**CA基本态势测绘：**

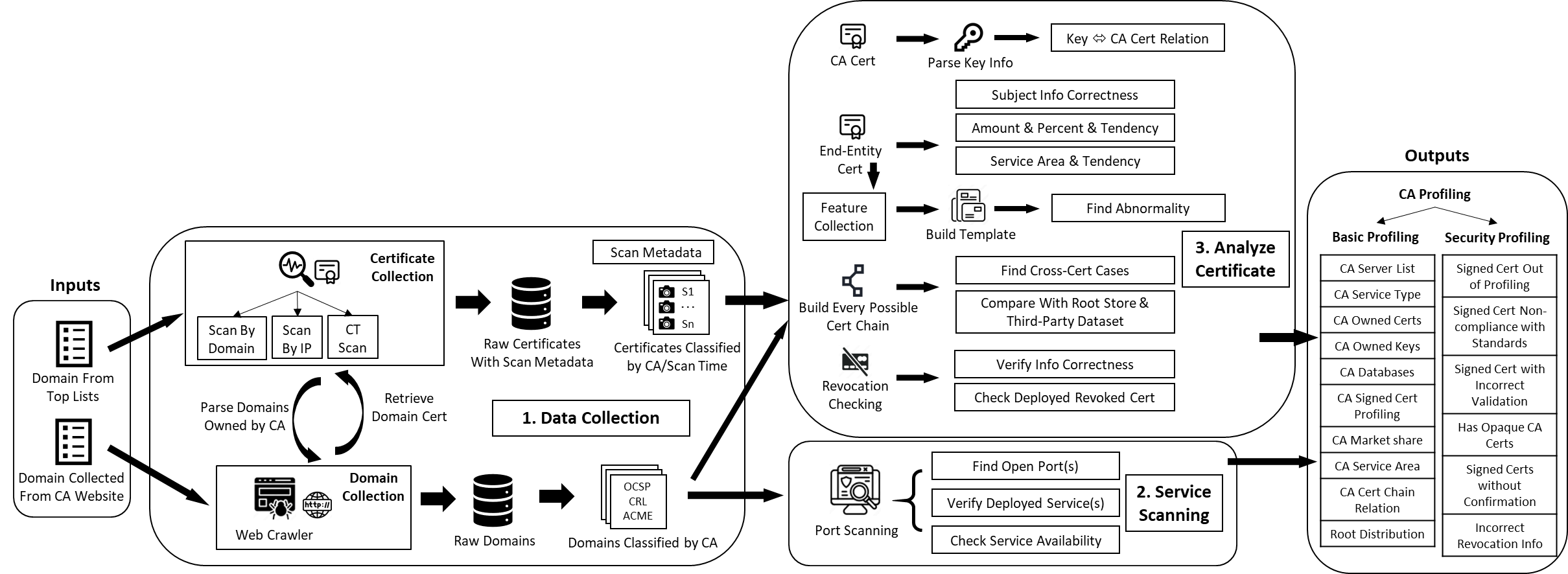
**分类标准：CA作为商业机构，在 PKI 中的行为**

|  |  |  |
| --- | --- | --- |
| **CA 行为** | **测量内容** | **测量方法** |
| **构建、维护作为CA企业的基本基础设施** | **服务基础设施（包括IPv4/v6）**  **HTTPS服务器：**   1. 提供ACME服务的服务器（以及测试部署环境服务器） 2. 用于验证申请者身份的服务器（IP地址会随时改变，会使用云服务器） 3. CT日志服务器（部分CA有维护） 4. 提供证书管理平台服务的服务器   **HTTP服务器：**   1. CRL/OCSP 服务器（以及测试部署环境服务器） 2. 签发者（储存CA证书 ）服务器   **服务器部署的服务：**   1. 服务器开放的端口 2. 服务器部署的具体服务类型 | **HTTPS服务器：**   1. 从CA 网站的公开文档中找到部分正在使用的网站域名作为seed 2. 获取seed网站的证书，获取SAN字段中的相似域名 3. 从IPv4/CT扫描的证书中搜寻包含与seed相同/相似域名 (比如域名含有acme/ocsp) 的证书，记录证书中所有相似域名（需确认subject是否一致） 4. 对于部署 HTTPS服务的网站，使用爬虫爬取页面上的相似域名链接 5. 以现有的集合作为seed重复2-4，直到结果收敛 6. 对于CA验证身份的服务器，需要考虑手动向CA申请证书后，抓取网络中的报文判断服务器的域名/IP 地址   **HTTP 服务器：**   1. CRL 服务器可通过证书CRL Distribution Point扩展字段中获得 2. OCSP服务器和签发者服务器可以从证书AIA扩展字段中获得 3. 测试服务器所签发的测试证书需要从专门存储测试证书的CT日志中获得   **服务探寻：**   1. 对搜寻到的服务器进行全端口扫描 2. 尝试获取解析开放端口以及部署的服务类型 |
| **加密认证基础设施**  **加密密钥对：**   1. 公钥的具体数据（私钥无法获取） 2. 密钥对类型（常用类型为RSA/ECDSA） 3. 密钥对参数（如RSA exponent，RSA密钥长度等） 4. 密钥对使用范围（TLS 认证/代码签名认证等）   **CA证书（Signing Certificates）：**   1. CA 拥有哪些CA证书 2. 这些CA证书处于证书链的何种位置（中间/根证书） 3. CA证书与密钥对的关系（一个密钥对是否对应多个CA证书） | **CA证书：**   1. IPv4/CT扫描获取证书链中的所有证书，根据subject与issuer的关系以及Basic Constraint字段区分终端证书/CA证书 2. 从终端证书中的AIA字段中获得对应的签发CA证书，与1中分类的CA证书一起作为扫描证书集合 3. 从CA公开的审计日志中获取CA证书列表   **加密密钥：**   1. 提取2-3步骤中所有证书的public key algorithm字段，以知晓密钥的类型 2. 根据密钥的类型提取public key字段的内容，并获取相应的密钥参数，公钥值和对应参数决定了一个独一无二的密钥对 3. 根据公钥的具体值/参数判断某个密钥对是否对应多个证书 |
| **数据库基础设施：**   1. 证书状态数据库（OCSP/CRL） 2. CT 日志数据库 3. CP/CPS文档（不需要测量） | **OCSP/CRL 数据库：**   1. 通过IPv4/CT扫描获得尽可能多的证书 2. 根据证书字段中CRL/OCSP 验证地址进行状态验证，存储结果 3. 如果证书没有提供验证地址，则根据证书的签发 CA 与前文搜集的所有 CA CRL/OCSP 服务器依次验证 4. 从2中抽取一部分证书，在其他的OCSP/CRL服务器进行状态验证，以观察是否存在数据不一致的情况 5. 针对未过期的证书，按照一定的频率统计证书状态的变化   **CT日志数据库（不重要）：**   1. 知晓CT日志的大小 2. 知晓CT日志信任的根证书列表 3. 统计CT日志中存储的证书签发者等字段特点即可 |
| **签发终端证书（叶证书）——为终端的可信性背书** | **静态特征：**   1. CA身份验证方式 (TLS ALPN-01/HTTP-01/DNS-01等) 2. CA签发终端证书的Profiling (签发证书每一个字段的特点) 3. CA签发终端证书的总数量/市场占比，互联网中哪些活跃证书是CA签发的（证书占比） 4. CA给哪些国家/地区的终端签发证书（服务范围） | **身份验证方式：**   1. 从CA 的官方文档中获得CA常用的验证方式 2. 购买CA的PKI证书服务，在申请证书时获得验证身份的所有选项   **签发证书Profiling：**   1. 通过IPv4/CT扫描获得终端证书，按照 (issuer\_cn, issuer\_org, issuer\_country) 三元组的方式分类 2. 假设每一类证书只有一个profiling，统计每一类证书各个字段信息，归纳出该profiling 3. 取消2中的假设，提取证书内容的字段信息，建立证书的fingerprint 4. 对3中特征化后的证书使用聚类算法（如Kmeans++），得出若干个证书的profiling 5. 比较2和4得到的profiling，验证合理性   **签发证书占比：**   1. 统计所有获得的终端证书的CA占比，以估计CA的市场占比 2. 统计IPv4扫描到的活跃证书的CA占比，以估计CA部署占比   **服务范围：**   1. 统计CA签发证书 subject 字段的信息，主要记录subject\_country 2. 如果证书没有subject\_country，如果该证书是活跃证书，可以采取IP-ASN查询证书的部署位置 3. 如果不是活跃证书，可以通过WHOIS查询该证书域名的国家/地区 |
| **动态特征：**   1. CA签发的活跃证书占比变化趋势 2. CA签发所有证书的数量增长趋势 3. CA服务范围的动态变化 4. CA实时验证身份的方法 | **活跃证书占比变化：**   1. 比较不同时间扫描证书的CA占比，估计CA活跃证书的占比变化   **签发数量变化：**   1. 定期获取CT日志增量，从中记录CA新签发的证书 2. 比较不同时间扫描证书结果，记录新出现的证书，作为CA新签发的证书   **服务范围变化：**   1. 比较不同时间扫描证书结果，分析增加/减少服务的地理位置   **身份验证：**   1. 在某主干网中实时获取来自/去往CA身份认证服务器的流量，分析流量的类别，判断CA使用的验证方式 2. 自己购买域名部署服务器 (Web, DNS)，主动向CA申请证书，根据服务器收到的报文判断CA的验证方式 |
| **建立完整证书信任链关系——CA需要得到别人的信任** | **静态特征：**  **证书链结构：**   1. CA 证书中的密钥签发了哪些证书 2. CA证书的签发者是谁，是否有多个签发者 3. 证书信任链是否跨越多个CA   **证书兼容性：**   1. CA 根证书存在于哪些常用的根证书库 (OS/Browser) 或第三方数据库 (CCADB) 中 2. CA 的根证书在根证书库中何时被添加/移除 | **证书链结构：**   1. 精确到密钥级别，统计CA证书的密钥签发了哪些证书 2. 在CA 证书集合中，寻找每一个 CA 证书的签发证书（可能有多个） 3. 综合1-2的结果，构建所有可能的证书链（证书网） 4. 分析3中的所有证书信任链是否存在/哪里存在跨越多个CA的现象   **证书兼容性：**   1. 搜集常用浏览器/OS 对应的根证书库，以及第三方数据库信任的 CA证书 2. 对收集到的所有CA根证书，检查其存在于哪些根证书库和第三方数据库 3. 若能搜集到根证书库的历史版本，可以检测根证书的历史添加/移除情况 |
| **动态特征：**   1. 根证书在不同国家/地区用户端的实时兼容性 | 需要从客户端的角度收集根证书库数据，目前无法测量 |
| **吊销证书——终止证书信任关系** | CA实时验证证书吊销的方法 | 与判断CA验证身份的方法一样，需要自己搭建服务器 |

**安全态势**

**分类标准：真实发生过的CA安全事件**

|  |  |  |
| --- | --- | --- |
| **CA安全事件类型** | **测量内容** | **测量方法** |
| **CA 签发的证书内容违反了标准——CA 签发证书内容一致性态势** | 验证 CA 签发的证书：  (1) 内容是否符合 CA签发证书的profiling  (2) 内容是否符合国际标准 | (1) 在使用统计方法构建CA签发证书的profiling时，如果出现了特殊的类别，则对应的证书存在问题  (2) 在使用聚类算法构建CA签发证书的profiling时，如果出现了离群点，则对应的证书存在问题  (3) 使用 Zlint 等规则验证工具，验证证书内容是否符合国际标准 |
| **CA 错误验证申请者身份——CA 验证身份正确性态势** | 验证 CA 签发的证书：  (1) subject信息（域名、组织和国家）是否一致  (2) 如果为活跃证书，部署证书的 IP 地址是否被证书 subject所管理控制  (3) CA 使用的验证身份的方式是否存在漏洞（如CAA 验证是否正确） | (1) 通过WHOIS等方式查询域名的拥有者是否是对应的组织/国家，组织和国家之间关系是否正确  (2) 通过IP-ASN等方式查询IP拥有者是否是对应的subject  (3) 仍然需要自己主动部署服务器，申请证书 |
| **CA 私自建立证书链验证关系——CA 构建证书链证书公开性态势** | (1) 验证根证书库中信任的根证书关系是否公开  (2) 验证 CA 证书是否公开  (3) 验证终端证书是否公开 | **验证根证书库信任关系：**  (1) 查询根证书库、第三方数据库是否有相关公告  **验证 CA 证书：**  (1) 查询 CA 是否有相关公告说明证书的签发  (2) 该证书是否在第三方认证报告中出现 （如 WebTrust）  (3) 是否在CA签发者服务器中出现  **验证终端证书：**  (1) 该证书是否在正确的网站部署，该证书部署的时间长度是多少  (2) 有没有错误报告说明该证书是有问题的 |
| **CA 错误更改证书状态信息——CA 存储证书吊销信息正确性态势** | (1) CA只能存储自己签发证书的证书状态  (2) CA存储的证书状态要与证书的真实状态一致  (3) CA存储证书状态至少到证书过期 | **按照以下规则验证前文构建的CA证书状态数据库：**  (1) 非自己签发的证书的结果应为 Unauthorized  (2) 未吊销证书结果应始终为 Good 直到过期  (3) 吊销证书后，结果应始终为 Revoked 直到过期  (4) 证书过期后，结果不应为 Good  (5) CRL/OCSP 储存的信息应一致  (6) CRL/OCSP 信息的更新需满足要求 |

**CA测绘框架：**