for Applied Cybersecurity ATHENE.

References

[1] Akamai. Cldap reflection ddos threat advisory. https://www.akamai.com/glossary/what-is-a-cldap-reflection-ddos-attack, 2017. Accessed 2024-05-20.

[2] Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, and Ralph Holz. Mission accomplished? https security after diginotar. In *ACM/USENIX Internet Measurement Conference (IMC)*, 2017.

[3] Apache Software Foundation. Apache Directory Studio. https://directory.apache.org/studio/, 2003. Accessed 2024-05-03.

[4] Apple. Apple Open Directory Documentation. https://developer.apple.com/documentation/opendirectory. Accessed 2023-09-11.

[5] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and David Cooper. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, May 2008.

[6] Gerald Carter. *LDAP System Administration: Putting Directories to Work*. "O'Reilly Media, Inc.", March 2003.

[7] Deland-Han. Enable Lightweight Directory Access Protocol (LDAP) over Secure Sockets Layer (SSL) - Windows Server. https://learn.microsoft.com/en-us/troubleshoot/windows-server/identity/enable-ldap-over-ssl-3rd-certification-authority. Accessed 2023-09-11.

[8] Zakir Durumeric, James Kasten, Michael Bailey, and J. Alex Halderman. Analysis of the HTTPS certificate ecosystem. In *ACM Internet Measurement Conference (IMC)*, 2013.

[9] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. ZMap: Fast internet-wide scanning and its security applications. In *22nd USENIX Security Symposium*, 2013.

[10] Marco Eichelberg, Klaus Kleber, and Marc Kämmerer. Cybersecurity challenges for pacs and medical imaging. *Academic Radiology*, 27(8):1126–1139, May 2020.

[11] Oliver Gasser et al. Github repository for goscanner. https://github.com/tumi8/goscanner, 2017. Accessed 2024-01-02.

[12] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council. Accessed 2023-04-13.

[13] Andrew Findlay. Best Practices in LDAP Security. https://www.skills-1st.co.uk/papers/ldap-best-2011/best-practices-in-ldap-security.pdf, 2011. Accessed 2024-05-10.

[14] Ronnie Flathers. ropnop/windapsearch. https://github.com/ropnop/windapsearch, May 2024. Accessed 2024-05-16.

[15] OpenLDAP Foundation. OpenLDAP Software 2.4 Administrator's Guide: Access Control. https://www.openldap.org/doc/admin24/access-control.html. Accessed 2023-10-11.

[16] Roger Harrison. Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms. Request for Comments RFC 4513, Internet Engineering Task Force, June 2006.

[17] Roger Harrison. Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms. RFC 4513, June 2006.

[18] Roger Harrison. Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms. Request for Comments RFC 4513, Internet Engineering Task Force, June 2006.

[19] V. Hassler. X.500 and LDAP security: a comparative overview. *IEEE Network*, 13(6):54–64, November 1999. Conference Name: IEEE Network.

[20] Joseph Anthony Pasquale Holsten. josephholsten/pbis. https://github.com/josephholsten/pbis, February 2023. Accessed 2024-05-16.

[21] Ralph Holz, Johanna Amann, Olivier Mehani, Matthias Wachs, and Mohamed Ali Kaafar. TLS in the wild: An internet-wide analysis of TLS-based protocols for electronic communication. In *Network and Distributed System Security Symposium 2016*, 2016.

[22] Ralph Holz, Lothar Braun, Nils Kammenhuber, and Georg Carle. The SSL landscape: a thorough analysis of the X.509 PKI using active and passive measurements. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11. Association for Computing Machinery, 2011.

[23] Ralph Holz, Jens Hiller, Johanna Amann, Abbas Razaghpanah, Thomas Jost, Narseo Vallina-Rodriguez, and Oliver Hohlfeld. Tracking the deployment of TLS 1.3 on the web: A story of experimentation and centralization. *SIGCOMM Comput. Commun. Rev. (CCR)*, 2020.

[24] Jian Hu, Qizhi Sun, and Hongping Chen. Application of Single sign-on (SSO) in Digital Campus. In *2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, October 2010.

[25] International Telecommunication Union. X.500: Information technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services. https://www.itu.int/rec/T-REC-X.500-201910-I/en, December 2012. Accessed 2023-10-11.

[26] International Telecommunication Union. X.501: Information technology - Open Systems Interconnection - The Directory: Models. https://www.itu.int/rec/T-REC-X.501-201910-I/en, October 2019. Accessed 2023-10-11.

[27] Liz Izhikevich, Renata Teixeira, and Zakir Durumeric. LZR: Identifying unexpected internet services. In *Proc. USENIX Security Symposium*, 2021.

[28] janicericketts. LDAP authentication with Azure Active Directory - Microsoft Entra. https://learn.microsoft.com/en-us/azure/active-directory/architecture/auth-ldap, July 2023. Accessed 2023-09-11.

[29] V. Karatsiolis, M. Lippert, and A. Wiesmaier. Planning for Directory Services in Public Key Infrastructures, January 2005. arXiv:cs/0411068.

[30] Katharina Krombholz, Wilfried Mayer, Martin Schmiedecker, and Edgar Weippl. "I Have No Idea What I'm Doing" - On the Usability of Deploying {HTTPS}. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1339–1356, 2017.

[31] Zane Ma, James Austgen, Joshua Mason, Zakir Durumeric, and Michael Bailey. Tracing your roots: Exploring the TLS trust anchor ecosystem. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 179–194, Virtual Event, November 2021. ACM.

[32] Metasploit. Metasploit project. https://www.metasploit.com. Accessed 2023-07-21.

[33] Microsoft. Access control lists - Win32 apps. https://learn.microsoft.com/en-us/windows/win32/secauthz/access-control-lists. Accessed 2023-10-11.

[34] Microsoft. Exchange account server settings - Microsoft Support. https://support.microsoft.com/en-us/office/exchange-account-server-settings-5025680d-5b3e-441e-93ed-c073a82cc364. Accessed 2023-09-11.

[35] Giovane C. M. Moura, Carlos Gañán, Qasim Lone, Payam Poursaied, Hadi Asghari, and Michel van Eeten. How dynamic is the ISPs address space? Towards internet-wide DHCP churn estimation. In *2015 IFIP Networking Conference (IFIP Networking)*, May 2015.

[36] National Electrical Manufacturers Association (NEMA). The DICOM standard, part 15. https://dicom.nema.org/medical/dicom/2023e/output/html/part15.html, 2023. Accessed 2024-01-24.

[37] Glen Newell. An introduction to Linux Access Control Lists (ACLs). https://www.redhat.com/sysadmin/linux-access-control-lists. Accessed 2023-10-11.

[38] Nmap.org. ldap-search NSE script — Nmap Scripting Engine documentation. https://nmap.org/nsedoc/scripts/ldap-search.html. Accessed 2024-05-16.

[39] Charlie Obimbo and Benjamin Ferriman. Vulnerabilities of LDAP As An Authentication Service. *Journal of Information Security*, 02(04):151–157, 2011.

[40] OpenLDAP Project. The ldapsearch Command-Line Tool. https://docs.ldap.com/ldap-sdk/docs/tool-usages/ldapsearch.html. Accessed 2024-05-16.

[41] Damian Poddebniak, Fabian Ising, Hanno Böck, and Sebastian Schinzel. Why TLS is better without STARTTLS: A security analysis of STARTTLS in the email context. In *Proc. USENIX Security*, 2021.

[42] Red Hat. Administration Guide Red Hat Directory Server 11. https://access.redhat.com/documentation. Accessed: 2024-01-23.

[43] John Rhoton. *Programmer's Guide to Internet Mail: SMTP, POP, IMAP, and LDAP*. Digital Press, October 1999. Google-Books-ID: Hs0JSZHajyIC.

[44] Patrick Sattler, Johannes Zirngibl, Mattijs Jonker, Oliver Gasser, Georg Carle, and Ralph Holz. Packed to the brim: Investigating the impact of highly responsive prefixes on internet-wide measurement campaigns. In *Int. Conf. on Emerging Networking Experiments And Technologies (CoNEXT)*, 2023.

[45] Anew Sciberras. Lightweight Directory Access Protocol (LDAP): Schema for User Applications. Request for Comments RFC 4519, Internet Engineering Task Force, June 2006.

[46] SecurityScorecard. CVE security vulnerability database. Security vulnerabilities, exploits, references and more. https://www.cvedetails.com/index.php. Accessed 2024-05-17.

[47] Tom Sellers. Project sonar study of ldap on the internet: Rapid7 blog. https://www.rapid7.com/blog/post/2016/11/08/project-sonar-study-of-ldap-on-the-internet/, Aug 2017. Accessed 2024-05-21.

[48] Jim Sermersheim. Lightweight Directory Access Protocol (LDAP): The Protocol. Request for Comments RFC 4511, Internet Engineering Task Force, June 2006.

[49] Jim Sermersheim, Ludovic Poitou, Howard Chu, and Ondřej Kuzník. Password Policy for LDAP Directories. Internet Draft draft-behera-ldap-password-policy-11, Internet Engineering Task Force, February 2022.

[50] Yaron Sheffer, Peter Saint-Andre, and Thomas Fossati. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). RFC 9325, November 2022.

[51] shellster. shellster/LDAPPER. https://github.com/shellster/LDAPPER. Accessed 2024-05-19.

[52] Softerra. LDAP Administrator - Powerful Directory Management Tool. https://www.ldapadministrator.com/. Accessed 2024-05-23.

[53] Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. Deceptive directories and "vulnerable" logs: a honeypot study of the ldap and log4j attack landscape. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2022.

[54] Zach Stein. Synzack/ldapper. https://github.com/Synzack/ldapper. Accessed 2024-05-19.

[55] Nick Sullivan. Padding oracles and the decline of CBC-mode cipher suites. https://blog.cloudflare.com/padding-oracles-and-the-decline-of-cbc-mode-ciphersuites, February 2016.

[56] Yuli Wang, Jiayin Tian, Cheng Yang, and Yaping Zhu. The research and design of unified authentication system based on CAS. In *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, pages 528–532, September 2017.

[57] Kurt Zeilenga. Lightweight Directory Access Protocol (LDAP): Directory Information Models. Request for Comments RFC 4512, Internet Engineering Task Force, June 2006.

[58] Kurt Zeilenga and Alexey Melnikov. Simple Authentication and Security Layer (SASL). Request for Comments RFC 4422, Internet Engineering Task Force, June 2006.

Appendix

A LDAP Server Confirmation

LDAP Error	**Port 389**		**Port 636**	
Result Codes	**v2**	**v3**	**v2**	**v3**
authMethodNotSupported	4	4	19	19
authorizationDenied	2	2	-	-
confidentialityRequired	12	100	-	-
inappropriateAuthentication	2,810	7,578	487	3,636
insufficientAccessRights	98	330	-	145
invalidCredentials	1,159	1,154	1,497	1,575
invalidDNSyntax	10	10	1	1
namingViolation	-	6	-	8
operationsError	87	102	3	15
protocolError	32,353	56	12,870	-
strongerAuthRequired	3	3	-	-
timeLimitExceeded	-	18	-	1
unavailable	1	1	-	1
unwillingToPerform	44	68	97	93

Table 10: LDAP error result codes.

Table 10 shows the errors for failed LDAPv2 and LDAPv3 binds. We observe that the majority of LDAPv2 bind requests were met with a `protocolError` response, indicating dwindling legacy support. Servers also commonly disallow anonymous binds (`inappropriateAuthentication`) or mark empty credentials as invalid (`invalidCredentials`).

Server Types	Total	OID	Version	Attributes
Linux User Man.	5,180	0	0	5,180
Microsoft Active Dir.	4,465	4,465	0	51
Sun Java System Dir.	1,818	1,818	0	0
iPlanet Directory	1,592	1,592	0	0
Apple Schema for AD	1,387	0	0	1,387
389 Directory Server	1,348	0	1,348	0
Apple with OpenLDAP	574	0	574	0
ApacheDS	171	0	171	0
DICOM	145	0	0	145
HL7	141	0	0	141

Table 11: Overview of the top 10 categorized server types and the categorization methods.

B Categorization of Servers

We have categorized the LDAP servers based on the OIDs they expose, their version number, and the attributes they use. Table 11 shows the top 10 list of all categorized server types.

C Personal Data Attribute Groups

We have extracted all attributes from the LDAP samples that potentially contain personal data and categorized them into general groups. The classification of the identified attributes into groups is shown in Table 12.

D Analysis of TLS and X.509

Only about 60% (56k) of the candidate hosts respond to our scans with a successful TLS connection. This is expected: recent findings by Sattler *et al.* [44] and Izhikevich *et al.* [27] show that many IPv4 addresses respond to a SYN packet but do not follow through with a full TCP handshake.

TLS Versions. TLS 1.3 experienced an unusually fast deployment on the Web, in part because of the widespread use of cloud hosting [23]. After just a few years, the deployment already exceeded 30%. We also find substantial deployment of this version (which mandates many strong security features) also for LDAP, with deployment at nearly 37% of all analyzed hosts. Once again, the numbers are significantly lower for the servers that leak information—and better for those that support authentication. Still, TLS 1.2, which has come under significant pressure in recent years, is still by far the most widely deployed version, especially for servers with data leakages.

Cipher Suites. Numerous cipher suites are defined for use in TLS. They indicate the symmetric cipher, block mode (if applicable), and the hash algorithm for the Message Authentication Code (MAC). In addition, they indicate how the session key for the symmetric cipher is to be derived. In our evaluation, we generally refer to RFC 9325 [50], which defines best practices for TLS. In particular, this means that the key exchange should be ephemeral Diffie-Hellman (DHE) to enable forward security, ideally on elliptic curves. Classic finite-field Diffie-Hellman is discouraged for TLS 1.2, and non-ephemeral Diffie-Hellman is generally viewed as too weak. Using RSA in the key exchange means that forward secrecy is not possible; the RFC advises strongly against this.

Personal Data Group	LDAP Attributes
Last Name	lastName, LastName, sn, sn;lang-cs, sn;lang-en, sn;lang-ja, sn;lang-el sn;x-role-2, uvmEduSurname, i3sicLastName, suDisplayNameLast
Email	EMAIL, eMailAddress, email, i3sicEmail, mail, mailAlias, mailAlternateAddress, mailLocalAddress, mailalias, zimbraMailAlias, umMailAlias, umAlternateMail, suMailAddress, mailAliasLast, mailAliasLastLaw, mailAliasesLaw, mailAliases
First Name	firstName, FirstName, givenName, givenName;lang-cs, givenName;lang-ja, givenName;lang-el, givenName;x-role-2, givenname, i3sicFirstName, middleName, umDisplayFirstName, suDisplayNameFirst, suDisplayNameMiddle, umMiddleInitial, ucMiddleName, cuMiddlename
Full Name	displayName, displayname, eduPersonNickname, umDisplayName, umDisplayNameLF, krbPrincipalName, username, nsNickName, umNickName, nsnickname, mozillaNickname, nickname, fullName, name, CallerIDName, suDisplayNameLF, umNameComponent, suOtherName
Birthday	apple-birthday, birthDay, dateOfBirth, schacDateOfBirth, birthDate
Gender	gender, sex, schacGender
Country	c, co, countryCode, state, st, country, userCountry, mozillaHomeState
Phone	homePhone, telephoneNumber, uvmEduOfficePhone, mobile, i3sicLocalPhoneNumber, facsimileTelephoneNumber, MobileNumber, HomeNumber, suGwAffilPhone1, telephonenumber, otherTelephone, osuAltPhoneNumber, otherMobile, phone
Address	homePostalAddress, streetAddress, street, postalAddress, uvmEduOfficeAddress, umStreetAddress, suGwAffilAddress1, mozillaHomeStreet, postalAddress;x-role-2, osuOfficeAddress, mozillaHomeStreet2
Location	l, location, uvmEduOfficeLocation, mozillaHomeLocalityName, provinceName, postalCode, zip, city
Public Key	userCertificate, userCertificate;binary, pgpKey, nDSPKIPublicKeyCertificate, nDSPKIPublicKey, userPKCS12, userSMIMECertificate
Title	isDoctor, title, title;x-role-2, personalTitle, umOfficialTitle, umDisplayTitle, umPrimaryTitle, umNamePrefix, umNameSuffix, suDisplayNamePrefix, suDisplayNameSuffix
Job	company, department, employeeNumber, employeeType, umEmployee, eduPersonAffiliation, eduPersonPrimaryAffiliation, Department, umPrimaryDeptName, umPrimaryInstitutionCode, umInstitutionCode, umPrimaryInstitution, umDepartment, suGwAffiliation2, suDisplayAffiliation, suGwAffiliation1, suAffiliation, ucDepartment, departmentName, osuPrimaryAffiliation, osuDepartment, companyName, brEduAffiliation, psDepartment, hireDate
Photo	jpegPhoto, photograph
SSN	socialSecurityNumber

Table 12: Personal data attribute groups.

RSA can be used for authentication. In this case, the RFC requires the key to be at least 2048 bit long. Concerning block ciphers, the RFC views GCM (Galois Counter Mode)

	Total IPs	Leak credentials	Personal data	SASL support	CVE	Internal info.
Supporting TLS	48,198 (100%)	910 (100%)	7,378 (100%)	18,502 (100%)	528 (100%)	7,118 (100%)
TLSv1.3	36.67%	20.00%	27.24%	42.04%	28.41%	49.34%
TLSv1.2	59.46%	77.03%	68.91%	55.92%	71.59%	46.90%
TLSv1.1	0.33%	0.33%	0.16%	0.23%	0.00%	0.10%
TLSv1.0	3.54%	2.64%	3.69%	1.81%	0.00%	3.67%
Recommended cipher suites	65.92%	36.59%	44.20%	57.49%	94.89%	61.42%
Other cipher suites	34.08%	63.41%	55.80%	42.51%	5.11%	38.58%
. . . with RSA key exchange	24.27%	57.14%	53.71%	41.77%	5.11%	36.93%
. . . using CBC	12.26%	8.35%	6.30%	3.64%	5.11%	5.49%
. . . using 3DES	0.31%	0.00%	0.03%	0.00%	0.00%	0.00%
Valid Cert. Chain	36.81%	17.69%	24.10%	40.38%	32.20%	47.78%
Invalid Cert. Chain	63.19%	82.31%	75.90%	59.62%	67.80%	52.22%
... Self-signed	32.30%	21.87%	22.61%	16.53%	1.70%	10.97%
... Expired/not yet valid	19.65%	43.63%	35.52%	25.15%	6.63%	14.34%
... Unknown authority	11.20%	16.70%	17.74%	17.91%	59.28%	26.89%

Table 13: Overview of the results of the Network Analysis Module.

as secure, whereas Cipher Block Chaining (CBC) is under pressure due to padding oracle attacks [55] possible in some contexts. This can be mitigated when CBC is negotiated with the "encrypt_then_mac" TLS extension. However, none of the major LDAP clients we tested sent this extension. Therefore, we excluded it from our scans. Older hashing algorithms such as SHA1 are deprecated; the more modern SHA2 and SHA3 or the Poly1305 family of hash functions should be used instead. The effective length of the symmetric session keys should never be less than 128 bits; the RFC explicitly mentions 3DES as an example of a cipher with insufficient key length. We implemented the most relevant parts of RFC 9325 to evaluate the use of TLS in our data set.

Table 13 summarizes our findings. We find that a decent number of connections (around 65%) use one of the recommended cipher suites. Across all TLS-supporting LDAP instances, we find that AES-GCM is widely used, often even with a secure key length of 256 bit. The modern stream cipher ChaCha20 is in significant use as well (20%). However, servers that leak credentials commonly use recommended cipher suites significantly less frequently. The same is true for server versions for which we find at least one CVE and, to a lesser degree, also for servers that contain personal data. Servers that provide support for SASL (Simple Authentication and Security Layer) or hold internally relevant, security-critical information, however, tend to be more commonly configured with better cryptography. Yet, this is still unsatisfactory: for example, 38% of servers that hold information of the latter kind still use cipher suits besides the recommended ones. Overall, this paints a picture where weaker cryptography correlates with leakages of sensitive data.

Analyzing X.509 Chains. We call certificate chains (globally) invalid if the root CA is not in the root store, but also if a CA uses the SHA1 algorithm (as modern libraries such as the Go standard library reject these certificates as invalid).

The use case of LDAP is often different from that of the Web, which is almost by definition a publicly accessible resource. As data on LDAP servers may only need to be accessible to very few clients, it is conceivable that an organization operates a custom in-house CA to issue certificates for the LDAP server and deploys the CA's root certificate only to its clients. Microsoft has tooling for such a purpose, for example. To our analysis, such certificates would appear to have invalid chains as the root CA is not part of any root store. Hence, when we check for valid chains, our numbers are a *lower bound* (and an upper bound for invalid chains). Custom CAs can be expected to issue relatively few certificates (as there are few servers per organization).

Common LDAP Clients and Negotiated Cipher Suites. To understand what cipher suites common LDAP clients would negotiate, we extracted their list of supported cipher suites from the *Client Hello* messages they send. Table 9 shows the results. This analysis was done several months after our initial scans.

Summarizing View. Table 13 gives an overview of what we consider TLS configurations with and without issues, breaking down the reasons into cipher suites (recommended by the RFC vs. others) and certificate chains. We observe that servers that leak credentials or personal data fare worse than the total regarding selected cipher suites and valid X.509 chains. Furthermore, we verified that cipher suite negotiation that major LDAP clients carry out results in very similar results as in our own scans. The correlation we see between weak cryptography and older TLS versions on the one hand, and data leakages and vulnerable LDAP versions on the other hand, would support the hypothesis that the more weakly configured servers have not been updated in several years.