安全+
SECURITY
2021/12 总第 051
技术版 ▶▶ 与安全人士分享技术心得 Share technique experience with security professionals

★ 本期焦点

如何利用威胁情报报告生成可用威胁子图

关于构建数据安全生态圈的研究与实践

逃逸风云再起：从CVE-2017-1002101到CVE-2021-25741

权利义务框架下的移动互联网APP个人信息保护
——《个人信息保护法》对APP个人信息保护影响分析

绿盟科技官方微信

NSFOCUS

SECURITY

安全+ 技术版
与安全人士分享技术心得
Share technique experience with security professionals

本期看点 HEADLINES

11 如何利用威胁情报报告生成可用威胁子图

24 关于构建数据安全生态圈的研究与实践

45 逃逸风云再起：从CVE-2017-1002101到CVE-2021-25741

70 权利义务框架下的移动互联网APP个人信息保护
——《个人信息保护法》对APP个人信息保护影响分析

主办：绿盟科技
策划：《安全+》编委会
地址：北京市海淀区北洼路4号益泰大厦三层
邮编：100089
电话：(010)6843 8880-5463
传真：(010)6872 8708
网址：www.nsfocus.com

2021/12 总第 051

欢迎您扫描封面左下角的二维码，关注绿盟科技官方微信，
分享您的建议和评论，或者来信nsmagazine@nsfocus.com
与我们交流。(《安全+》部分图片来源于网络)

安全+ SECURITY

目录 CONTENTS

卷首语	叶晓虎	2
技术前沿		3-23
云原生应用安全防护思考（一）	浦明	3
如何利用威胁情报报告生成可用威胁子图	薛见新	11
基于规则向量化的 HTTP 资产识别方法探索	张卓 张迎苹	17
能力构建		24-44
关于构建数据安全生态圈的研究与实践	曾令平	24
波谲云诡，浅谈工业互联网的跨界安全挑战	任心	32
绿盟科技智慧安全 3.0 助力车联网行业网络安全实践	刘大鹏	36
浅谈电子政务密码应用推进思路	李成日	41
攻防对抗		45-61
逃逸风云再起：从 CVE-2017-1002101 到 CVE-2021-25741	阮博男	45
人工智能赋能网络靶场创新发展	孙翔	58
安全趋势		62-72
个人信息安全法律保护伞 ——《中华人民共和国个人信息保护法》解读	曾令平	62
权利义务框架下的移动互联网 APP 个人信息保护 ——《个人信息保护法》对 APP 个人信息保护影响分析	林涛	70

安全的本质在于攻防对抗，安全的发展依托于新技术变革和安全行业演进。安全企业如果要在未来时刻把握成功，就需要对外洞察国家、行业大势，同时苦练内功，厚积薄发。绿盟科技始终以技术为先，在攻防技术、行业前瞻等方面均有独到的积累。

云计算深刻改变着各个领域，如5G、边缘计算、工业互联网等。随着云计算进入下半场，云上对抗日益激烈，云化基础设施的整体安全防护迫在眉睫。万物互联时代，IT、OT、CT技术相互融合，工业互联网、车联网和传统互联网结合后，其安全建设与攻防对抗也值得关注。绿盟科技星云实验室、格物实验室的相关文章不仅分析了前沿趋势，也对相关技术进行了详细阐述，值得一读。

攻防技术的不断演进，使得各类新技术在赋能安全防护方面极具前景与想象空间，如人工智能应用于智能化攻击、知识图谱辅助威胁追踪溯源等。天枢实验室的文章介绍了绿盟科技用人工智能赋能安全的若干实践。

同时，2021年出台的多项数据安全法律法规催生了大量合规性要求和产业需求，我们可以看到，数据安全已经成为继网络安全之后的另一个重要安全赛道。本刊三篇文章从多个维度进行了数据安全、个人隐私相关的法律解读与体系探讨。

在可见的未来，绿盟科技必将把握攻防技术、新行业发展和安全内涵演进三个方向，在重要领域不断加深产品核心能力积累，在创新方向锐意进取，将研究成果孵化为具有竞争力的创新产品和解决方案。

叶晓虎

云原生应用安全防护思考（一）

绿盟科技 创新中心&星云实验室 浦明

摘要：应用是云原生体系中最贴近用户和业务价值的部分。作为云原生应用安全防护系列的第一篇，本文主要针对传统应用安全、API 安全、云原生应用业务安全三方面风险，提出一些防护见解及思考。

关键词：安全防护 云原生应用 API 安全 业务安全 传统应用安全防护

1. 传统应用安全防护

从 50 期《关于云原生应用，这些安全风险了解一下》一文中对传统应用风险的介绍，我们得知传统应用为云原生应用奠定了基石，因而笔者认为云原生应用安全防护也可参照传统应用进行安全防护。接下来笔者将为各位读者介绍传统应用的安全防护方法，主要包含以下四方面。

应用程序代码漏洞缓解

如上一期风险篇中[1]对传统应用安全的分析，应用程序的已知漏洞几乎是造成所有风险的主要原因，因而针对应用程序的漏洞缓解措施非常必要。

应用程序依赖库漏洞防护

应用程序的漏洞缓解措施只能在一定程度上规避开发者不规范编码造成的风险，而应用程序本身除了开发者编写的代码外，还可能需要引入第三方依赖库。那么依赖库是否含有已知漏洞将会直接决定该应用程序是否相对安全，因而针对应用程序依赖库的漏洞防护也非常必要。

应用程序访问控制

“访问权限的错误配置”“脆弱的函数运行”等会导致应用存在未授权访问风险，因而做好应用程序的访问控制非常重要。

应用程序数据安全防护

我们知道，应用程序最终为业务服务，而数据为业务带来价值，从上一期风险篇的分析中我们得知，数据泄露风险是目前应用程序面临的巨大风险之一，如何防止数据泄露是我们需要关心的重要问题。

1.1 应用程序代码漏洞缓解

应用程序代码漏洞缓解应当从安全编码和使用代码审计工具两方面考虑。

1.1.1 安全编码

针对安全编码，开发者需要具备安全编码的能力。例如面对 SQL 注入漏洞，开发者需要将数据和命令语句及查询语句分离，那么最佳的选择便是使用相对安全的 API，而避免使用解释器，提供参数化界面的接口及迁移至 ORM 或实体框架。此外，对参数输

入的有效过滤，例如白名单机制，也有助于防御恶意注入行为。再如针对 XSS 类型的漏洞，主要的防护原则是将不可信的输入源与动态的浏览器内容分离，具体实现的手段也非常多，例如使用从设计上就会将危险输入进行编码或转义以防止 XSS 的 Web 框架，Ruby on Rails 或 ReactJS 等。由于漏洞类型较多，本文受篇幅限制，不再赘述，更多针对代码漏洞的防护方法可以参考 OWASP 组织在 2017 年发布的应用十大风险报告 [2]。

1.1.2 使用代码审计工具

应用程序代码在未部署至服务器前是静态的，我们可以通过手动编写规则脚本进行漏洞筛查，但这样做往往效率较低，可行的方法是使用自动化代码审计工具，业界比较主流的有 AppScan、Fortify、Burp 等。需要注意的是这些工具也不是万能的，可能会有误报或漏报的情况。

1.2 应用程序依赖库漏洞防护

针对应用程序依赖库漏洞造成的风险，我们可以使用受信任的源或软件组成分析技术进行防护。

1.2.1 使用受信任的源

使用受信任的源是最直接的方法，应用开发者可以仅从官方渠道获取第三方组件，同时也可以关注已含有 CVE、NVD 漏洞的第三方组件，避免试错过程，这些含有漏洞的第三方组件可在官方网站上进行查询，例如 Node.js 库 CVE 漏洞列表 [3]、Java 库 CVE 漏洞列表 [4]、Python 库 CVE 漏洞列表 [5]。

1.2.2 使用软件组成分析工具

如果应用程序较为复杂，涉及的组件较多，仅通过手动移除含有漏洞的第三方组件往往效率较低，且容易遗漏漏洞。鉴于此，业界通常采取软件组成分析 (Software Component Analysis，简称 SCA) 技术，其原理是通过对现有应用程序中使用的开源依赖项进行统计，并同时分析依赖项间的关系，最后得出依赖项的开源许可证及详细信息，详细信息具体包括依赖项是否存在安全漏洞、漏洞数量、漏洞严重程度等。最终 SCA 工具会根据这些前提条件判定应用程序是否可以继续运行。目前主流的 SCA 产品有 OWASP Dependency Check[6]、SonaType[7]、Snyk[8]、Bundler Audit[9]，其中 SonaType、Snyk、Bundler Audit 均为开源项目。

1.3 应用程序访问控制

在业务逻辑相对简单的应用中，我们可通过为每个用户赋予不同的权限，实现访问控制。但随着业务量逐渐增大，用户数量不断增多，准确识别每个用户需要或不需要哪些权限是一件具有挑战性的工作，且为每个用户赋予单一权限的方法易造成权限泛滥。因而我们应遵循最小特权原则，即给予每个用户必不可少的特权，从而保证所有用户都能在所赋予的特权之下完成应有的操作，同时也可以限制每个用户所能进行的操作。

使用基于角色的访问控制是实现最小特权原则的经典解决方案，基于角色的访问控制就是将主体（用户、应用）划分为不同的角色，然后为每个角色赋予权限，例如上述提到在业务量大、用户

数多的应用程序中，使用基于角色的访问控制就很有效，因为我们可以定义每类角色所具备的访问权限，这样即便有成千上万个用户，我们只需按照用户的类型去划分角色，从而可能只需要有限个数的用户角色即可完成访问控制。

1.4 应用程序数据安全防护

笔者认为应用程序的数据安全防护应当覆盖安全编码、密钥管理、安全协议三方面。安全编码涉及敏感信息编码，密钥管理涉及密钥的存储与更换，安全协议涉及函数间数据的安全传输。

1.4.1 安全编码

在应用的开发过程中，开发者为了方便调试，常常将一些敏感信息写在日志中。但随着业务需求的不断增多，开发者容易忘记删除调试信息，从而引发敏感信息泄露的风险。更为严重的是这种现象在生产环境中也频频出现，例如 Python 的 oauthlib 依赖库曾被通用缺陷列表 (Common Weakness Enumeration，简称 CWE) 指出含有脆弱性风险 [10]，原因是其日志文件中写入了敏感信息。以下为此依赖库对应含有风险的代码：

```
    if not request.grant_type == 'password':
        raise errors.UnsupportedGrantTypeError(request=request)
        log.debug('Validating username %s and password %s.', request.username,
request.password)
    if not self.request_validator.validate_user(request.username,request.password,
request.client, request):
    raise errors.InvalidGrantError('Invalid credentials given.',request=request)
```

从以上可以看出开发者将用户名密码记录在了 Debug 日志中，这是非常危险的写法，因为不法分子可能会利用此缺陷获取用户的登录方式，并进行未授权访问，甚至窃取用户隐私数据，因而针对应用程序的数据安全，安全编码十分重要。

安全编码具体应该怎么做是读者们关心的问题。笔者认为，最重要的是禁止将敏感信息（如：用户名密码、数据库连接方式）存储至源码、日志及易被不法分子发现的地方，同时我们应对存储的所有敏感数据进行加密。

此外，一些开源项目可以帮助开发者避免敏感信息被硬编码至源码中，例如 AWS 的开源项目 git-secrets[11] 和 Yelp 的开源项目 detect-secrets[12]，各位读者可以参考。

1.4.2 使用密钥管理系统

为了应用程序环境的安全，我们应当使用密钥管理机制，该机制主要用于密钥的创建、分配、更换、删除等操作，目前许多企业采用密钥管理系统 (Key Management System) 的方式，例如国外以 AWS KMS[13]、Azure Key Vault[14]、Google CKM (Cloud Key Management) [15] 等为主，国内则以阿里云密钥管理服务 [16]、腾讯云密钥管理服务 [17] 等为主。

1.4.3 使用安全协议

为避免敏感数据在传输过程中泄露，应确保传输中的数据是加密的，例如 Web 应用中，我们可以通过使用 HTTPS 协议替代 HTTP 协议，确保用户传输的数据不被窃取和篡改，从而在一定程度上避免被中间人入侵。

2.API 安全

通过上一期风险篇对 API 的风险分析，我们知道，虽然云原生应用架构的变化导致了 API 数量的不断增多，但在造成的 API 风险上并无太大区别，因而在相应的 API 防护上，我们可以参考传统的 API 防护方法。此外，我们还可采用 API 脆弱性检测的方式防止更多由于不安全的配置或 API 漏洞造成的种种风险。最后，在云原生应用架构下，我们可使用云原生 API 网关，其与传统的 API 网关有何不同，能为云原生应用风险带来哪些新的防护是我们关心的问题。因此，本节将 API 安全分为传统 API 防护、API 脆弱性检测、云原生 API 网关三个部分进行介绍。

2.1 传统 API 防护

针对传统 API 风险，我们可以使用传统的 API 防护方式，例如针对失效的认证，我们可以采取多因素认证 [18] 的方式或采用账号锁定、验证码机制来防止不法分子对特定用户的暴力破解。再如针对失效的功能授权，我们应当默认拒绝所有访问，并显式授予特定角色访问某一功能。关于更多典型的 API 防护方式，各位读者可以参考 OWASP 组织在 2019 年发布的 API 十大风险报告 [19]，该报告针对每种典型风险均提出了较为详细的防护方法，本文限于篇幅，不再赘述。

2.2 API 脆弱性检测

API 脆弱性主要针对的是服务端可能含有的代码漏洞、错误配置、供应链漏洞等，目前较为可行的方式是使用扫描器对服务端进行周期性的漏洞扫描，国内各大安全厂商均提供扫描器产品，例如绿盟科技远程安全评估系统（RSAS）[20] 和 Web 应用漏洞扫描系统（WVSS）[21]，其中 RSAS 已支持针对容器镜像的扫描。同时，我们也可以使用其它商业版扫描器，例如 AWVS（Acunetix Web Vulnerability Scanner）、AppScan、Burp Suite、Nessus 等。

2.3 云原生 API 网关的功能

云原生 API 网关，顾名思义指云原生应用环境下的 API 网关。笔者认为，云原生 API 网关与传统 API 网关的区别主要有两方面，一方面是应用架构带来的区别，另一方面是部署模式的区别。

针对应用架构带来的区别，传统 API 网关更关注管理 API 带来的挑战，而云原生 API 网关由于应用微服务化后，每个服务都可能会由一个小团队独立开发运维，以快速向客户交付相应的功能，所以为了让每个团队都能独立工作，服务应当具备及时发布、更新及可观测性的特点。鉴于此，云原生 API 网关更关注业务层面，例如可通过为终端用户提供静态地址，并动态地将请求路由至相应的服务地址实现服务发布，又如可在终端用户访问服务过程中通过收集关键可观测性指标实现对服务的监控，再如可支持动态地将终端用户的请求路由至服务的不同版本以便进行金丝雀测试。

针对部署模式的区别，传统的 API 网关通常在虚拟机或 Docker 容器中进行部署，而云原生 API 网关则主要在微服务编排平台部署，典型的如 Kubernetes。

微服务应用环境中，云原生 API 网关充当着非常重要的一环，它不仅要负责外部所有的流量接入，同时还要在网关入口处根据不同类型请求提供流量控制、日志收集、性能分析、速率限制、熔断、重试等细粒度控制行为。云原生 API 网关为云原生应用环境的防护带来了一定优势。首先，由于云原生 API 网关接管南北向流量，所以将外部访问与微服务进行了一定隔离，从而保障了后台微服务的安全。其次，在早期的微服务治理框架中，例如 Spring Cloud，由于其将服务治理逻辑嵌入了具体服务代码中，所以导致了应用的复杂性增加，而云原生网关具备一定的服务治理能力，从而可节省后端服务的开发成本，进而有益于应用层面的扩展。最后，云原生 API 网关也可以解决外界访问带来的一些安全问题，例如 TLS 加密、数据丢失防护、防止跨域访问、认证授权、访问控制。

云原生 API 网关以开源项目居多，近些年来，随着技术的不断发展，Kubernetes 显然已成为容器编排平台的业界标准，因而云原生 API 网关也都相应支持在 Kubernetes 上进行部署，目前主流的云原生 API 网关有 Ambassador、Zuul、Gloo、Kong 等。为了让各位读者一览以上提到的云原生 API 网关在安全功能上的支持，笔者进行了相应统计，以供各位读者参考，具体表 1 所示：

	Ambassador	Zuul	Gloo	Kong
Web应用防火墙	支持	支持	支持	支持
访问控制	支持	支持	支持	支持
基本认证授权	支持	支持	支持	支持
SSL 证书管理	支持	支持	不支持	支持
数据丢失防护	不支持	支持	支持	不支持
跨域（CORS）	支持	支持	支持	支持
JWT	支持	支持	支持	支持
限速服务	支持	支持	支持	支持

表 1 主流开源云原生 API 网关安全功能支持

从表 1 可以看出 Zuul 全项支持，但因 Zuul 与 Spring Cloud 的深度集成，故只能针对使用 Java 环境的微服务进行防护。其余云原生 API 网关均有一项不支持，主要为 Ambassador 对数据丢失防护不支持，Gloo 对 SSL 证书管理不支持，Kong 也是对数据丢失防护不支持。需要注意的是，这三个 API 网关与 Zuul 有较为明显的区别，Ambassador 与 Gloo 均为 Kubernetes 原生网关，且从官方网站上 [22][23] 都能看到它们兼容微服务治理框架 Istio 方案，如果各位读者使用 Istio 治理微服务，可以选择 Ambassador 和 Gloo。在这四个开源项目中，Kong 最为活跃及成熟，从官方的解决方案中 [24] 可以看到，其支持 Kubernetes 部署方案，凭借 Kong 在 API 安全上的积累，相信很快可以在云原生 API 网关上占据一席之地，成为大多数人的选择。

3. 云原生应用业务安全

针对《关于云原生应用，这些安全风险了解一下》一文中提到

的云原生应用业务层面安全问题，基于基线的异常检测是一类比较有效的方法：首先，建立正常业务行为与参数的基线，其次，找出偏移基线的异常业务操作。其中，基线的建立需要结合业务系统的特性和专家知识来共同完成。

在电商系统中，业务参数基线主要基于专家知识来建立。例如商品价格不仅与商品本身相关，也与时间和各类优惠活动等相关。这类基线需要运维人员的持续维护。对于业务逻辑基线的建立，由于业务系统在正式上线运行以后，其操作逻辑一般不会有较大的变化，同时异常操作所占的比例较少，所以可以采集业务系统历史的操作数据，结合统计分析与机器学习的方法建立业务逻辑的基线。相比于人工方法，这种方法可以提高基线建立的效率，有效减轻运维人员的工作量。

为此，可利用分布式追踪工具对云原生应用中产生的数据进行采集，笔者对当前主流的分布式追踪工具 Zipkin、Jaeger、Skywalking、Pinpoint 进行了调研，这些分布式追踪工具大体可分为三类，即基于 SDK、基于探针和基于 Sidecar。

基于 SDK 的分布式追踪工具。以 Jaeger 为例，Jaeger 提供了大量可供追踪使用的 API，通过侵入微服务业务的软件系统，在系统源代码中添加追踪模块实现分布式追踪。此类工具可以最大限度地抓取业务系统中的有效数据，提供足够的可参考指标。但其通用性较差，需要针对每个服务进行重新实现，部署成本较高、工作量较大。

基于探针的分布式追踪工具。以 SkyWalking Java 探针为例，在使用 SkyWalking Java 探针时，需要将探针文件打包到容器镜像中，并在镜像启动程序中添加 -javaagent agent.jar 命令实现探针的启动，以完成 SkyWalking 在微服务业务上的部署。SkyWalking 的 Java 探针实现原理为字节码注入，将需要注入的类文件转换成 byte 数组，通过设置好的拦截器注入到正在运行的程序中。这种探针通过控制 JVM 中类加载器的行为，侵入运行时的环境实现分布式追踪。此类工具无须修改业务系统的源代码，相对 SDK 有更好的通用性，但其可获取的有效数据相对 SDK 类工具较少。

基于 Sidecar 的分布式追踪工具。Sidecar 作为服务代理，为其所管理的容器提供服务发现、流量管理、负载均衡和路由等功能。在流量管理过程中，Sidecar 可以抓取进出容器的网络请求与响应数据，这些数据可以记录该服务所完成的一次单个操作，与追踪中的跨度信息对应，因此可将 Sidecar 视为一种基于数据收集的分布式追踪工具。Sidecar 无须修改业务系统代码，也不会引入额外的系统开销。但由于 Sidecar 所抓取的跨度不包含追踪链路上下文，要将 Sidecar 所抓取的跨度数据串联成追踪链路非常困难。

通过使用以上分布式追踪工具进行数据采集后，针对上一期风险篇提出的三种业务异常场景（业务参数异常、业务逻辑异常、业务频率异常），笔者设计并实现了业务异常检测引擎，如图 1 所示。其中，采集模块主要用于采集业务系统的运行数据，训练模块主要针对业务系统历史数据进行训练以提取行为特征数据，检测模块主要对正在运行的业务系统进行异常检测。

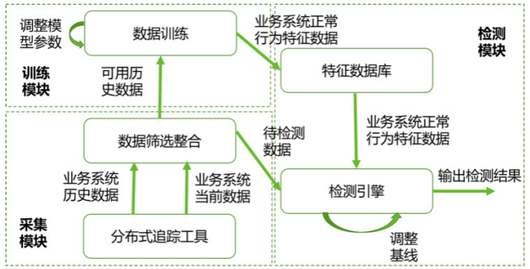


图 1 业务异常检测引擎设计图

检测引擎中每部分的具体功能为：

分布式追踪工具。相比 Skywalking、Sidecar，Jaeger 可获取的数据字段最多，能够检测的异常场景最丰富。然而，Jaeger 需要在业务系统的源代码中进行插桩，对开发团队而言有较强的侵入性。相反，Sidecar 模式没有代码和镜像的侵入性，但通过反向代理截取流量的模式也决定了它不能获得丰富的上下文，如云原生应用的 API 调用关系树（TraceID）是无法获得的。如何利用侵入性更低的采集工具收集到的数据来实现覆盖更多场景的异常检测，仍需要很多后续工作。

数据筛选与整合模块。此模块的主要功能为过滤掉数据集中的脏数据，以及提取出可以表示业务系统行为的数据。在云原生应用中，可以表示业务系统行为的数据为 API 调用关系树、服务名、操作名、HTTP POST 参数等。

数据训练模块。将预处理后的历史数据利用机器学习或统计学的方法，训练出业务系统中的正常行为，并生成与业务系统正常行为匹配的特征数据。这里进行训练的先验知识为，我们认为业务系统中大量存在的行为是正常行为，而数量很少的行为是异常行为。在训练过程中，需要根据专家知识对训练结果的检验来不断调整训练模型的参数。

检测引擎。将业务系统当前数据与特征数据库中的数据进行检索匹配，并利用序列相似性计算等方法，找出特征数据库中与当前行为最为匹配的特征数据。检测引擎需要将特征数据与当前数据的相似性与基线进行比较，若比较结果显示当前行为与正常行为的差异在基线限制范围内，则为正常行为，若超出基线限制范围，则判定为异常行为。对于基线，首先需要根据专家知识设置合理的初始基线，并根据不同场景，或利用无监督模型自行调整基线，或由运维人员手动维护基线。

4. 结语

本文为各位读者介绍了云原生应用在传统应用安全、API 安全、云原生应用业务安全三个维度的相应防护方法，结合之前风险篇的相应介绍，首先，我们可以看出传统应用防护技术适用于云原生应用，所以深刻理解传统应用防护内容非常重要。其次，云原生应用架构的变化为 API 带来了更多特点，也带来了新的防护方法，如云原生 API 网关的合理使用可以有效改善用户环境下的 API 安全状况。最后，云原生应用业务方面的异常会给相应的业务系统带来巨大的损失。而由于 API 业务安全与业务场景的强耦合性，需要在系统设计之初就考虑各种业务

场景下的 API 安全问题。一方面加强 API 的认证授权机制，另一方面要加入必要的数据采集功能，为后续业务异常场景的分析提供支撑。

更多内容可参考由绿盟科技星云实验室编写的《云原生安全：攻防实践与体系构建》一书（随书配套 Github 仓库[25]），本书从容器基础设施、编排系统和微服务等多层面完整地讲解了云原生的风险、攻防和安全架构，干货多多，欢迎大家购买阅读并参与讨论。

参考文献

[1]《关于云原生应用，这些安全风险了解一下》https://book.yunzhan365.com/tkgd/ilcm/mobile/index.html

[2] https://owasp.org/www-project-top-ten/

[3] https://www.npmjs.com/advisories

[4] https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=java

[5] https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=python

[6] https://owasp.org/www-project-dependency-check/

[7] https://www.sonatype.com/

[8] https://snyk.io/

[9] https://github.com/rubysec/bundler-audit

[10] https://cwe.mitre.org/data/definitions/534.html

[11] https://github.com/awslabs/git-secrets

[12] https://github.com/Yelp/detect-secrets

[13] https://aws.amazon.com/cn/kms/

[14] https://azure.microsoft.com/en-us/services/key-vault/

[15] https://cloud.google.com/security-key-management

[16] https://www.aliyun.com/product/kms

[17] https://cloud.tencent.com/product/kms

[18] https://owasp.org/www-project-api-security/

[19]https://zh.wikipedia.org/wiki/%E5%A4%9A%E9%87%8D%E8%A6%81%E7%B4%A0%E9%A9%97%E8%AD%89

[20] https://www.nsfocus.com.cn/html/2019/209_1009/66.html

[21] https://www.nsfocus.com.cn/html/2019/206_0911/8.html

[21] https://www.getambassador.io/user-guide/with-istio/

[23]https://www.solo.io/blog/istio-1-5-api-gateway-with-gloo/

[24] https://konghq.com/solutions/kubernetes-ingress/

[25] https://github.com/Metarget/cloud-native-security-book

如何利用威胁情报报告生成可用威胁子图

绿盟科技 天枢实验室 薛见新

摘要：本文介绍了一种基于威胁情报报告生成入侵子图的方法，该入侵子图通过子图匹配可以直接应用到终端回溯图中来实现入侵检测与回溯。

关键词：安全知识图谱 网络威胁情报 威胁子图

1. 前言

当前企业环境面临的入侵越来越趋于隐蔽，而且具有长期性，为了更好地针对这些入侵进行有效的检测、回溯和响应，企业通常会部署大量的检测设备。安全运营人员需要根据这些检测设备的日志和告警来对入侵事件进行检测与回溯。然而入侵技术的发展通常领先于检测设备检测能力。当新入侵技术或是新漏洞被发现时，通常是以报告的形式公开，针对这些新入侵的检测能力往往很难快速地部署到检测设备中。

网络威胁情报（CTI），通常是技术报告、白皮书、博客和新闻组中报告，是有关网络入侵的宝贵信息来源。这些报告用自然语言描述了入侵的许多方面，包括行动的顺序、对被入侵系统的影响以及破坏指标（IOC）。威胁情报报告中包含子入侵相关的主要知识，可以帮助安全运营人员了解入侵过程并应用于检测与回溯。已有一些研究工作利用 NLP 技术从威胁情报报告中提取入侵行为的相关知识。但该工作现在仍处在起步阶段，还远没能应用到威胁检测上。

主要面临如下挑战：

(1) 在威胁情报报告中，与入侵行为相关的描述可能只占很小一部分。报告中大量的描述信息与入侵行为没有直接关系。

(2) 威胁情报报告中使用的语句结构与日常生活中使用语言具有较大的差异，其中包含了大量的简写、代指和被动语法等。传统的自然语言处理工具难以对 CTI 报告进行处理。

(3) 对威胁情报报告中全局的信息进行提取需要理解入侵行为之间的关系，而理解技术报告中复杂的逻辑是 NLP 领域公认的难题。

本文以文献 Extracting Attack Behavior from Threat Reports[1] 为主要参考来介绍如何基于威胁情报报告提取有效的入侵子图。该文献提出了一个工具 EXTRACTOR，该工具可以精确地、自动地从威胁情报报告中抽取入侵行为。EXTRACTOR 的主要创新性在于其对文本没有强假设，可以从非结构化文本中提取入侵行为回溯图。提取的这些入侵行为回溯图可以应用威胁狩猎。

2. 相关研究内容与技术框架

入侵技术的快速发展对安全防护提出了更高的要求，如何快速地针对新入侵技术生成有效的检测与回溯机制是当前面临的主要挑战。从威胁情报中提取可用于检测与回溯的有效信息是一种可能。但其可行性是能够基于报告提取到可用于威胁检查与回溯的信息，这样才可以第一时间对新入侵进行检测与回溯。

图 1 是 njRAT 恶意样本的报告中的一段关于入侵行为的描述。图 1 右侧表示是根据威胁情报报告构建的回溯图。该回溯图的节点表示与入侵行为相关的实体，如进程、注册表等，表示实体之间的操作行为。njRAT 恶意样本的入侵子图可以与终端日志的回溯图相对应，根据该子图在终端日志回溯图中进行子图匹配来进行入侵检测。

The malware connects to the Command & Control (CnC) server.
The "Authorization.exe" malware has keylogger functionality.
It stores the logged keystrokes in the following file: [CWD]\.tmp
When the "Authorization.exe" malware is executed it :
Creates a copy of itself in the following locations: %APPDATA% %USERNAME%
Tries to open the following file: [CWD]\Authorization.exe.config
Entrenches in the system for persistence in the following registry locations:
HKCU\...\bf7a7ffda58092e10 HKLM\...\bfda58092e10
Beacons to the following C2 node IP: * over TCP port 1177:"217.66.231.245"
Makes the following modification to the registry to bypass the Windows Firewall:
HKLM\...\msnco.exe
The downloaded file is decoded, written to disk as %APPDATA%\...\ccSvcHst
The following files created when the Authorization.exe malware executed: msnco.exe
authorization.EXE-0AD199D6.pf
Msnco.exe and Authorization.EXE-0AD199D6.pf are created by Authorization.exe.

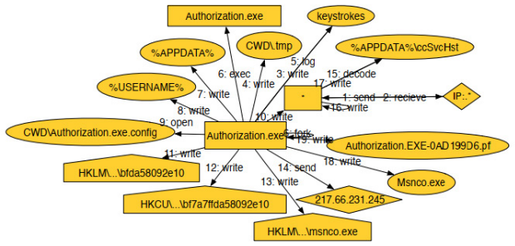


图 1 njRAT

从威胁情报的报告中抽取精确可用的入侵子图依然存在不少挑战。首先，需要在报告中识别与入侵行为相关的文本，因为威胁情报报告通常很长，其中包含了大量的与入侵过程不相关的信息。这需要了解报告中出现的与入侵相关的实体，以及这实体之间的相关入侵行为，据此构建可用于回溯图威胁检测的入侵子图。其次，威胁情报文本的复杂性将影响回溯图的构建性能，这意味着要解决自然语言写作中存在的不同类型的模糊性和复杂性。

下面以 EXTRACTOR 为例介绍利用 NLP 技术从威胁情报报告中提取有效的入侵子图的框架。

图 2 EXTRACTOR 技术框架

EXTRACTOR 通过对威胁情报报告进行多次转换，将其从复杂的、会产生有歧义的形式转换成简单的文本。对简化后的文本进行进一步处理，得到一个可以成功用于威胁检测的回溯图。如图 2 所示，整个过程主要包含四个步骤 :1 标准化 ;2 解析过程 ;3 文本归纳 ;4 回溯图生成。标准化是一个初始的数据预处理过程，把报告中的文本内容转换成规范的形式。文本解析过程是对数据进行消歧。文本概要是删除文本中与入侵行为不严格相关的信息。入侵子图构建是挖掘入侵行为的时序与因果关系，并构建可用于威胁检测的入侵子图。这些步骤需要一些包含与威胁情报语言相关的术语字典来辅助。EXTRACTOR 一共使用了两个词典，一是系统调用同义词典，包含了表示系统调用（如写、读 fork）的动词以及其同义词。这些同义词表示可能是威胁情报报告中使用的表示系统调用的动词。二是报告中的名词词典，该词典包含了报告中常用的名词词典，以及同一概念的不同文本表示。其中系统调用词典包含了 87 个动词，名词词典包含了 1112 个名词短语。

3. 技术细节

为了详细说明具体细节，以图 1 中关于 njRAT 恶意样本的入侵描述为例来进行说明。njRAT 恶意样本 Authorization.exe 连接 C&C 服务器，该恶意样本具有键盘记录功能，把相关的键盘记录写到本地 [CWD]\.tmp 文件中。当恶意样本 Authorization.exe 执行时，会在本地 %APPDATA% %USERNAME% 目录下创建一个副本。同时该木马试图打开文件 [CWD]\Authorization.exe.config，同时修改下面的注册表信息 HKCU\...\bf7a7ffda58092e10 HKLM\...\bfda58092e10 以保证该样本的持续运行。同时修改 HKLM\...\msnco.exe 以绕过防火墙，然后通过端口 1177 跟 c2 “217.66.231.245” 进行通信，下载相关文件并写入磁盘 %APPDATA%\...\ccSvcHst，在 Authorization.exe 运行过程中创建了 msnco.exe 和 authorization.EXE-0AD199D6.pf。

接下来针对该样本的入侵过程分别从标准化、语法与语义解析、文本归纳以及子图构建四部分进行介绍。

3.1 标准化

为了解决威胁情报的文本复杂性挑战，并最大限度地提高入侵子图构建的准确性，首先必须对报告中的句子进行规范化处理。为此，基于标准化方法对报告进行规范化，把复杂的长句转换成简单的短句。直观来说，标准化的目标是把复杂的长句变成多个短句，每一个短句表示入侵主体对入侵客体执行的某个行为。标准化过程包含了三部分，分别是句子边界切割、同质化和句式转换。这些步骤分别执行了句子边界检测、词的同质化和被动词到主动词的转换。

3.1.1 句子边界检测

当前的分词器（如 NLTK[2]）主要依据句子的标点符号来识别句子的边界。而在安全领域，包含多个动作信息的长句或是不规范分隔符的情况很多。针对该问题，除了使用典型的句子分隔符之外，还使用新行、点句、枚举数、标题和头信息作为句子分隔符把长句划分为多个短句。通过长句划分得到的多个词语的短序列如果满足下列条件中的一个则认为这个短序列是一个句子。(1) 该序列是由一个大写的主语开始，包含了构成一个完整句子的所有组成（主语、谓语和宾语），并且该序列之前或是之后的序列也可以构成一个完整的句子；(2) 该序列以系统调动词典中的动词开始，除了主语之外包含了所有组成，同样该序列的之前或是之后的序列可以构成一个完整的句子。

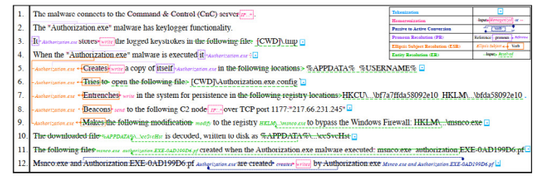


图 3 njRAT 报告的句子边界检测

图 3 给出该过程处理 njRAT 报告的示例。第 4 行到第 9 行是一个长句子，描述恶意样本 Authorization.exe 执行时的相关行为。

首先需要将其按行切分成多个短序列。然后，每个短序列通过词性标注和依存标注打标签，并检测该序列是否满足上面两个条件。可以看到第 4 行满足条件 1，第 5 行到第 9 行满足条件 2。该过程的结果是一系列短句子，这些短句子更有可能描述一个行为。

3.1.2 词的同质化

威胁情报报告经常包含可以产生歧义并影响最终结果质量的结构和同义词。例如，图 1 描述中出现的 C2 和 Command and Control 是同一实体的不同表示。词的同质化是指对同一概念的不同文本表示进行统一。使用两个专门构建的字典对名词短语和动词执行同质化，它们将报告中出现的不同术语以及名词和动词的同义词映射到审计日志中可以观察到的实体和动作。例如，C2、C&C 和 Command and Control 需要被映射成“IP:.*”这样的 IP 地址通配符。以同样的方式，使用系统调用动词在系统调用字典中翻译作为系统调用同义词的动词。词的同质化可以显著地减少报告文本中的异构性，使从报告中提取可行的情报成为可能。

3.1.3 句式转换

威胁情报报告文本标准化的最后一步就是把被动词转换成主动词。这种转换可以更方便地发现系统对象与系统目标，同时能更精确地进行因果推理。

为了进行这种转换，首先需要根据词性标注和依存标记来进行被动句检测。这种类型的句子主要是由依存树中特定的已知模式表示。以如下句子为例“This kind of sentence is predominantly represented by specific and known patterns in DP trees”，在依存树中，is 表示辅助动词或是被动词，deleted 表示动词或是依存树的头，“the downloaded file”是被动句的主语，“by malware”是被动句的谓语。然而在一些情况下，一些代词是隐含在被动句中，而没有明显地出现在句子中。例如，图 3 中第 10 行，代词 malware 没出现在句子中，通过上下文可以知道其是指 njRAT 恶意样本 Authorization.exe。通过该模式，可以检测出被动句式，同时能识别句子中显式或是隐式代词。通过该过程可以把报告中的长句转换成短句，每个短句表示一个行为。

3.2 语法与语义解析

在规范后，需要对文本中相关的引用进行解析。尤其是对文本中一些暗含的引用必须进行明确的识别。

省略主语是一种语言结构，也就是指句子中的主语不存在。省略主语会给子图构建带来挑战，从而导致一些入侵的源节点错误。图 3 中第 5 — 9 行描述的所有动作都是省略主语的例子。针对该挑战，EXTRACTOR 开发了一个 Ellipsis Subject Resolver（ESR）模块。该模块利用词性标注和依存标注以及系统调用的字典。解决这个问题的第一步是检测缺失主语的句子。一旦检测到这种句子，ESR 就会在当前句子之前的句子中出现的实体中建立一个候选主体列表。接下来，该模块根据候选者与缺失主语的句子的距离（以句子数计算），从列表中挑选出最可能的候选者。特别是，距离越近的候选者被选中的概率就越高。例如，在图 3 中，第 5—9 行的句子中缺少主语。ESR 模块检测了前面的句子中的主语和其他对象，它选择了冒号前出现的代词 it 作为主语。

代词解析是指代词被映射和替换到它们所指的前述实体的过程。在没有 PR 的情况下处理文档（构建出处图）会导致一个实体出现多个节点（即代词）。为了解决代词问题，EXTRACTOR 采用了一个流行的核心词解析模型——NeuralCoref[3]。这个模型在解决威胁情报报告领域的代词方面效果最好。

隐喻是指用一个词或代词来指代句子中以前使用过的另一个词或短语，以避免重复。在解析步骤完成后，文本由具有明确主语、宾语和动词的句子组成。ESR 模块也在一定程度上减少了文本的数量。然而，主要的文本缩减步骤是在“解析”之后执行的，接下来将介绍。

3.3 文本概要

为了减少多余信息，获得可直接用于检测入侵行为的简明描述，需要删除大量与入侵行为不相关的描述。针对冗长的报告，需要识别哪些句子描述了入侵行为。另外，需要简化句子的描述。在每个句子里，通常会出现修饰性词语，如副词和形容词，这些词语对构建入侵行为子图的帮助并不大，需要删除。针对该需求 EXTRACTOR 设计了一个两步法。其流程如图 4 所示，主要包括一个 BERT[4] 分类器和一个 BiLSTM 网络 [5]。

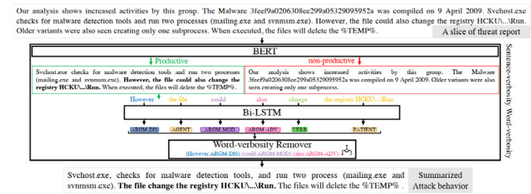


图 4 文本概要模块示例

为了区分实际威胁行为的句子和普通行为的句子，需要超越主题分类，对文本有更深入的理解。因此，为了对这些联系进行分类，模型必须建立一个关于单词上下文的细粒度表示法。目前，建立这种细粒度表示法的最佳模型之一是 BERT。

文本概要的第二步是从 BERT 得到的句子中删除修饰性词语。它由两个阶段组成，一个阶段是得出句子成分语义的 BiLSTM 网络，另一个阶段是去词。

在一个句子被 BiLSTM 网络处理后，其成分被标记为 Agent、Patient 和 Action，以及其他类型的参数。在下一个阶段，不必要的句子成分被删除。从理论上讲，这只能通过保留句子中的 Agent、Action 和 Patient 成分来实现。然而，在某些情况下，这种方法会删除重要的信息。

3.4 入侵子图构建

经过前面的步骤，得到的文本是这样一种形式：系统主语（如进程）、对象（如文件、套接字）和动作（如执行）是明确的、有序的，而且大部分多余的信息均已被删除。

这一阶段的目标是根据处理好的文本生成一个有效的入侵子图。该步骤主要是基于文本识别语义实体与关系。实体的语义识别依据语义角色标签（SRL），关系与信息流方法通过因果关系挖掘方法实现。

SRL 是一种发现句子中语义角色的技术。图 5 中的两句子是对上文报告恶意样本的两个入侵行为描述。一个是主动形式的，一个是被动形式的。SRL 能够从每个句子中提取两个角色（用 Raw SRL 表示），并理解哪个名词是目标者（也就是动作落在上面的人，用 ARG1 表示），哪个是代理人（携带动作的名词，用 ARG0 表示）。一个 SRL 角色可以被认为是一个动作。因此，SRL 能够正确地将句子中的每个成分与语义标签联系起来。

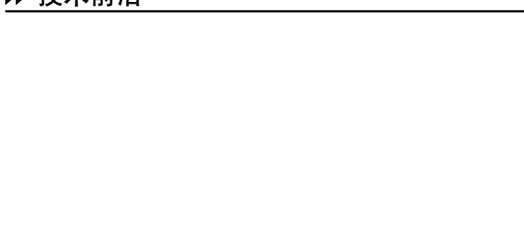


图 5 SRL 标注示例

然后要根据 SRL 的输出构建入侵子图。首先，将具有相同文本的 SRL 合并到同一个节点中，并剔除不属于系统实体的词。接下来，使用以下方法构建图：（1）源节点－边－目标节点三元组。对于每个句子，如果它至少有三个角色，包括一个动词角色（作为连接器的系统调用表示）和两个实体，据此生成一个三元组。（2）边的方向依据系统调用与系统数据流动方向来确定。

4. 结语

针对检测设备检测能力的滞后性，如果能够从威胁情报中自动提取相关入侵子图直接应用到终端日志回溯图中，则可以大大提高检测的时效性。EXTRACTOR 提供了一套有效可行的技术框架。但是由于 NLP 技术的局限，在实际应用过程中依然存在不少的挑战；在实验验证过程中，该方法针对恶意样本效果较好，针对其他复杂多步入侵的效果依然有待提高。

参考文献

[1] Satvat K，Gjomemo R，Venkatakrishnan V N . EXTRACTOR: Extracting Attack Behavior from Threat Reports[J]. 2021.

[2] https://www.nltk.org/

[3] https://github.com/huggingface/neuralcoref.

[4] Devlin J，Chang M W，Lee K，et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[5] Huang Z，Wei X，Kai Y . Bidirectional LSTM-CRF Models for Sequence Tagging[J]. Computer Science, 2015.

基于规则向量化的HTTP资产识别方法探索

绿盟科技 合规安全技术部 张卓 张迎苹

摘要：在资产探测识别中，基于应用层协议的报文信息，利用知识规则，按照特定方式对报文信息进行规则匹配来获取资产信息，是目前资产探测识别的主要手段。而基于 HTTP 协议层报文信息规则匹配的方法，是目前云计算、物联网、移动互联等场景下进行资产识别比较基础和常用的一种资产识别技术。

关键词：HTTP 资产探测识别 规则向量化

1. 背景简介

从整体上来看，基于 HTTP 协议报文进行资产识别，主要是基于特定 HTTP 请求的响应报文所进行的规则匹配。从所关注匹配的响应报文的侧重不同来看，基于 HTTP 协议的资产识别方法可以进一步被分为基于头部字段顺序差异与语法差异的识别方法、基于服务标识（Banner）的识别方法，以及基于处理方式差异的识别方法等 [1]。

然而，无论匹配处理的对象是哪种响应报文或响应报文的哪一部分，基于特定知识规则匹配的资产识别方式均存在着诸如缺少弹性、关注信息局限以及抗干扰能力差等问题。比较典型的即人为信息修改 / 干扰所造成的匹配规则失效问题。例如基于 HTTP Banner 信息进行资产识别方法，其主要通过对 Banner 中提取获得的 Server、User-Agent、Authorization 等 Response 头部字段进行匹配，来识别相应的资产信息。而在实际场景中，Banner 信息很容易被人为修改、模糊甚至伪装，这就使得相应的资产识别规则失效，甚至可能会被引导产生特定的误报结果 [2]。

本文基于 HTTP 响应报文的 Banner 信息，首先，对业务中单纯基于规则匹配进行资产识别的方法中所存在的问题进行总结和讨论，结合具体实例对一些典型问题进行分析；其次，在充分分析问题的基础上，结合 NLP 深度学习方法，基于分布式词向量模型，对 HTTP Banner 响应头信息以及响应体进行特异性向量化，并在规则匹配的基础上，基于 SVD 矩阵分解方法，探索一种基于匹配规则和文本向量化技术相结合的规则向量化技术；最后，结合实际业务，基于规则向量的相似度计算模型，在规则匹配无法生效的场景下，验证规则向量资产的识别和推断能力。

2. 资产规则匹配及其主要问题

在当前业务中，基于应用层协议的资产识别技术，一般都是基于匹配规则进行的。这里的匹配规则主要是指基于字符串的正则表达式匹配和关键词匹配。由业务驱动形成匹配规则的流程一般如表 1 所示。

步骤	流程	描述
Step 1	待识别资产确定	一般是在业务中提出需要识别哪种资产，直接体现即一个资产字段或者产品名称。
Step 2	确定应用层协议范围	一般依据业务场景的不同，所选择的应用层协议也有所不同，如摄像头的RSTP协议，一般情况下使用较多的是HTTP/HTTPS协议。
Step 3	收集协议报文样本	一般在条件允许的情况下，需要搭建或寻找目标资产的环境，通过构造不同的请求报文来触发目标资产的响应信息，收集响应信息。或者，可以基于诸如NTI平台的累积数据，直接拉取收集相关信息。
Step 4	分析协议报文样本	分析收集到的报文样本，结合待识别目标，寻找其特征信息，一般情况下都是寻找特征字段。
Step 5	形成目标资产匹配规则	基于上一步发现的特征信息，抽取关键特征，形成匹配规则；在特征明显时，一条规则就可以覆盖当前的报文样本，而在其他情况下，可能形成多条规则。
Step 6	匹配规则验证	一般分为两个场景的验证，一是在当前报文样本上的验证，一是实际业务中的反馈验证。

表 1 由业务驱动的资产匹配规则形成

一般情况下，经过上述流程所形成的匹配规则，都能在特定业务环境中取得不错的识别效果。

但仔细分析该过程可以发现，上述流程所形成的匹配规则主要存在以下三个问题：

(1) 匹配规则对某些外部干扰、变动十分敏感；

(2) 匹配规则只关注到报文当中资产信息的局部特征；

(3) 匹配规则的质量无法保证，极易受报文样本的质量限制。

上述三个问题之间互有联系，也相互影响，但侧重稍有不同。下面将结合具体业务数据，对三个问题进行进一步分析。

2.1 匹配规则对目标的某些变动干扰十分敏感

通常情况下，一条规则只能覆盖目标资产的部分样本实例，当响应报文中有关资产信息模式的某些字符发生轻微变动时，该规则即无法完成对相应资产的匹配识别。这是硬匹配识别的主要特征，与匹配规则构成存在强依赖关系。

以识别某国产 WEB CMS 框架为例，几乎在获得的所有报文样本中，其框架信息都由关键字段构成，其形式如下图 1 所示。

```
<br />
Powered by <strong>PHPCMS</strong> <em>V9.6.3</em> &copy; 2011 <img src="?statics/images/copyright.gif"/></p>
</div>
```

图 1 某国产 WEB CMS 框架的关键字段

这样我们可以按照匹配规则形成流程，基于关键特征信息，形成匹配规则，如下表 2 所示。

资产	某国产WEB CMS 框架
匹配规则	"""Powered by PHPCMS (.*?)"""

表 2 基于报文样本所形成的某国产 WEB CMS 框架匹配规则

然而，在实际业务测试中，我们发现一些该资产的新报文样本，其特征类型并未发生改变，特征字段出现的位置也未发生改变，但关键信息构成却发生了轻微变动，这导致了上述规则的识别失效。轻微变动的特征字段如图 2 所示。

```
<a href="http://www.[redacted].com/index.php?m=link" target="_blank">友情链接</a>
<br /> Powered by <strong><a href="http://www.phpcms.cn" target="_blank">PHPCMS</a></strong> &copy; 2015<img src="ht
tp://www.[redacted].com/statics/images/copyright.gif"/>[redacted]
```

图 2 某国产 WEB CMS 框架的轻微变动特征

原因在于我们无法穷举目标资产的所有具体实例。同时，应用层协议报文具有的灵活性，可以毫无约束地对相关字符进行人为修改，这就导致了基于匹配规则进行的资产识别无法覆盖所有实例场景，而基于部分样本所抽取形成的匹配规则也极易被外部干扰，进而导致识别失效。

2.2 匹配规则只关注到报文中资产信息的局部特征

匹配规则极易失效，也是因为匹配规则往往只能关注到报文中资产信息的局部特征。一般情况下，基于相应资产样本报文所抽取的匹配规则，只是特征抽取所有样本报文共同的显式信息，这些特征往往是报文信息中的局部信息，无法反映出报文的全局信息以及其所蕴含的隐藏信息，诸如结构等。

这种用来形成匹配规则的显式局部信息很容易被篡改、掩盖，这也将导致匹配规则对相应资产识别的失效。

我们以 Nginx WEB 服务器框架的识别为例，在 301 状态码下，基本上以 Server 响应头取值为 Nginx 的报文样本，都具有如下格式。

```
HTTP/1.1 301 Moved Permanently
Date: Thu, 03 Sep 2020 00:05:57 GMT
Content-Type: text/html
Content-Length: 185
Connection: keep-alive
Server: nginx/1.14.0
Location: [redacted].com/

<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx/1.14.0</center>
</body>
</html>
```

图 3 301 状态码下 Nginx 框架代表样本

显然，按照匹配规则形成以及 HTTP 协议响应头的含义理解，我们可以使用 Server 字段进行关键词匹配。但在这种情况下，图 4 也是 Nginx 框架的报文样本，进行规则匹配是失效的。

```
HTTP/1.1 301 Moved Permanently
Date: Sat, 05 Sep 2020 10:41:40 GMT
Content-Type: text/html
Content-Length: 162
Connection: keep-alive
Location: [redacted]/schempion
Server: JeffBezosDickRocket

<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

图 4 其他隐式含有 Nginx 框架的报文样本

当然，这里使用 HTML 响应报文体中的 Nginx 字段同样可以匹配覆盖两种报文。然而，我们想要说明的是，除了局部特征 Server 字段以外，我们还可以从整体报文包含的结构信息来提取获得其属于 Nginx 框架的全局特征信息，例如我们可以从响应头顺序找到其一致性，进而脱离 Server 字段取值对资产判断的约束，避免人为信息模糊和修改所产生的误报问题。进一步说，使用基于 RNN 的序列建模技术，向量化表征报文的全局结构和上下文关联信息，可以更好地判断两者的相似性，提高资产识别的抗干扰能力。

2.3 匹配规则的质量极易受报文样本质量的限制

在理想情况下，所有的报文样本具有共同的特征，一条匹配规则即可以覆盖所有的报文样本，完成相应资产的识别。在极端情况下，所有的样本报文可能都不具有共同的可识别捕捉到的特征信

息，这样每条样本可能都需要一条规则来匹配，这种情况下资产识别的结果置信度将会很低，容易产生相应资产识别的误报。

我们仍以某国产 WEB CMS 框架的识别为例，在收集到的报文样本中，很难找到相对统一的资产规则特征。而在获取的特征中，我们在进一步收集信息后，发现这些特征并非该资产独有，容易造成误报。因此针对该框架的识别，在花费大量成本收集报文样本后，构建了多条规则，且每条规则的置信度都无法保证。

图 5 某国产 WEB CMS 框架的报文样本示例 1

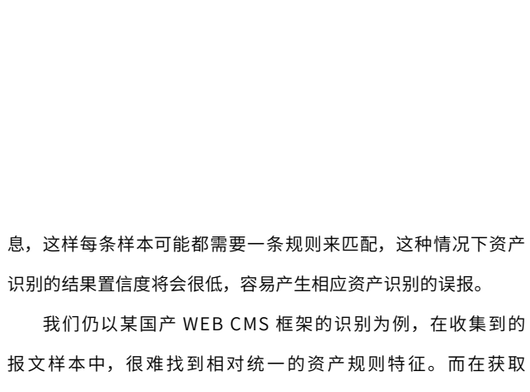


图 6 某国产 WEB CMS 框架的报文样本示例 2

3. 基于文本向量化技术的规则向量化方法探索

在上一节，我们针对资产匹配规则形成流程，结合实际业务场景，讨论了基于匹配规则进行资产识别方法中存在的主要问题。在本节，我们基于分布式词向量训练模型与 SVD 方法，结合响应报文 Header 的特征向量，探索提出一种规则向量化与资产刻画向量化的算法构建流程，对其在实际业务中的表现进行讨论。

3.1 算法概述

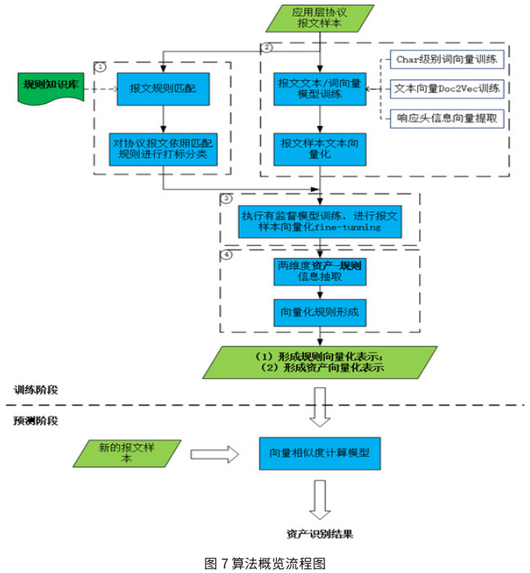


图 7 算法概览流程图

图 7 为规则向量化算法的概览流程图，详细的算法计算流程表 3 所示。

输入：

（1）**HTTP** 响应报文样本 $X=(\mathrm{x}^{(1)},\mathrm{x}^{(2)},\mathrm{x}^{(3)},.....,\mathrm{x}^{(m)})$；

（2）规则知识库，将其表示为 $Pattern=\{(pat_k,asset_k)\}_{k=1}^{Anum}$，其中 pat_k 表示第 **k** 条资产识别规则；$asset_k \in Asset=\{asset\_label_k\}_{k=1}^{Anum}$，$asset_k$ 表示第 **k** 条资产识别规则对应的网络资产，P_{num} 表示资产识别规则的总条数，$Asset$ 表示网络资产集合，$Anum$ 表示网络资产的总种数；

（3）指定词（字符）向量维度 Dim_1 与文本向量维度 Dim_2。

输出：

（1）资产的向量化表示（即网络资产的规则特征）为：$\{Rule_a\}_{a=1}^{Anum}$；

（2）匹配规则的向量化表示 $\{P_k\}_{k=1}^{Pnum}$。

其中，$Rule_a$ 表示第 **a** 种网络资产的向量化表示，P_k 表示第 **k** 条资产识别规则的向量化表示。

1、调用资产匹配规则对所有报文样本进行匹配，依据匹配结果对样本进行标注分类：对于样本 $\{x^i\}_{i=1}^m$，标注形成 $\{(x^{(i)},pattern^{(i)},asset^{(i)})\}_{i=1}^m$；

其中：$x^{(i)}$ 表示第 i 个响应报文样本，$pat^{(i)}$ 表示第 i 个响应报文样本对应的资产识别规则，$asset^{(i)}$ 表示第 i 个响应报文样本对应的网络资产。

2、按照以下步骤，对所有响应报文样本进行特征提取；

2.1、提取报文头信息：针对每个响应报文样本，可原样保留大小写信息，获得规模为 H 的报头词表，抽取报头词表形成报文头的特征向量，则对于 $x^{(i)}$，有 $X_{Char}^{(i)}=(h_1^{(i)},h_2^{(i)},.....,h_H^{(i)})$，$h_t^{(i)}$ 表示报头词表中第 t 个报头字段在第 i 个响应报文样本 $x^{(i)}$ 中的位置信息，$t=1,2,....,H$。

2.2、分别提取词 **Word** 级别和字符 **Char** 级别的字符特征：以 $\{Word_w^{word}\}_{w=1}^{vocab\_size}$ 表示所有响应报文样本包含的词所形成的词表的词向量，其中，$vocab\_size$ 表示词表中词的总数量，$Word_w^{word}$ 表示词表中第 w 个词的预设词向量，即词表中第 w 个词 **Word** 级别的词向量。对于第 **i** 个响应报文 $x^{(i)}$ 而言，该响应报文 **Word** 级别的向量化表示（即 **Word** 级别的字符特征）$X_{word}^{(i)}$ 为：

$$X_{word}^{(i)}=\sum_{word_w^{word}\in x^{(i)}} Word_w^{word}$$

上述公式表示对该响应报文中各词 **Word** 级别的词向量中对应相同位置上的元素进行相加处理，得到该响应报文 **Word** 级别的向量化表示。

以 $\{\mathrm{Char}_c\}_{c=1}^{vocab\_size}$ 表示所有响应报文样本包含的字符所形成的字符表的字符向量，其中，char_size 表示字符表中字符的总数量，$Char_c$ 表示字符表中第 c 个字符的字符向量。则针对词表中的第 w 个词，可从 $\{\mathrm{Char}_c\}_{c=1}^{Wor\_dlv}$ 中查找组成该词的各字符的字符向量，对各字符的字符向量进行融合处理，得到第 w 个词的目标词向量 $Word_w^{Wor}$，即第 w 个词的目标词向量 $Word_w^{Wor}$，即第 w 个词 Char 级别的词向量。

对于第 i 个响应报文 $x^{(i)}$ 而言，该响应报文 Char 级别的向量化表示（即 Char 级别的字符特征）$X_{Wor}^{(i)}$ 为：

$$X_{Wor}^{(i)}=\sum_{Word_w^{Wor}\in x^{(i)}} Word_w^{Wor}$$

上述公式表示对该响应报文中各词 Char 级别的词向量中对应相同位置上的元素进行相加处理，得到该响应报文 Char 级别的向量化表示。

2.3、提取报文体特征向量：基于 Doc2Vec 嵌入，获得所有响应报文样本的报文体特征向量 $\{D_{doc}^i\}_{i=1}^m$，其中 D_{doc}^i 表示第 i 个响应报文样本的报文体特征向量；

3、基于有监督网络，综合上述特征进行规则分析，获得统一的样本报文向量：以各响应报文样本的报头特征向量、报文体特征向量、Word 级别的字符特征、以及 Char 级别的字符特征作为有监督模型训练的预训练嵌入，以拟合各响应报文样本实际对应的资产识别规则为训练目标，重新训练获得统一的文本向量 $\{Doc^{(i)}\}_{i=1}^m$；

4、基于规则标签形成特征矩阵，对资产识别规则进行向量化表示：针对第 k 条资产识别规则，可以 dim_2 为行、以该条资产识别规则对应的每个响应报文样本的 Doc 为一列，构成规则矩阵 S，对 S 进行奇异值分解处理，再基于分解结果对 S 进行主维度变换，从而获得该条资产识别规则的规则特征 P_k，k=1，2，… P_{num}；

5、基于同一资产下的不同匹配规则，依照公式加权计算资产向量：

$$Rule_a=\sum_{w=1}^{W_a} w_r^a P_r^a$$

$$w_r^a=\frac{m_r^a}{\sum_{r=1}^{W_a} m_r^a}$$

其中，R_a 为第 a 种网络资产对应的资产识别规则的条数，w_r^a 为第 a 种网络资产对应的第 r 条资产识别规则的权重，P_r^a 为第 a 种网络资产对应的第 r 条资产识别规则的规则特征，m_r^a 为第 a 种网络资产对应的第 r 条资产识别规则所匹配的响应报文样本个数。

表 3 规则向量化算法

对于未知报文的匹配识别，识别阶段的算法规则如下：

（1）将获取的待识别网络资产的响应报文输入到训练好的规则分析模型中进行规则分析，得到响应报文的规则特征，对响应报文的规则特征与各条资产识别规则的规则特征进行相似度计算；

（2） 依据如下规则进行资产判别：

- 若存在相似度高于预设阈值的第一资产识别规则，则依据如下规则进行判定：

$$asset\_label=\arg\max_{\{P_k\}_{k=1}^{Pnum},\ asset_k\in Asset}\left\{Similarity\left(x^{new},P_k\right)\right\}$$

假设预设阈值为 95%，该阈值直接影响的是判别结果的置信度；

- 若不存在相似度高于预设阈值的第一资产识别规则，从各资产识别规则中选择与响应报文对应的规则特征之间的相似度最高的N条第二资产识别规则，将这N条第二资产识别规则对应的网络资产作为第一资产集合，并确定响应报文对应的规则特征与建立的各网络资产的规则特征之间的相似度，将与响应报文对应的规则特征之间相似度最高的M个网络资产作为第二资产集合，然后，从第一资产集合和第二资产集合中，确定待识别网络资产的资产识别结果。

在上述算法流程中，核心是第 3 步与第 4 步。通过第 3 步，基于两种词向量（word 级别与 char 级别）和两个文本级别的特征向量，通过对规则匹配标签的拟合训练，获得了有监督训练条件下统一表达的文本向量；通过第 4 步对同一规则下的样本向量进行矩阵分解，提取所有样本的共有信息来形成规则的向量化表达。下面我们对这两个计算过程进行简单介绍。

3.2 基于多种分布式词向量方法的文本向量化

首先需要说明的是，为什么要引入 word 级别和 char 级别两种词向量来表征文本向量。主要原因是 word 级别的词向

量采用 Word2Vec 进行训练，能够很好地表征词的语义信息，对文本的上下文特征具有很好的刻画效果。而采用 char 级别的字符向量则主要是为了解决 Word2Vec 对于未登录词的向量化表示，能够进一步完善文本在结构上的信息表达。最后，将两种词向量进行综合，获得文本整体的向量表达。

其次，采用 Doc2Vec 对文本进行直接的文本向量刻画，是为了从整体上刻画文本的构成，解决词向量训练方式只能反映词的局部上下文构成特征的问题。

由于许多重要信息都主要包含在响应报文的 Header 中，所以这里额外引入了响应头的顺序特征向量，来抽象表达响应头的构成特点。

最后，将这 4 种向量作为预训练嵌入层引入到如下训练模型中，通过相应的参数映射获得文本向量的统一表达。

模型训练的结构如图 8 所示。

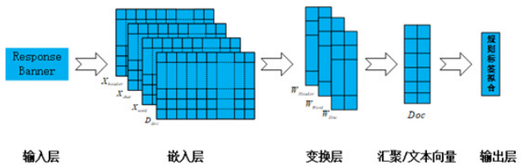


图 8 统一文本向量训练简图

变换层线性映射公式如下：

$$Doc_{header}^{(i)} = X_{header}^{(i)} W_H$$

$$Doc_{word}^{(i)} = \left(X_{word}^{(i)} + X_{char}^{(i)} \right) W_{Word}$$

$$Doc_{\mathrm{doc}}^{(i)} = D_{doc}^{(i)} W_{Doc}$$

最终的文本向量表达为：

$$Doc^{(i)} = Doc_{doc}^{(i)} + Doc_{word}^{(i)} + Doc_{header}^{(i)}$$

3.3 基于 SVD 变换的规则向量化

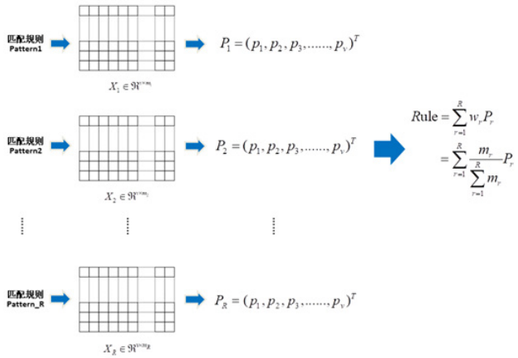


图 9 规则向量与资产向量计算

在经过监督迭代之后，依据嵌入层微调参数与变换层参数，我们可以得到所有报文样本的文本向量。如图 9 所示，同一资产下可能对应的多个匹配规则包含 / 覆盖了多个样本，我们以样本作为信息描述的维度，基于如下公式对规则样本矩阵进行分解。

$$X = U \sum V^T$$

$$X \in \Re^{D \times D}, U \in \Re^{D \times D}, V \in \Re^{M \times M}$$

取 V 矩阵的第一特征向量 $v \in R^{\times 1}$，则可以获得每个规则的向量化表达为：

$$P_{pattern} = X_{pattern} v_1$$

最后，再按照图 9 计算方式，即可获得资产信息的向量化表示。

3.4 实验举例

图 10 为实际业务中获得的 Banner 信息，按照已有的匹配规则对其进行匹配识别时，并未匹配到目标资产。响应头中的 Server 字段代表的资产含义也并未存在于目前的资产知识库当中，因此常规情况下我们无法对其进行识别推测。

```
HTTP/1.1 302 Found
Date: Tue, 06 Jul 2021 17:09:20 GMT
Server: SUOS
Set-Cookie: sugap6_session=Bz4OP1d1AzcFeVV3ATHPO10%2FUDQLKVFyB2dQJVYlBmwPawM%2BXlMAPVdrAC1cYFEsVWhRMwgyBTpcLFQ
7DzYLOFH3UWZQYFc3CTIAMANhB2oHOw49VzMDbAVjVTQBbQ87XT9QQQs9UTcHM1AyVmMGNw9lA2BeagA2VzkAKVxgUSxVaFExCDAFO1wsVGMPf
wsEUzdRM1A0VyY2JwAgAyIHcQdkDnZXagM0BTFVPgErDztdP1AgCziRNAcsUGdWbwY8DyoDYV44AGJXLQAwXChRZVVjUTAIOgUlXHtUeQ9qCyl
TCVE3UDdXMQ1sAAcDcwdoBzROP1dgAzYFcVUN2BATgPOV01UCwLZ1FgB3FQI1Y2BmQPZwM%XmUAIldmAH9cIlF9VUJRaQg6BTFcOFQAD2oLblN
vUWZQY1diCTIAaAMiB2kHZg4%2FV3IDeAVxVWEBew9VXWtQbwt%2FUWAHIFBtVlUGPw80A2peLg8ZVzQAdg%3D%3D; path=/; secure
Set-Cookie: sugap6_session=a%3A0%3A%7B%7D; expires=Tue, 07-Jul-2020 03:09:20 GMT; path=/
Location: http://localhost/login/mainbody
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1
Content-Length: 0
Connection: close
Content-Type: text/html
```

图 10 实际数据

利用上述模型对样本进行向量化之后，与规则向量进行计算，在 83% 的置信条件下，我们可以推测出目标主机的 WEB 服务器框架类型，结果如图 11 所示。

```
相似度0.831352174282074，对应服务： ['lighttpd']
相似度0.8255043029785156，对应服务： ['lighttpd']
相似度0.8239476084709167，对应服务： ['lighttpd']
相似度0.8225648999214172，对应服务： ['lighttpd']
相似度0.822230339050293，对应服务： ['lighttpd']
相似度0.8196273446083069，对应服务： ['lighttpd']
相似度0.8167386651039124，对应服务： ['lighttpd']
相似度0.8159902095794678，对应服务： ['lighttpd']
相似度0.8155393004417419，对应服务： ['lighttpd']
相似度0.8154323101043701，对应服务： ['lighttpd']
相似度0.8148780465126038，对应服务： ['lighttpd']
相似度0.8147661089897156，对应服务： ['lighttpd']
相似度0.8145559430122375，对应服务： ['lighttpd']
相似度0.8138542771339417，对应服务： ['lighttpd']
```

图 11 模型计算结果

4. 结语

本文结合实际工作中基于匹配规则的资产识别经验，对其中的一些问题进行了讨论，提出了我们对相关问题的分析与认识，并基于此，结合 NLP 文本处理与 SVD 矩阵分解等技术，探索了一种基于规则向量化的资产识别方法。结合实际的业务数据，在一定程度上验证了该方法的可行性。然而，在展开实际工程化计算时，我们依旧面临着以下一些问题：

（1）可被规则匹配并标注的数据依旧太少，很难支撑其深度网络参数训练的数据规模，这导致了浅层迭代获得的参数稳定性相对较差，参数并未收敛到稳定点；

（2）模型对有些规则的向量化刻画效果比较好，可以对自身的规则样本进行很好的召回（置信度 95% 以上的相似度计算）。然而还有一些规则的向量化效果比较差，向量中蕴含的规则信息噪声较多，对规则样本的识别召回效果较差；

（3）我们基于的训练数据全部来自于 NTI 平台，这就存在一个很大的问题——对于没有明显特征的 Banner 样本，即便模型依据相似度计算推测出目标的资产类型，我们也很难去实际验证；

（4）由于同一端口暴露的资产信息具有多样性，模型的判断结果很容易由出现最多、最常见的资产类型主导，进而会忽略其他资产类型的识别判断。

参考文献

[1] 赵建军 . 网络空间终端设备识别技术研究 [D]. 兰州理工大学 , 2016.

[2] 曹来成 , 赵建军 , 崔翔 , 等 . 基于余弦测度下 K-means 的网络空间终端设备识别 [J]. 中国科学院大学学报 , 2016, 33(004):562 － 569.

关于构建数据安全生态圈的研究与实践

绿盟科技 咨询设计部 曾令平

摘要：2021 年 3 月，《中华人民共和国国民经济和社会发展第十四个五年规划和 2035 年远景目标纲要》正式发布，其中提到“坚持放管并重，促进发展与规范管理相统一，构建数字规则体系，营造开放、健康、安全的数字生态”。“数字生态”的提出为构建数据安全生态圈的研究和实践提供了有力支撑。在国家数据安全战略的指导下，正逐渐形成全社会共同参与、共同维护数据安全和促进发展的良好环境。本次研究提出数据安全生态圈的整体框架包括一个中心、两个循环、三个体系、五个关键和八大路线，以数字生态为建设目标，从组织如何落实数据安全治理与建设要求的角度出发，确定数据安全生态圈的具体内容以及各个指标，最终形成数据安全各层级的落地执行路径。

关键词：数据安全 生态圈 个人信息保护 数据安全合规 数据安全治理 数据安全考核指标

Abstract：In March 2021, “the 14th five year plan for national economic and social development of the People’s Republic of China and the outline of long-term objectives for 2035” were officially released, which mentioned that “we should pay equal attention to open and control, promote the unity of development and standardized management, build a digital rule system, and create an open, healthy and safe digital ecology”. The proposal of “digital ecology” provides strong support for the research and practice of constructing a data security ecosystem. Under the guidance of the national data security strategy, a good environment that the whole society participates, maintains data security and promotes development is gradually formed. This study proposes the overall framework of the data security ecosystem, including one center, two cycles, three systems, five keys and eight routes. Taking the digital ecology as the construction goal, the specific contents and indicators of the data security ecosystem are determined from the perspective of how to implement the data security governance and construction requirements. Finally, a landing execution path at all levels of data security is formed.

Key words：data security; ecosystem; personal information protection; data security compliance; data security governance; data security KPI

数字经济的发展离不开对数据安全的保障，当前发展过程中显露出的数据安全问题阻碍着数字经济的发展。此外，一些网络安全问题聚化为数据安全问题，如跨境数据流动、数据泄露等。Risk Based Security 公司的数据显示，2020 年 3 个季度数据泄露的总数达到 360 亿条 [1]。与此同时，数据垄断、数据滥用、数据权属、数据流通等新型问题也进入研究和管理视野 [2]。

美国、欧盟、中国等陆续出台的重要政策和执法措施越来越聚焦“数据安全”和“隐私保护”。欧盟发布了《通用数据保护条例》（General Data Protection Regulation，GDPR）[3]，美国加利福尼亚州公布了《加利福尼亚州消费者隐私法案》（California Consumer Privacy Act，CCPA）[4]。中国则颁布了《中华人民共和国网络安全法》（以下简称《网络安全法》）《中华人民共和国数据安全法》（以下简称《数据安全法》）和《中华人民共和国个人信息保护法》（以下简称《个人信息保护法》）。这 3 部法律在立法定位上坚持总体国家安全观，共同构成了我国数据新秩序下的“三驾马车”。与此同时，截止到 2021 年 8 月，全国信息安全标准化技术委员会（TC260）围绕数据安全和个人信息保护发布 9 项、在研 22 项国家标准 [5]。这一系列操作对数据商业化利用与公民个人信息保护之间的关系进行了规则约束，保护了数据和公民个人信息安全，维护了国家安全和公民合法权益，对数字经济发展起到了极大的推动作用。

随着国家政策、法律法规逐渐完善，传统安全企业布局正悄然发生转变，互联网头部企业开始布局网络安全领域，网络安全厂商、电信运营商、设备厂商以及一些新生的独角兽企业都不同程度聚焦在 5G 安全、数据安全、安全合规等前沿热点领域和方向。在“十四五”规划的征求意见稿中，“数据”一词出现了 60 余次。随着“新基建”的不断发展以及数据开放的齿轮转速不断加快，在合规的前提下，数据业务所带来的价值将是巨大的。智慧城市、智慧医疗、智能配送等都在运用大数据分析技术，数据价值已成为一种新常态。

1. 数据安全概述

随着我国《网络安全法》《数据安全法》和《个人信息保护法》的制定与实施，合规性已成为数据安全治理与建设的重要驱动力。在数据安全合规视角下，数据安全的需求和驱动力发生了根本性的改变。因此，本次研究以《网络安全法》正式实施为分界线，将数据安全治理与建设分为两个阶段：无合规性需求与有合规性需求，并分别定义为数据安全 1.0 与数据安全 2.0 阶段 [6]。两个阶段的比较如表 1 所示：

比较维度	数据安全1.0	数据安全2.0
主要特点	是静态的：主动的、数据资产驱动的、投入成本小的	是动态的：被动合规驱动为主、主动数据安全建设为辅，投入成本比较高昂的
关注方向及目标	作为网络安全的一个分支；保障数据完整性、保密性、可用性	保护的数据对象范畴更大，应用场景丰富多样，覆盖数据全生命周期的各个环节
主要目标	未经授权披露、丢失、篡改或销毁	不用/滥用/合法合规利用+权益保护
保护对象	围绕各类资产数据库，包括敏感文档和核心技术材料等	国家主权、国家安全、公共利益或者个人、组织合法权益
主要变化	从小范畴的数据安全变成大范畴的数据安全，从单点的数据安全建设变成体系化的数据安全建设	

表 1 数据安全两个阶段比较

在数据安全 2.0 阶段，开展数据安全治理与建设主要有 3 个驱动力：合规驱动、风险驱动和业务驱动，数据安全合规性已成为其中最重要的驱动力。因此，本次研究将从构建数据安全生态圈基本框架以及各层级的基本内容出发，结合落地实践经验，提出数据安全建设落地执行路径。

2. 数据安全生态圈的定义

在网络安全领域，已经提出构建由中共中央网络安全与信息化委员办公室统一领导、网络安全主管部门协调负责、各相关部门齐抓共管、网络行业组织积极推动、网络公司主动履责、网民及社会各界广泛参与的“六位一体”网络安全生态圈 [7]。借助网络安全生态圈的构建思路，本次研究提出的数据安全生态圈是指全民共同

守护、全社会共同参与、全世界合作互融，形成以“一个中心，二个循环，三个体系，五个关键，八大路线”为顶层设计的数据安全生态融合体系，如图 1 所示。该体系相互影响、相互制约、相互信任、不断演变，并在一定时期内处于相对稳定的动态平衡状态。

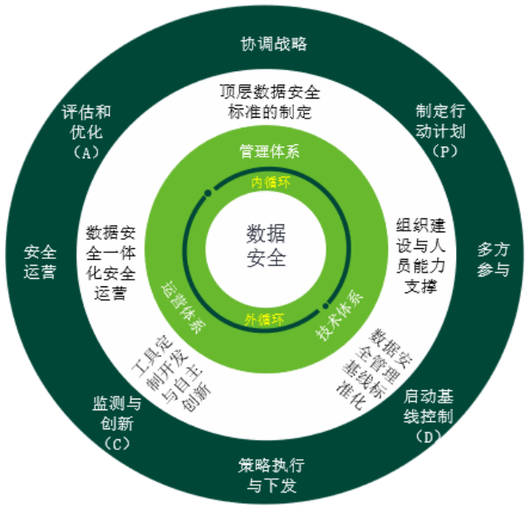


图 1 数据安全生态圈架构

3. 数据安全生态圈的构成

3.1 一个中心

《信息安全技术 数据安全能力成熟度模型》[8]（以下简称“DSMM”）以数据为中心，重点围绕数据生命周期，从组织建设、制度流程、技术工具、人员能力四个维度给出了组织数据安全能力的成熟度模型架构。基于 DSMM 思想，提出以数据安全防护为中心的数据安全生态圈建设目标，围绕数据处理活动展开安全防护，实现组织数据安全治理与能力建设。

3.2 两个循环

在组织数据安全治理与能力建设上要实现“双循环”新格局，“双循环”可解释为“内循环”和“外循环”。“内循环”指组织需要选择适合自身的数据安全建设体系，从管理、技术、运营等不同维度出发，不断获得相对安全；“外循环”则是指在“内循环”的基础上，随着数字产业化的不断发展和完善，组织的数据业务通过合法合规的流动产生价值，为组织带来数字红利。

3.3 三个体系

管理和技术不分家，两者相辅相成。管理和技术的不断融合需要持续运营和不断优化调整，以实现持续自适应的数据安全防护能力。对于不同数据责任主体，数据安全体系建设的工作目标和侧重点也有所不同。参照《电信和互联网行业数据安全标准体系建设指南》[9]、DSMM 等标准，借鉴行业最佳实践，对组织提出建设包含管理体系、技术体系和运营体系的数据安全体系。

3.3.1 管理体系

从组织管理视角出发，通过对政策法规、标准规范及行业主管部门的管理要求进行解读，管理体系设计主要从组织建设、管理策略与制度、管理流程和能力认证四个维度进行横向扩展，再对每个维度进行纵向细分，形成可落地执行的数据安全管理体系，如图 2 所示：

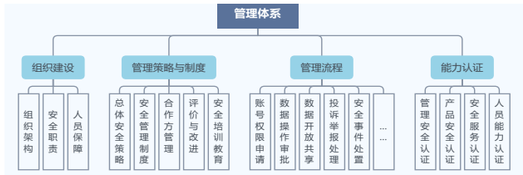


图 2 数据安全管理体系设计

3.3.2 技术体系

数据安全技术体系是数据安全实践工作的保障条件，也是数据安全管理的辅助手段。数据安全技术体系设计从数据采集、数据传输、数据存储、数据处理、数据交换、数据销毁的全生命周期出发，综合组织所有安全域进行整体规划，考虑需要具备的技术手段和工具，如图 3 所示：

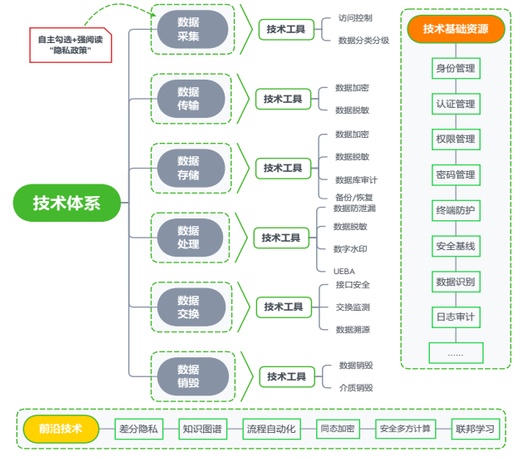


图 3 数据安全技术体系设计

3.3.3 运营体系

数据安全治理与能力建设是一个长期持续的过程，需要在组织内持续落实数据安全管理和技术要求，并基于组织的自身特点、具体业务场景和技术发展不断调整和优化，形成数据安全运营长效机制，为数据安全风险评估、报告、信息共享、监测预警、处置等提供能力支撑。数据安全运营体系设计由浅入深主要分为四个层级：数据监测、常态管控、风险预警、持续改进，如图 4 所示：

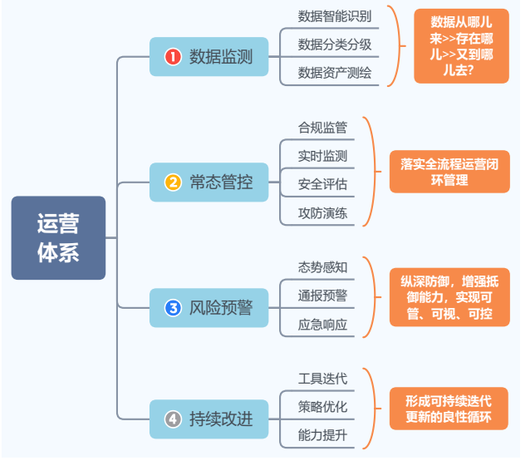


图 4 数据安全运营体系设计

3.4 五个关键

在进行数据安全体系整体规划和设计时，需要根据组织具体

业务和应用场景等合理进行。在此过程中，需要注意以下 5 个关键点。

3.4.1 顶层数据安全标准的制定

数据安全标准的制定遵循层级式设计理念。依据政策法规、标准规范和行业要求，结合实际业务需求，制定数据安全管理标准四级框架，形成一套完整的、可操作的管理制度和管理流程，确保数据安全工作有法可依、有规可循。

3.4.2 组织建设与人员能力支撑

进行数据安全管理首先要成立专门的数据安全管理组织机构，以明确数据安全管理的政策、落实和监督由谁长期负责，确保数据安全管理的有效落实。组织机构可按照决策层、管理层、执行层、监督层的四个层级进行设计，在具体执行过程中，可赋予已有安全团队与其他相关部门数据安全管理的工作职能，或寻求第三方专业团队开展工作。在人员能力支撑方面，已经有关于网络安全行业特殊人才的提案，且近期国家主管部门也在统筹制定人才认定的标准。组织在培养数据安全人员能力时，需要重点关注以下四个能力：数据安全管理能力、数据安全技术能力、数据安全运营能力和数据安全合规能力。

3.4.3 数据安全管理基线标准化

数据安全管理基线（以下简称“数据安全基线”）可以理解为数据安全管理要求，是指组织开展数据处理活动和有关平台系统应遵循的原则和安全保护要求，包括组织保障、制度建设、规范与流程建立等管理性要求、规范执行相关配套技术性要求，以及实行数据安全运营的可持续性要求。如图 5 所示。国务院 2021 年度立法工作计划就包括了制定数据安全管理条例 [10]，2020 年 TC260 发布的《信息安全技术 网络数据处理安全规范》（征求意见稿）[11] 也提出了网络运营者利用网络开展数据处理活动应遵循的规范和安全要求。因此，数据安全基线标准化势在必行。

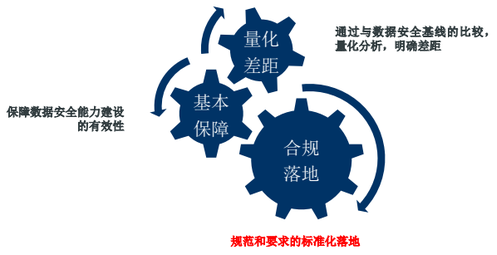


图 5 数据安全基线标准化

3.4.4 工具定制开发与自主创新

多年来，国内外大环境的不断变化为信创产业的诞生创造了绝佳时机。信创产业是数据安全、网络安全的基础，也是“新基建”的重要内容。无论是网络基础设施建设主体、设备厂商、安全厂商还是其他企业，都需要有自主创新思维和自主研发能力，构建起自己的产业标准和生态，逐渐摆脱西方国家的技术限制 [12]。对于组织而言，基于数据资产和数据应用场景，并围绕数据处理活动，通过开发建设数据安全技术产品与工具，形成覆盖事前、事中、事后的数据安全技术能力。

3.4.5 数据安全一体化安全运营

数据流动才能产生价值。数据安全有效地流动需要通过建设数据安全一体化安全运营机制来保障。Gartner 发布的 2021 年 10 大数据和分析趋势之一便是 XOps（包括数据、机器学习、模型、平台），其目标是运用 DevOps 的最佳实践实现效率和规模经济，在保证可靠性、可重用性和可重复性的同时，减少技术和流程的重复并实现自动化。XOps 使企业机构能够通过数据和分析的运营化推动业务价值的实现 [13]。

2021 年 3 月 19 日，绿盟科技正式发布“智慧安全 3.0”理念体系。该理念体系提出以体系化建设为指引，构建“全场景，可信任，实战化”的安全运营能力，达到“全面防护，智能分析，自动响应”的防护效果。可见，数据安全一体化安全运营已不再遥远，其将逐渐成为一种新趋势、新业态。

3.5 八大路线

数据安全生态圈建设的有效落地需要先做好计划，然后实施，实施中进行复核检测，进而改进，如此反复、阶梯式完成，形成一个螺旋式上升的 PDCA 循环。因此便有了以下八大路线：协调战略、制定行动计划（P）、多方参与、启动基线控制（D）、策略执行与下发、监测与创新（C）、安全运营、评估和优化（A）。这八大路线包含两条逻辑链路，一条是“技管并重，分级防护”，即确定统一的数据安全战略与方针，制定多部门共同参与的机制，采用管理和技术相结合的方式，针对数据资产和数据应用场景采取差异化的管控措施，建立持续自适应的数据安全风险和信任评估机制，合理选择安全控制方式，有效降低数据安全风险。另一条是“集中运营，循序渐进”，即建立层次化的数据安全管理组织和集中的数据安全管控措施，全面覆盖数据安全治理与能力建设各个领域，构建可度量、可管理、可改进的集中运营保障体系，为业务的平稳运行提供可信的数据安全支撑环境。

通过以上八大路线的两条逻辑链路的落地实施，最终形成完善的、有效衔接的、响应及时的和运转高效的数据安全生态运营体系。

4. 数据安全 KPI 指标体系实践

近年来，我国不断出台数据安全和个人信息保护法律、法规和相关政策。例如，《网络安全法》聚焦网络安全方向，维护网络空间良好生态；《数据安全法》是数据领域的基础性法律，强调了保障数据安全与促进数据开发利用并重；《民法典》设立专章规范隐私权和个人信息保护，提出个人信息权益并将其定位为人格属性；《个人信息保护法》是针对个人信息保护的专门法律，兼顾个人信息的安全和利用；《信息安全技术 个人信息安全规范》[14] 也是针对个人信息安全，提供了具有可操作性的指引，规范了个人信息处理活动应遵循的原则和安全要求。

本文将根据上述政策法规及标准规范，从数据安全 KPI 指标体系设计入手，进一步分析数据安全体系建设可落地执行的路径，包括 KPI 指标体系模型、KPI 指标体系总体框架以及数据安全评价指标和评价要素 [15] 等。

4.1 KPI 指标体系模型

目前，无论是国家层面还是行业实践，都暂未形成带有考核性质的数据安全 KPI 指标体系标准。因此，从保障数据安全持续优化改进的角度出发，参考智慧城市评价模型以及中国互联网协会发布的团体标准《数据安全治理能力评估方法》[16]，拟在规划数据安全体系建设全景图的基础上，研究设计符合组织管理、技术、运营特点的数据安全 KPI 指标体系模型。模型主要由能力类指标、成效类指标和运营类指标组成，如图 6 所示：

图 6 数据安全 KPI 指标体系模型

4.2 KPI 指标体系总体框架

根据数据安全 KPI 指标体系模型，数据安全 KPI 指标体系总体框架如图 7 所示：

图 7 数据安全 KPI 指标体系总体框架

在数据安全 KPI 指标体系总体框架中，能力类、成效类及运营类指标所涉及的各个方面均为一级指标，每个一级指标包含若干二级指标评价要素，每个二级指标评价要素代表对一级指标某个侧重面的考量依据。总体框架共包含 13 个一级指标、46 个二级指标评价要素。二级指标评价要素的确立可根据当年政策法规及组织自身管理要求动态变化，以适应数据安全与业务的健康发展。

4.3 评价指标及评价要素

根据数据安全 KPI 指标体系总体框架，这里仅对能力类一级指标下的二级指标评价要素进行部分列举和说明，如表 2 所示：

一级指标	二级指标评价要素	评价要素说明
组织建设	战略规划健康度	根据数据安全风险状况建立组织整体数据安全战略规划，数据安全规划的内容应覆盖数据全生命周期的安全风险管控
	资源投入占比率	设立负责组织内部数据安全工作的管理责任部门、岗位和人员，并与人力资源管理等部门进行联动，确保所需资源投入充足
	岗位设定覆盖率	数据安全管理责任部门应配备数据安全管理责任人员（数据安全负责人、个人信息保护负责人等）；相关工作执行部门应设置数据安全工作岗位，负责具体落实数据安全管理工作，包括但不限于数据资产梳理、分类分级、合规性评估、权限管理、安全审计、应急响应、教育培训等
	数据安全责任制	按照岗位人员互斥制衡的原则，建立文档化的数据安全责任矩阵，明确数据操作各环节的操作内容与责任，并定期更新矩阵

表 2 能力类一级指标下的二级指标评价要素

5. 结语

2021 年是“十四五”开局之年，也是构建数据新秩序的开篇之年。国家层面既针对数据安全和个人信息保护领域立法，也针对安全人才培养和认定、安全意识提升等提供认证渠道，这对数据安全治理与能力建设起到很大的推动作用。在有条不紊地推进数据安全生态圈落地实施的同时，应将数据安全与健康发展、维护用户合法权益等作为企业或组织必不可少的内容，确保数据合法有序地流动、共享、交易，积极推动自身数字化转型升级和业务增长。

参考文献

[1] Risk Based Security. 2020 Q3 Data Breach QuickView Report[EB/OL].(2020-10-29)[2021-09-05].https://pages.riskbasedsecurity.com/en/en/2020-q3-data-breach-quickview-report-0.

[2] 方禹 . 关于我国数据治理法治构建的几点思考 [J]. 中国信息安全 ,2020,(10):62 － 64.

[3] 吴沈括 李雨鑫 . GDPR 时代的数据共享治理路径 [J]. 信息安全研究 , 2018, 4(7): 589 － 592.

[4] DPO 社群 . 美加州消费者隐私法案 (CCPA) 修正案汇总中译文 (DPO 沙龙出品) [EB/OL]. (2019-10-22) [2021-09-05]. https://mp.weixin.qq.com/s/7WfeGReYrxQ6HAsIJK5vHA.

[5] CCIA 数据安全工作委员会 . 支撑“个保法”“数安法”落地，相关国标梳理 [EB/OL].(2021-08-23) [2021-09-05].https://mp.weixin.qq.com/s/q2RbdF5oveb12_1MkPjbKQ.

[6] 绿盟科技 . 拥抱合规、超越合规：数据安全前沿技术研究报告 [EB/OL].(2020-12-29) [2021-09-05].https://www.nsfocus.com.cn/html/2020/92_1229/144.html.

[7] 王晓光 . 建设“六位一体”的网络安全生态圈 [J]. 信息安全研究 , 2019, 5(2): 183 － 184.

[8] 中国国家标准化管理委员会 . GB/T 37988-2019 信息安全技术 数据安全能力成熟度模型 [S]. 北京：中国标准出版社，2019.

[9] 中华人民共和国工业和信息化部 . 工业和信息化部办公厅关于印发《电信和互联网行业数据安全标准体系建设指南》的通知 [EB/OL].(2020-12-25)[2021-09-05]. https://www.miit.gov.cn/jgsj/kjs/wjfb/art/2020/art_09151e5d51dc4467a93af73e4ac3116e.html.

[10] 中华人民共和国中央人民政府 . 国务院办公厅关于印发国务院 2021 年度立法工作计划的通知 [EB/OL].(2021-06-11)[2021-09-05].http://www.gov.cn/zhengce/content/2021-06/11/content_5617194.htm.

[11] 全国信息安全标准化技术委员会 . 关于征求《信息安全技术 网络数据处理安全规范》国家标准意见的通知 [EB/OL].(2020-08-31)[2021-09-05].https://www.tc260.org.cn/front/postDetail.html?id=20200830094619.

[12] 四川信创联盟 .2020 年度《四川省信创产业发展报告》正式发布 [EB/OL].(2021-01-21) [2021-09-05].https://mp.weixin.qq.com/s/_ldaOwT6ztG3bz3YxJVDPA.

[13] Gartner 公司 .Gartner 2021 年十大数据和分析趋势 [EB/OL].(2021-02-25) [2021-09-05].https://mp.weixin.qq.com/s/HkYJTFF4wP3qyvmIf3xSeA.

[14] 中国国家标准化管理委员会 . GB/T 35273-2020 信息安全技术 个人信息安全规范 [S]. 北京：中国标准出版社，2020.

[15] 中国国家标准化管理委员会 . GB/T 34680.1-2017 智慧城市评价模型及基础评价指标体系 第 1 部分：总体框架及分项评价指标制定的要求 [S]. 北京：中国标准出版社，2017.

[16] 中国互联网协会 . T/ISC-0011-2021 数据安全治理能力评估方法 [S] . 北京：中国标准出版社，2021.

波谲云诡，浅谈工业互联网的跨界安全挑战

绿盟科技 能力中心&格物实验室 任心

摘要：随着工业互联网的发展，信息安全问题愈发引人关注，工业互联网信息安全已跨界影响到另一个安全领域。研究表明，工业互联网信息安全可能直接影响工控系统的功能安全，本文就工业互联网信息安全对工控系统功能安全的影响进行讨论。

关键词：工控 工业互联网 功能安全 信息安全

1. 引言

中国是工业大国，自“十三五”以来，工业总量规模稳步提升。当下我国数字战略已全面启动，工业互联网是连接工业制造业和数字信息技术的桥梁。在工业互联网中，工控系统发挥着重要作用，它在工业互联网中的重要性相当于人体中的视、听、嗅、味、触，都是获取信息的主要通道。随着工业互联网的发展，工控系统接口越来越开放，解决外部对系统恶意入侵的信息安全问题日益迫切，国家出台大量举措，大力加强工业互联网信息安全保障能力建设。与此同时，另一个安全领域——功能安全，伴随着工业互联网信息安全问题出现。早期人们对电子技术和计算机系统的可靠性、安全性没有信心，而功能安全保障电气、电子、计算机、现场总线技术构成的安全相关系统安全，如今通过外部网络入侵也可能会影响功能安全。2008 年，伊拉克向土耳其输送原油的输油管道发生爆炸，土耳其为监控 1099 英里石油管道，在这条管道内安装了大量探测器和摄像头。然而在爆炸将管道破坏前，却没有引发遇险信号。根据事故调查，原因是黑客关闭了警报、切断了通信联系、给管道内的原油大幅增压。

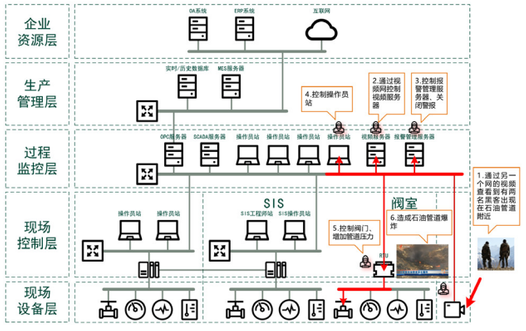


图 1 石油管道入侵链示意图

可以看出工业信息安全可能直接影响功能安全问题，轻则造成财产损失，重则造成人身伤害甚至危害国家安全，于是为保障安全相关系统的功能安全再次进入人们的视线。

2. 功能安全的定义及标准

功能安全是依赖自动保护的系统或设备整体安全的一部分，该自动保护系统需要对其输入做出正确响应，对失败有可预测的反应，包括人为错误、硬件故障和操作 / 环境压力等。例如，锅炉控制系统在点火后会对压力进行监控，当压力到达限定值可能造成危险时，锅炉控制系统会自动关闭燃料系统。如果该机制失效，锅炉持续燃烧，压力超过限定值并持续增高，就会发生爆炸。

工业革命使我们的生活发生了翻天覆地的变化，机器取代人力，大规模工厂化生产取代个体手工生产。然而，人们在享受工业红利的同时，由此引发的灾难也接踵而至。在二十世纪，工伤事故已成为人类最严重的死因之一 [1]。尤其是石油化工与核工业领域发生了多次爆炸或泄露事件，如 1957 年英国军用温茨凯尔反应堆事故、1979 年美国三里岛核事故、1984 年印度博帕尔的联合碳化物工厂毒气泄漏事故、1986 年苏联的切尔诺贝利核事故。这些工业事故的起因都是安全相关系统的功能失效。

就是在这种背景下，经过不断实践和摸索，欧美颁布了成套的功能安全相关产品指令和设计标准，并深入到各个领域，如汽车 (ISO26262)、轨道控制 (EN5012X)、核电 (EN61513)、工业装备及机器控制 (EN62601,ENISO 13849-1/2)、过程控制 (EN61511) 等 [2]，国际上，IEC（国际电工委员会）形成的 IEC61508、IEC61511 等系列标准在工业领域引起强烈反响，已经逐步成为各国家、行业广泛认可的基本功能安全标准，中国也仿效并形成了相应的国家标准，其他行业性功能安全标准也在参照并将逐步形成为国家行业性标准。

标准号	名称	行业/领域	颁布方	对应国标
IEC 61508:2010	电器/电子/可编程电子安全相关系统的功能安全	电器/电子/可编程电子	IEC/TC65	GB/T 20438:2006
IEC 61511:2003	过程工业安全仪表系统的功能安全	过程工业仪表	IEC/TC65	GB/T 21109:2007
IEC 61513:2011	核电厂核能工业的安全仪表系统	核电	IEC	/
IEC 62061:2005	机械安全与安全有关的电气、电子和可编程电子控制系统的功能安全	机械安全	IEC/TC44	GB 28526:2012
IEC/TR 62061-1:2010	指导ISO 13849-1和IEC62061中用于机械的安全相关控制系统设计的应用指南	机械安全相关控制系统	IEC/TC44、ISO/TC199	GB/T 34136:2017
IEC 61784-3:2016	工业通信网络协议集第三部分：现场总线功能安全	数据通信	IEC/TC65	GB/T 34040:2017
IEC 61131-6:2012	可编程序控制器第6部分：功能安全	设备安全	IEC/TC65	GB/T 15969.6-2015

表 1 功能安全相关标准

3．工业信息安全对功能安全的影响

经过整理归纳，我们总结出工业互联网信息安全影响功能安全的三种情况。

3.1 网络入侵导致功能安全的失效，继而影响系统安全

以 SIS（安全仪表系统）为例，SIS 系统大多应用于石油化工、电力等行业，在工控系统发生危险时，SIS 系统使生产装置进入一个预定义的安全停车工况，从而使危险降低到可以接受的最低程度，以保证人员、设备、生产装置和环境的安全。工控系统设计之初，没有将信息安全考虑在内，这使得不法分子能够通过网络入侵工控系统，致使原本的功能安全失效，造成系统故障，继而演变成危险源，使工控系统出现不可接受的风险时不能将风险降低到可接受范围，最终导致事故的发生。

3.2 工业网信息安全产品影响功能安全

当前工业信息安全产品经过公安部检测、取得销售许可和检测报告就可以被企业购买，应用在工控系统中。然而公安部在检测工业信息安全产品时依据的是信息安全技术相关产品的技术要求和测试评价方法，并不会考虑应用场景中的功能安全与工业信息安全产品相结合导致的新问题。2020 年 12 月，内蒙古某电厂就发生了由工业信息安全产品问题导致的机组跳机事件。此事影响重大，经查是一款旁路的设备，连接两台机组的交换机。由于设备在未加电状态时两个网口处于 Bypass 连通状态，两台机组 DCS 网络直接互通，最终导致两台机组跳机。Bypass 一般应用于工业场景，且只有在串联设备中才可以起到作用，在串联设备故障或断电时能第一时间保障业务不被中断。

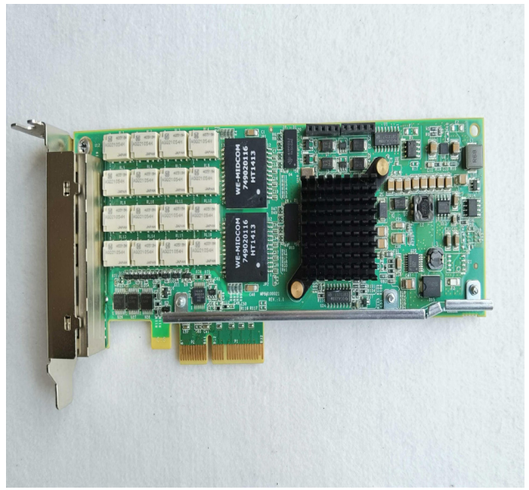


图 2 硬件 Bypass 图

旁路设备携带 Bypass 主要由于部分安全厂商倾向于避免硬件设计差异，将携带 Bypass 硬件的设备同时应用于多款安全产品，包括串联部署的防火墙、旁路部署的流量分析、入侵检测、日志采集、安全管理等设备，加上开发时缺少相应管理流程，Bypass 未从底层关闭，最终导致事故的发生。

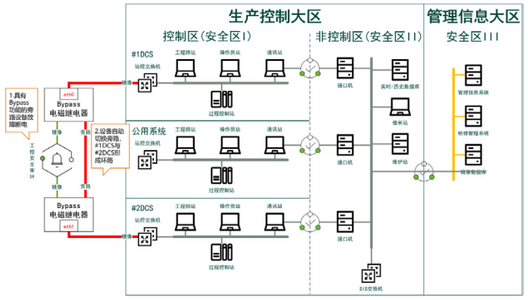


图 3 停电事故分析图

工控系统中对业务的实时性要求非常高，网络延迟、抖动都有可能会影响功能安全，串联部署的工业信息安全产品显然会增加这方面的不确定性。虽然目前还没有相关案例证明延迟、抖动会影响工控系统的功能安全，但随着工业信息安全的不断深入下沉，这种风险在不断提高。同时，由于工业信息安全产品实施的人员缺少对工控系统功能安全的了解，配置不正确的工业信息安全产品策略也可能会阻断正常通信，影响功能安全。

3.3 工业信息安全加强功能安全，继而提升工业安全

工业信息安全与功能安全有许多相似性，当两者对安全的需求出现重叠时，工业信息安全就可以加强功能安全。如网络安全等级保护中的安全通信网络就在通信传输中有要求，应采用校验技术或密码技术保证通信过程中数据的完整性，功能安全中也有通信完整性的需求，因此在不法分子攻破一个组件时，另一个组件依旧可以起到作用。还有一种情况，工业信息安全产品可以监测工控系统的状态，如在某石油管道的关键路径位置部署的工业安全审计系统，不光可以分析网络流量异常行为，同时还对 IEC104 和 OPC 数据采集进行监控，当发现网络数据中断时，可分析故障原因，缩短故障处理时间。

4. 结语

功能安全在漫长的岁月中经历了数不尽的事故才逐步走向完善，工业信息安全技术在 Purdue 模型中大多应用于 2 至 4 层，随着工业互联网、5G、物联网等技术的发展，信息安全技术不断向下与功能安全紧密结合，二者只谈其一保证不了工业互联网的安全。本文分析了工业系统信息安全对功能安全的三种影响，但还停留在浅层辨析二者关系。工业信息安全与功能安全融合后还会暴露更多的问题，这种问题不光存在于技术层面，管理层面也同样存在。未来，我们应从实践中建立联系，逐渐积累关于二者之间影响，找到覆盖工业互联网全生命周期的一套方法，保障工业互联网不受危害。

参考文献

[1] 史学玲 . 功能安全标准的历史过程与发展趋势 [J]. 仪器仪表标准化与计量 ,2006,(002):6 － 8.

[2] 什么是功能安全，https://blog.csdn.net/weixin_42229404/article/details/80935785.

绿盟科技智慧安全3.0助力车联网行业网络安全实践

绿盟科技 物联网安全产品部 刘大鹏

摘要：近年来，车联网产业快速发展，技术创新日益活跃，新型应用蓬勃发展，智能化水平不断提升，涵盖通信、终端设备、整车制造、运营服务、测试认证、高精度定位及地图等完整的产业生态基本形成。伴随智能化和网联化不断推进，“万物互联”下的车联网使得汽车不再是传统的相对独立的封闭系统，网络入侵、木马病毒、个人隐私泄露等互联网安全威胁也逐步渗透至车联网领域，绿盟科技紧跟车联网行业发展趋势，秉承“智慧安全 3.0”理念，深度参与车联网安全技术研究与攻防实践，为构建车联网安全保障体系，提升行业车联网的网络安全水平，贡献绿盟科技的力量。

关键词：车联网 智慧安全 3.0 智能网联汽车

1. 车联网发展概况

车联网（智能网联汽车）产业是汽车、电子、信息通信、道路交通运输等行业深度融合的新型产业形态，是目前各国推进汽车及通信产业技术创新和产业转型化发展的重要领域，我国在宏观顶层设计上也在不断引导和支撑着车联网行业的发展。随着国家《交通强国建设纲要》与“新基建”政策相继出台，“新基建”与传统基建的深度融合，持续提升交通路网的感知能力、信息化能力、交互能力，聚焦人、车、路、云一体化协同发展的车路协同应用成为当前交通行业“新基建”的发展核心。

车路协同是车联网的高级发展阶段，全面融合通信、汽车、交通、信息等多个领域，构建了一个全新的生态。车路协同主要有三大核心组成部分：智能车载系统（车端）、智能路侧系统（路侧端 + 云端）和通信平台。智能车载系统负责车载端的海量数据实时处理和多传感器数据融合，保证车辆在各种复杂的情况下稳定、安全行驶；智能路侧系统负责路况信息搜集与边缘侧计算，完成对路况的数字化感知和就近云端算力部署；通信平台负责提供车 - 车、车 - 路间实时传输的信息管道，通过低延时、高可靠、快速接入的网络环境，保障车端与路侧端的信息实时交互。三者恰好构成智慧交通场景下协同感知与协同决策的闭环。

2. 车联网安全风险

车路协同领域研究不断深入，越来越多的网络安全风险也随之暴露。车侧网络监管安全相关测评方法和评判标准尚未实现统一，在跨平台、跨车型、跨终端、跨模组间缺乏一致性安全要求，且车侧业务、安全等数据间互联互通的流转存在阻塞，数据烟囱壁垒短时间无法打破；路侧智能设备种类繁多，且长期处于无人值守的环境中，缺少了人对终端节点的有效监控。终端节点更具有脆弱性，将面临更多的安全威胁，容易遭受物理入侵、伪造或假冒入侵、信号泄露与干扰、资源耗尽入侵、隐私泄露威胁等；车—车、车—路间网络通讯缺乏有效安全保障方式，面临网络监听、数据泄露等风险。据工信部 2020 年数据，通过车联网安全威胁监测平台，累计已监测发现针对整车企业、车联网信息服务提供商等网络和平台的恶意行为 280 余万次 [1]；另据 Upstream 最新报告说明，互联车辆数量的增加导致黑客利用的漏洞和入口点增加，仅在 2020 年，就公开报告了 200 多起汽车网络安全事件 [2]。

3. 车联网全防护思路

绿盟科技在车联网领域的安全研究已经深耕多年，在公司新的“智慧安全 3.0”理念体系指导下，我们对车联网的网络安全防护思路也有了全新的认识，包括新领域全要素全类型防护思路、基于可信任的安全防护思路、以实车安全检测及攻防对抗促进安全防护思路等，通过不断研究和探索，以保障车联网网络持续安全运行。

3.1 新领域全要素全类型防护

在《国家综合立体交通网规划纲要》中，车联网被称为重要融合基础设施，强调要提升其安全保障能力，即车载 ECU、车内网络、汽车移动 APP、高低频无线通信、智能网联汽车云平台等车联网全要素的安全防护能力。

车联网网络入侵类型主要包括远程恶意控制、敏感信息泄露、车内网络入侵、系统权限恶意获取、任意应用安装、报文恶意篡改、拒绝服务、升级包恶意篡改、系统恶意刷写、恶意报文执行、功能篡改、功能失效等。因此，需要对以上入侵类型有充分研究，具备全类型纵深安全防护能力。

3.2 基于可信任机制的安全防护

基于可信任机制的安全防护需要从以下方面考虑：

(1) 可信通信网络

传输链路的安全可信加密通信能力，包括是否搭载硬件加密卡、是否支持国密算法、是否支持交通行业证书导入和适配，以及是否支持与证书签发系统对接，进而实现双向身份的认证和验签。

(2) 可信安全边界

满足车联网“云、管、边、端”的边界隔离、访问控制、入侵防范、恶意代码防范、安全审计以及物联网安全扩展要求，包括对门架专用终端设备、路侧装置自动识别（车道控制器、RSU、车牌图像识别设备、高清摄像机、激光雷达、毫米波雷达、路侧智能站等），并能实现门架及收费站设备实时仿冒检测。

(3) 可信终端设备

关键组件系统加固，实现对终端 CPU、内存、进程、网络、文件、DNS 等的主动监测和防护，形成完整车载终端流量数据采集、监控、防护、预警处置等能力。并与安全运营中心形成车联网端到端安全联动，在合理范围内执行终端网络阻断等处置策略，最终实现车联网安全运营中心 SOC 下车企、车联网运营商等业务安全赋能。

(4) 可信平台 / 安全运营中心

对车联网车端、通信链路、平台、移动 APP 等资产及海量业务安全数据进行统一采集和归类汇总，对业务场景中各类网络安全事件进行实时监控和集中审计，并进一步进行事件关联性挖掘，识别安全异常和潜在安全事件；识别和阻断具有入侵特征的网络流量，防御已知 / 未知入侵类型，执行入侵预警、拦截；将威胁事件与车联网业务进行有机结合，通过态势感知大屏将全局的安全态势以图形化的方式直观呈现，将安全由不可见变为可见。通过构建可信平台 / 安全运营中心，最终实现车路协同计算环境、边界、通信网络整体安全可信管理。

3.3 以实车安全检测及攻防对抗促进安全防护

国家相关部门非常重视车联网的网络安全检测和实车安全评估工作，相关检测标准正在逐步完善中，同时还通过建设车联网安全态势感知、智能网联汽车安全检测等平台，逐渐增强车联网安全监测预警及检测评估服务能力。

行业主机厂、零部件厂等车联网企业需要做好车联网平台、APP 以及关键部件等重点区域的网络安全风险自查、自评工作。通过采用合规认证和渗透测试等形式，采用模拟实施网络入侵、软件及数据篡改、挖掘漏洞、敏感数据加解密等测评方法，完成风险点的分析、整改及测评等工作，符合国家车联网系统相关安全标准法规，保障对整车、部件、车联网系统等方面的网络安全可控、健康运行。

4. 车联网安全实践

4.1 车联网安全技术研究

近年来，绿盟科技在车联网安全方向持续发力，积极跟踪和支撑工信部车联网安全测评和调研工作，参与编制行业多部安全标准、白皮书。2020 年，绿盟科技参与并成功中标 2020 年工信部《车联网安全态势感知平台》《智能网联汽车信息安全检测平台》两个车联网安全专项。此外，绿盟科技申报的《车联网安全监测与防护（SOC&SDK）系统》入选 2020 — 2021 年度工信部物联网关键技术与平台创新类、集成创新与融合应用类示范项目。并且，在产品研发层面，陆续将攻防检测与防护技能植入到车联网安全产品中，逐步形成了车联网安全合规审计、攻防检测、终端安全防护、平台安全态势监管等四大类车联网安全产品能力。

车联网安全产品四大能力

4.2 车联网实车安全检测与实战攻防

绿盟科技积极参与工信部等相关部委组织的车联网安全检测评估及实车安全测评等顶层实践工作，多次参加国家级、行业级车联网安全攻防比赛，以实战化安全检测及攻防对抗比赛促进安全

检测与防护技术研究，已完成对一汽、比亚迪、长城、长安、奇瑞、吉利、柳汽、上汽通用五菱等整车、零部件系统信息安全测评，涉及车型数十款。

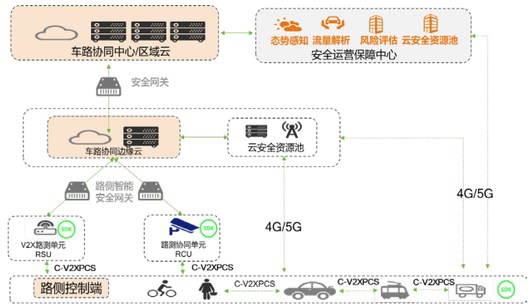
信息安全挑战赛是世界智能驾驶挑战赛的重要赛事内容之一，比赛围绕汽车信息安全理论知识构建及汽车远程非物理接触式入侵测试展开，旨在促进汽车行业信息安全的发展和汽车整体防护水平的提升。绿盟科技 M01N 战队在 2020 年的赛事上成为唯一一家进入十六强并获奖的网络安全企业战队。

车联网信息安全技能大赛作为中国汽车安全与召回技术论坛的重要活动之一，旨在通过竞赛形式提升车辆信息安全水平，挖掘车辆信息安全防护漏洞，聚焦智能网联汽车安全。绿盟科技 M01N 战队凭借丰富的传统安全实战攻防经验和格物实验室对车联网安全的研究积累，通过层层考验，最终获得第一届、第二届车联网安全攻防挑战赛优胜奖，第三届大赛车联网信息安全技能大赛一等奖，车联网云平台精英夺旗赛第一名，智能汽车门锁无线破解赛挑战成功奖的荣誉。

4.3 车路协同网络安全解决方案

4.3.1 车路协同网络安全架构

绿盟科技根据车路协同感知层、网络层、平台层和应用层四级网络架构，基于车－路－云数据交互场景，构建了如下图所示的车路协同网络安全架构。



车路协同网络安全架构图

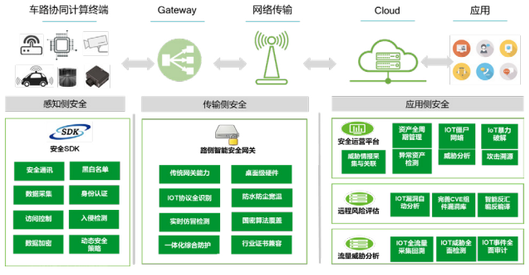
整个安全架构包括车端（感知侧）、管端（传输侧）、云端（平台 / 应用侧）三层。

- 车端层（感知侧），主要构建包括接口安全防护、安全网关防护、车载网络防护和车载终端节点安全防护等车端安全检测、加密、防护等能力。
- 管端层（传输侧），主要构建包括安全网关防护、安全通信协议、可信接入网络等能力。
- 云端层（平台/应用侧），主要构建包括基于云安全资源池的边缘云平台防护以及车路协同中心/区域云的安全运营保障中心，形成边界安全、安全认证、隐私保护、流量解析、风险评估及实时感知网络入侵等能力。

4.3.2 车路协同网络安全技术方案

绿盟科技车路协同网络安全技术方案，着眼于规模化车路协同

应用，采用了车载可信级安全 SDK+ 路侧智能安全网关 + 安全运营平台端到端的安全联动架构模式，构建监测、检测、预警、防御、响应与应急处置安全能力，全面覆盖感知侧、传输侧、平台 / 应用侧防护场景，为智能交通领域的网络安全保驾护航。



车路协同网络安全技术方案

- 感知侧安全

通过在车载端 OBU 设备、路侧端 RSU 设备上部署具备 AI 自动化终端可信安全 SDK 监测、检测与防护产品，构建车载终端主动免疫和可信保障机制，上报异常信息进行深度威胁分析，执行安全策略指令下发、OTA 应急恢复。形成终端可信级安全规则和基线库，并加持基于 HSM 硬加密、双向身份认证以及数据完整性保护能力，实现敏感信息安全存储、终端安全认证等端侧设备（OBU、RSU 等）网络安全主动免疫能力。

- 传输侧安全

通过在通信网络中部署接入安全网关，实现网内流量数据采集、RSU 设备监控。根据业务需求提供一体化访问控制策略，有效防御多类型网络入侵行为。基于数据流量和资产指纹，实时监控底层设备被冒用情况。同时设备适配基于 SM1-SM4 国密算法的 IPsec VPN、SSL VPN 功能，兼容硬件加密卡与交通行业证书，有效保障交通行业适配性与兼容性。

- 平台/应用侧安全

通过在车路协同云端部署安全运营中心 SOC，形成基于网联汽车入侵流量监测与预警、车联网端攻防态势感知与应急、车联网漏洞闭环处置等安全能力，还形成高并发、低时延、大连接的车联网系统及终端异常行为分析及安全预警能力。既迎合汽车智能化、网联化技术对网络安全迫切需求，也能赋能车联网企业，辅助其企业系统及终端应用数据安全合理回溯与应急处置，还丰富了车联网安全服务内容，包括但不限于：远程网络安全风险评估，依托威胁知识库建立，执行车联网端到端威胁分析、风险评估、漏洞挖掘等安全风控工作。

从方案效果看，本方案通过 OBU、RSU 终端及平台探针部署、威胁情报采集等，收集车路协同通信网内安全数据信息，并基于大数据关联分析处理，形成主动探测、被动诱捕、流量分析、僵木蠕、DDoS 入侵、APT 检测等安全监测、检测、预警、防御、响应与应急处置安全能力，结合安全咨询、渗透测试、全生命周期安全风控等安全服务，构建车路协同安全运营体系；从方案价值看，能够满足车路协同安全合规及“新基建”网络安全建设迫切需求，进一步保障整个车路协同应用安全、可控、健康发展。

5. 结语

绿盟科技将紧跟车联网行业发展趋势，秉承“智慧安全 3.0”理念，持续深耕车路云网一体化风险评估、网联汽车多域多层安全检测、车云联动安全监测与防护、规模化网联汽车安全态势感知与预警、车联网安全漏洞持续探测与修复、安全运营体系等车联网安全技术研究；深度参与行业内车联网安全相关课题项目及安全标准 / 白皮书编制；积极参与构建产业生态，为车联网行业客户安全赋能，合力推动国家车联网行业安全、健康、有序发展。

参考文献

[1] http://caijing.chinadaily.com.cn/a/202010/16/WS5f894065a3101e7ce9729a5e.html.

[2] 2021 年全球汽车网络安全报告 [EB/OL]Upstream Security-Global Automotive Cybersecurity Report 2021.

浅谈电子政务密码应用推进思路

绿盟科技 行业技术中心 李成日

摘要：随着《密码法》《关键信息基础设施安全保护条例》《信息安全技术 网络安全等级保护基本要求》《信息安全技术 信息系统密码应用基本要求》等法律法规及标准规范的发布，密码使用由行政推进向依法规范应用转变，强化了密码应用要求。本文围绕一体化政务服务平台、信息资源整合共享等政务应用场景，介绍商用密码在政务领域应用解决方案，浅谈电子政务密码应用推进思路。

关键词：商用密码 电子政务 一体化政务服务平台

1. 概述

密码算法作为密码应用的核心要素，在不同领域的密码应用场景中发挥着重要作用。当前，商用密码算法在电子政务领域应用取得了较好的进展，但仍存在密码应用覆盖少、密码算法不合规、密码应用不合理等情况。因此，我们需要以密码使用优质高效、密码管理安全高效为核心目标，依照相关法律、法规规定和密码相关国家标准的强制性要求，同步规划、同步建设、同步运行密码保障系统。同时，完善商用密码管理制度，规范商用密码应用，推进新技术新模式密码应用，全面提升政务信息系统密码使用管理和服务水平，增强网络安全防护和风险防控能力，夯实安全保障基础。

2. 问题与挑战

以互联网 + 政务服务、一体化政务服务平台建设为代表的政务信息化建设，核心变革在于充分利用互联网技术，从传统的单一业务受理转向面向群众办事的综合性事项办理，实现政务资源共享，打通政府部门间办事流程，最终让群众获得协同、高效的一网通办服务。但在为群众办事提供便利的同时，也面临愈来愈突显的网络安全威胁与风险。

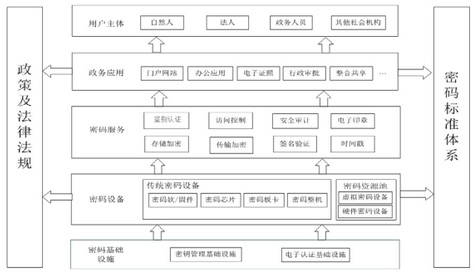
首先，**网络身份难以甄别**，假冒鉴别难度大。用户不见面，脱离了传统面对面身份核实环节，网络身份甄别环节易被入侵或身份易被假冒，轻则损害群众利益，重则影响社会稳定。其次，**敏感数据泄露**，“衍生灾害”严重。随着大量电子政务信息、应用数据通过互联网汇聚、发布、展示、共享，各种政府、组织、个人信息数据遍布终端、网络和云上，加之互联网黑色产业链不断发展的趋势，数据泄露威胁日益加剧。

综上，我们需要认识到，政务信息系统一旦遭到破坏、丧失功能或者数据泄露，可能严重危害国家安全、社会稳定和公共利益。政务信息系统应基于原有安全防护体系，充分发挥密码的支撑保障作用，全面推进以密码技术为核心的政务信息系统安全防护体系建设。

3. 政务信息系统密码应用框架

基于相应政策及法律法规文件要求，围绕政务信息系统密码

应用需求，构建政务信息系统密码应用框架，如下图所示。



政务信息系统密码应用框架由密码基础设施、密码设备、密码服务、政务应用、用户主体、密码标准体系、相关政策及法律法规组成。

(1) 密码基础设施

密码基础设施是相对独立的密码基础组件，包括电子认证基础设施、密钥管理基础设施。电子认证基础设施提供数字证书管理服务。密钥管理基础设施包括非对称密钥管理和对称密钥管理，提供密钥产生、存储、传输、备份、更新等全生命周期的服务。

(2) 密码设备

密码设备是提供数字签名 / 验证、加密 / 解密等密码运算功能的部件或产品，应当通过国家密码管理部门审查，获得商用密码产品型号证书。密码设备包括密码软 / 固件、密码芯片、密码板卡、密码整机等。密码资源池是在云计算环境下基于密码设备构建的。物理密码设备被虚拟化为虚拟密码设备，按需分配给租户使用，对外提供云密码服务。

(3) 密码服务

密码服务由存储加密、传输加密、签名验证等通用密码服务和鉴别认证、访问控制、安全审计、时间戳、电子印章等典型密码服务组成。基于密码技术，通过鉴别认证、访问控制、安全审计、时间戳、签名 / 验证等密码服务提供信任支撑，实现可信、可控、可管、可追溯。通过加密传输，实现端到端传输安全；通过数据加密存储，保证重要数据存储安全。

(4) 政务应用

是由政务信息系统提供的各类应用，典型应用包括门户网站、整合共享、办公应用、电子证照、行政审批等，相关政务应用需要密码应用框架的基础支撑。

(5) 用户主体

面向社会公众提供服务的政务信息系统用户主体包括自然人、法人、政务人员及其他社会组织等。

4. 典型政务密码应用

政务信息化发展推动传统单一垂直政务系统建设模式全面转向跨区域、跨部门、跨业务的全国一张网、全业务一窗办理的新建设模式，实现自然人、法人用户通过政府政务服务门户网站来获得一站式服务，政务业务在多个政府部门业务的全网通办，以及全程无纸化、全在线的一窗办理目标。新模式下，应用创新、服务创新、科技创新将成为常态，基于传统的身份认证、数据机密性、完整性保护以及业务抗抵赖密码应用，拓展到基于“两微一端”的移动密码应用、基于云计算的密码云服务、基于数字标签的共享交换服务的新模式新领域，为门户网站、政府办公、资源共享交换等众多政务系统保驾护航。

(1) 门户网站密码应用

随着“互联网 + 政务服务”工作的大力推进，政务服务门户与传统政府网站相比，关注度更高，相关信息涉及个人隐私和企业商业秘密，应着力加强政府网站的安全防护能力。通过密码技术，提供数据机密性保护、完整性保护、个人信息保护等相应密码安全支撑，防止门户网站网页篡改、服务中断、信息窃取，强化政府网站安全防护能力。依托密码服务，对门户网站提供用户权限管理和统一身份认证，实现门户网站的授权访问控制和单点登录服务。另外，通过协同数字签名技术，实现移动终端用户的身份认证。

(2) 办公系统密码应用

办公系统主要提供公文管理、内容管理、资产管理、人事管理、档案管理、邮件服务、信息报送、OA 系统等主要业务。然而，由于终端的脆弱性、网络开放性和无线传输安全的脆弱性，办公系统存在终端身份合法性、终端数据安全保护、远程传输信息的机密性和完整性保护以及应用安全防护等方面的安全隐患。通过为政务办公提供身份鉴别、授权访问、签名验签、电子签章、数据加密、安全邮件等密码服务，保障办公系统的终端安全、接入安全、传输安全和应用安全。

(3) 信息资源整合共享交换

政务信息系统整合共享是大数据战略的重要支撑，也是国家目前重点推动的重要工程。政务外网接入部门各信息系统之间要实现全国一体化的数据共享交换，在打破信息孤岛的情况下，也打破了原有信息系统防护边界和责任相对独立和明确的状况。因此，既要实现数据交互的畅通，又要明确数据安全责任边界，保障数据共享交换的安全，是政务信息资源整合共享全面推进和落实的基础。通过密码技术，构建统一的数据安全支撑服务体系，为数据在汇集、发布、共享交换、使用环节中提供加解密、签名 / 验签、时间戳、认证与访问控制、数据安全审计等平台化支撑服务，防止数据滥用和数据泄露，加强敏感数据和个人隐私数据的有效管控和防护，实现数据可信验证、数据防泄漏、数据防篡改、数据脱敏，保证共享交换过程安全。基于密码技术、数据标签技术，给数据加上“身份”，在发布、交换和使用等环节中明确主体身份，实现数据分类分级管理，提升数据可控使用、数据回溯和追责能力。

5. 关于电子政务密码应用推进的几点考虑

推进政务信息系统全面应用商用密码具有重要意义，这是一项长期复杂的系统性工程，做好这项工作需要坚持几个原则，注意几个问题。

(1) 依法依规，全面落实密码应用要求

政务信息系统应当坚决贯彻落实国家法律、法规规定，密

码有关政策要求，采用的密码算法、技术应当遵循密码相关国家标准和行业标准，系统中使用的密码产品与密码模块应通过国家密码管理部门核准，采用信息安全产品的密码部门应当通过密码检测，取得信息安全产品密码认证证书，第三方服务机构提供的密码应用安全性评估应通过国家密码管理部门许可。

(2) 放心好用，全面服务政务民生应用

积极推进政务服务平台公共支撑一体化建设，针对政务服务平台建设运行、安全保障等关键技术环节，建立以密码技术为核心的配套支撑体系，不断提升政务服务便捷化、个性化、智慧化、安全化水平。建设统一电子政务外网，基于密码技术加强网络安全保障。建设统一身份认证系统，实现不同电子认证基础设施之间的互信互认，实现“一次认证、全网通办”。基于数字签名等技术，建设统一电子印章系统、统一电子证照共享服务系统，实现电子印章和证照全国互信互任。

密码产品服务政务信息系统时应遵循好用易用原则，针对各类应用主体和安全保障需求，尽量简化操作步骤甚至进行自动化应用和配置，优化业务流程，做好容灾备份，提升密码产品使用体验，尽力做到用户无感化的安全保障过程。

(3) 创新发展，积极运用新技术解决新问题

为满足政务应用复杂化、精细化、专业化的需求，要坚持创新发展理念，深化云计算、大数据、区块链等新技术与密码技术的融合应用、深化应用。运用大数据和密码技术，实现政务数据精细化管理、智能化分析，推动数据资产的开放使用；运用云计算和密码技术，实现集约建设、共享利用，特别在政务云平台技术中，着力打造统一规范、安全可靠的政务云平台；运用区块链和密码技术，研究共享数据确权确责，为政务数据共享交换安全提供新方案。综合运用云计算、大数据、人工智能等新技术，开展态势分析、行为监控、网络安全预警，提升社会治理能力。

(4) 分工负责，全面推进协调联动机制建设

政务信息系统密码保障系统应与政务信息系统协同共建，整合各类政务服务资源，整体统筹，提升建设集约化、管理规范化、服务便利化水平和密码应用安全保障水平。各地区各部门政务信息系统的建设、运维、管理部门，应与同级密码管理部门建立健全管理协调工作机制，分工负责，积极配合，在政务信息系统规划建设、政府采购、测评检查等重点环节，加强协调联动，从管理、服务、监督等方面形成合力，确保政务信息系统安全可靠、方便高效运转。

6. 结语

为了保障电子政务的关键信息基础设施安全，维护网络安全，我们应以密码应用框架作为电子政务整体安全保障框架的基石，聚焦政务信息化，结合等保测评与密码应用安全性评估，实现“同步规划、同步建设、同步运行及密码应用安全性评估”，提升电子政务安全保障能力，为政务数字化转型保驾护航。

逃逸风云再起：从CVE-2017-1002101到CVE-2021-25741

绿盟科技 创新中心&星云实验室 阮博男

摘要：近日，研究人员向 Kubernetes（K8s）安全团队报告了一个可导致容器逃逸的安全漏洞，获得编号 CVE-2021-25741。事实上，该漏洞的本质是 CVE-2017-1002101 漏洞的补丁不充分。本文将对这两个漏洞进行关联分析。

关键词：容器逃逸 符号链接 subPath 竞态条件

1. 前言

近日，研究人员向 Kubernetes（K8s）安全团队报告了一个可导致容器逃逸的安全漏洞[1]，获得编号 CVE-2021-25741，目前的 CVSS 3.x 评分为 8.1[2]，属于高危漏洞。该漏洞引起社区的广泛讨论[3]。有人指出，CVE-2021-25741 漏洞是由 2017 年的 CVE-2017-1002101 漏洞的补丁不充分导致，事实也的确如此。

CVE-2017-1002101 是一个 Kubernetes 的文件系统逃逸漏洞，允许黑客使用 subPath 卷挂载来访问卷空间外的文件或目录，CVSS 3.x 评分为 9.8[4]。所有 v1.3.x、v1.4.x、v1.5.x、v1.6.x 及低于 v1.7.14、v1.8.9 和 v1.9.4 版本的 Kubernetes 均受到影响。该漏洞由 Maxim Ivanov 提交[5]。

这两个漏洞都与 Linux 系统的符号链接机制有关，而这一机制曾引发了数量可观的安全漏洞。

简而言之，CVE-2017-1002101 的成因是，Kubernetes 在宿主机文件系统上解析了 Pod 滥用 subPath 机制创建的符号链接，故而宿主机上任意路径（如根目录）能够被挂载到黑客可控的恶意容器中，导致容器逃逸。官方对此的修补思路是，借助路径检查和类似“锁”的机制，确保恶意用户通过 subPath 挂载的路径不是非预期的符号链接。然而，百密一疏，纵使官方的修复方案已经考虑了种种情况，但最后的挂载操作是由系统上的 mount 工具执行，而该工具默认解析符号链接，这就引入了 TOCTOU 问题（竞态条件问题的一种），也就是近来曝光的 CVE-2021-25741。

本文将对这两个漏洞进行关联分析。下文的组织结构如下：

(1) 给出理解漏洞的必要背景知识；

(2) 剖析、复现 CVE-2017-1002101 漏洞；

(3) 剖析、复现 CVE-2021-215741 漏洞；

(4) 基于以上分析，给出绿盟科技的总结与思考。

由于 CVE-2021-25741 漏洞较新，本文仅结合公开资料对漏洞进行分析，给出脱敏复现截图，帮助大家理解这一漏洞。请勿将相关知识、技术应用于非法活动。

绿盟科技星云实验室的开源云原生靶场项目 Metarget[6] 现已支持自动化构建 CVE-2017-1002101 和 CVE-2021-25741 漏洞环境，欢迎研究者使用（后文会给出具体构建方法）。

穿越之旅即将开始，请坐稳扶好。

2. 背景知识

2.1 符号链接

符号链接，也被称作软链接，指的是这样一类文件——它们包含了指向其他文件或目录的绝对或相对路径的引用。当我们操作一个符号链接时，操作系统通常会将我们的操作自动解析为针对符号链接指向的文件或目录的操作。

在类 Unix 系统中，ln 命令能够创建一个符号链接，例如：

ln -s target_path link_path

上述命令创建了一个名为 link_path 的符号链接，它指向的目标文件为 target_path。

欲了解更多关于符号链接的内容，可以参考维基百科 [7]。

2.2 SubPath

在容器内部，本地文件通常是非持久化的。对于 Kubernetes 来说，当容器由于某种原因终止运行并被 Kubelet 重启后，非持久化的本地文件就会丢失。另外，集群中同一 Pod 内部或 Pod 间常常会有文件共享的需求。Kubernetes 提供了 Volume 资源用来解决上述问题，官方文档对 Volume 进行了详尽描述 [8]。

有时，我们需要把一个 Volume 在多处使用。volumeMounts.subPath 特性允许我们在挂载时指定某 Volume 内的子路径，而非其根路径。

以经典的 LAMP Pod（Linux Apache Mysql PHP）为例，利用 subPath 特性，同一 Pod 内的 mysql 和 php 容器可共享同一 Volume site-data，但在容器内部分别挂载该 Volume 的不同子路径 mysql 和 html。

```
apiVersion: v1
kind: Pod
metadata:
  name: my-lamp-site
spec:
    containers:
    - name: mysql
      image: mysql
      env:
      - name: MYSQL_ROOT_PASSWORD
        value: "rootpasswd"
      volumeMounts:
      - mountPath: /var/lib/mysql
        name: site-data
        subPath: mysql
    - name: php
      image: php:7.0-apache
      volumeMounts:
      - mountPath: /var/www/html
        name: site-data
        subPath: html
    volumes:
    - name: site-data
      persistentVolumeClaim:
        claimName: my-lamp-site-data
```

欲了解更多关于 SubPath 的内容，可以参考官方文档 [9]。

2.3 Pod 安全策略（Pod Security Policies）

Pod 安全策略为 Pod 的创建和更新提供了细粒度的权限控制。从实现上来讲，Pod 安全策略是一种集群级资源，用于对 Pod 的安全敏感设定进行管控。

Pod Security Policy 对象定义了一系列 Pod 运行必须遵从的条件，允许管理员对 Pod 进行管控，例如：

控制的角度	字段名称
运行特权容器	privileged
使用宿主机命名空间	hostPID，hostIPC
使用宿主机的网络和端口	hostNetwork，hostPorts
Volume 类型的使用	volumes
使用宿主机文件系统	allowedHostPaths
允许使用特定的 FlexVolume 驱动	allowedFlexVolumes
分配拥有 Pod 卷的 FSGroup 账号	fsGroup
以只读方式访问根文件系统	readOnlyRootFilesystem
设置容器的用户 ID 和组 ID	runAsUser，runAsGroup，supplementalGroups
限制权限提升为 root	allowPrivilegeEscalation，defaultAllowPrivilegeEscalation
Linux 权能（Capabilities）	defaultAddCapabilities，requireDropCapabilities，allowedCapabilities
设置容器的 SELinux 上下文	seLinux
指定容器能挂载的 Proc 类型	allowedProcMountTypes
指定容器使用的 AppArmor 模板	annotations
指定容器使用的 seccomp 模板	annotations
指定容器使用的 sysctl 模板	forbiddenSysctls，allowedUnsafeSysctls

欲了解更多关于 Pod 安全策略的内容及如何启用 Pod 安全策略，可以参考官方文档 [10]。

3. CVE-2017-1002101：寒风初起

3.1 漏洞分析

在针对 CVE-2017-1002101 的分析开始之前，我们先要搞清楚一件事——这个漏洞本质上是“Linux 符号链接特性”与“Kubernetes 自身代码逻辑”两部分结合的产物。符号链接引起的问题并不新鲜，在这里它与虚拟化隔离技术碰撞出了逃逸问题，以前还曾有过在传统主机安全领域与 SUID 概念碰撞出的权限提升问题等 [11]。

言归正传，CVE-2017-1002101 漏洞是怎么产生的呢？

首先，结合源码，我们来深入了解一下创建一个 Pod 的过程中与 Volume 有关的部分。笔者采用的是 v1.9.3 版本的 Kubernetes 源码，git commit 为 d2835416544。

在一个 Pod 开始运行前，Kubernetes 需要做许多事情。首先，Kubelet 为 Pod 在宿主机上创建了一个基础目录。

```
// in pkg/kubelet/kubelet.go syncPod function
// Make data directories for the pod
if err := kl.makePodDataDirs(pod); err != nil {
    kl.recorder.Eventf(pod, v1.EventTypeWarning,
events.FailedToMakePodDataDirectories, "error making pod data directories: %v",
err)
    glog.Errorf("Unable to make pod data directories for pod %q: %v",
format.Pod(pod), err)
    return err
}
```

如果跟进看 makePodDataDirs 函数，可以发现其中就包括 Volumes 目录。

```
// in pkg/kubelet/kubelet_pods.go
// makePodDataDirs creates the dirs for the pod datas.
func (kl *Kubelet) makePodDataDirs(pod *v1.Pod) error {
  uid := pod.UID
    // ...
```

```
  if err := os.MkdirAll(kl.getPodVolumesDir(uid), 0750); err != nil && !
os.IsExist(err) {
      return err
  }
    // ...
}
```

接着，Kubelet 等待 Kubelet Volume Manager（pkg/kubelet/volumemanager）将 Pod 声明文件中声明的卷挂载到上述 Volumes 目录下。

```
// in pkg/kubelet/kubelet_pods.go
// Volume manager will not mount volumes for terminated pods
if !kl.podIsTerminated(pod) {
    // Wait for volumes to attach/mount
    if err := kl.volumeManager.WaitForAttachAndMount(pod); err != nil {
        kl.recorder.Eventf(pod, v1.EventTypeWarning, events.FailedMountVolume,
"Unable to mount volumes for pod %q: %v", format.Pod(pod), err)
        glog.Errorf("Unable to mount volumes for pod %q: %v; skipping pod",
format.Pod(pod), err)
        return err
    }
}
```

在上述工作完成后，Kubelet 需要为容器运行时（Container Runtime，简称 Runtime）生成配置文件。

```
// in pkg/kubelet/kuberuntime/kuberuntime_container.go
func (m *kubeGenericRuntimeManager) startContainer(podSandboxID string,
podSandboxConfig *runtimeapi.PodSandboxConfig, container *v1.Container, pod
*v1.Pod, podStatus *kubecontainer.PodStatus, pullSecrets []v1.Secret, podIP
string) (string, error) {
    // ...
  containerConfig, err := m.generateContainerConfig(container, pod,
restartCount, podIP, imageRef)
    // ...
```

其中核心函数 generateContainerConfig 最终追溯到了位于 pkg/kubelet/kubelet_pods.go 中的 GenerateRunContainerOptions 函数。该函数中调用了 makeMounts 用来生成 Runtime 需要的挂载映射表。

```
// in pkg/kubelet/kubelet_pods.go GenerateRunContainerOptions function
mounts, err := makeMounts(pod, kl.getPodDir(pod.UID), container, hostname,
hostDomainName, podIP, volumes)
```

makeMounts 函数是问题关键所在，我们深入看一下。

```
// in pkg/kubelet/kubelet_pods.go
// makeMounts determines the mount points for the given container.
func makeMounts(pod *v1.Pod, podDir string, container *v1.Container, hostName,
hostDomain, podIP string, podVolumes kubecontainer.VolumeMap)
([]kubecontainer.Mount, error) {
    // ...
  mounts := []kubecontainer.Mount{}
  for _, mount := range container.VolumeMounts {
      // ...
      hostPath, err := volume.GetPath(vol.Mounter)
      if err != nil {
          return nil, err
      }
      if mount.SubPath != "" {
          if filepath.IsAbs(mount.SubPath) {
              return nil, fmt.Errorf("error SubPath `%s` must not be an
absolute path", mount.SubPath)
          }

          err = volumevalidation.ValidatePathNoBacksteps(mount.SubPath)
          if err != nil {
              return nil, fmt.Errorf("unable to provision SubPath `%s`: %v",
mount.SubPath, err)
          }
```



```
          fileinfo, err := os.Lstat(hostPath)
          if err != nil {
              return nil, err
          }
          perm := fileinfo.Mode()
          // 关键点1
          hostPath = filepath.Join(hostPath, mount.SubPath)

          if subPathExists, err := utilfile.FileOrSymlinkExists(hostPath);
err != nil {
              glog.Errorf("Could not determine if subPath %s exists; will not
attempt to change its permissions", hostPath)
          } else if !subPathExists {
              // Create the sub path now because if it's auto-created later
when referenced, it may have an
              // incorrect ownership and mode. For example, the sub path
directory must have at least g+rwx
              // when the pod specifies an fsGroup, and if the directory is
not created here, Docker will
              // later auto-create it with the incorrect mode 0750
              if err := os.MkdirAll(hostPath, perm); err != nil {
                  glog.Errorf("failed to mkdir:%s", hostPath)
                  return nil, err
              }

              // chmod the sub path because umask may have prevented us from
making the sub path with the same
              // permissions as the mounter path
              if err := os.Chmod(hostPath, perm); err != nil {
                  return nil, err
              }
          }
      }
      // ...
        // 关键点2
      mounts = append(mounts, kubecontainer.Mount{
          Name:           mount.Name,
          ContainerPath:  containerPath,
          HostPath:       hostPath,
          ReadOnly:       mount.ReadOnly,
          SELinuxRelabel: relabelVolume,
          Propagation:    propagation,
      })
  }
    // ...
  return mounts, nil
}
```

通过仔细分析可以发现，makeMounts 在生成挂载映射表时，并未单独列出 subPath 的情况。对于指定了 subPath 的挂载项，Kubelet 直接将 subPath 与 hostPath 进行简单的字符串合并，然后加入到挂载映射表（上述代码中的 mounts 变量）中。

最终，这个挂载映射表被传递给 Runtime 来创建容器。

初看，这个流程没什么问题。但是，如果我们把以下几点特征放在一起，就会发现以下问题 [12]：

(1) subPath 是 Pod 拥有者可控的；

(2) 卷是可以由同一 Pod 内不同生命周期的容器或不同 Pod 之间共享的；

(3) Kubernetes 将宿主机上的文件路径进行解析并传递给 Runtime，Runtime 将这些路径绑定挂载(bind mount) 到容器内部。

设想这样一种情况：

假如某人拥有某集群内的 Pod 创建权限，但是不能任意挂载卷（比如受到 Pod 安全策略的限制，否则就可以直接挂载宿主机目录实现逃逸），那么他先创建一个 Pod-1，在其中声明挂载 Volume-1。Pod-1 运行后，利用 Pod-1 的 shell 在 Volume-1 中创建一个指向 / 的符号链接 symlink-1。接着再创建一个 Pod-2，Pod-2 同样声明挂载 Volume-1，但是使用了 subPath 特性，指明 subPath 为 symlink-1。这样一来，基于我们前面的分析过程，Kubelet 会直接在宿主机上生成指向 hostPath+subPath 的路径传递给 Runtime。当 Pod-2 的容器运行起来后，它就会直接挂载宿主机上该符号链接指向的内容。

这就是 CVE-2017-1002101 漏洞所在。

3.2 漏洞复现

3.2.1 环境准备

首先，我们需要部署一个存在 CVE-2017-1002101 漏洞的 Kubernetes 集群，我们可以借助前言部分提到的开源靶场工具 Metarget 部署漏洞环境。在安装 Metarget 后，执行以下命令，即可部署上述集群。

./metarget cnv install cve-2017-1002101 --domestic

模拟的场景如下：

在集群中，黑客具有某命名空间下 Pod 的创建及相关权限，但是受到 Pod 安全策略的限制 [10]，在创建时如果挂载了 hostPath 类型的卷，只允许挂载某些非重要路径下的目录或文件，例如 /tmp。这样一来，黑客很难通过挂载宿主机敏感目录的方式实现容器逃逸。但是借助 CVE-2017-1002101，黑客能够绕过此限制，成功挂载宿主机敏感目录，继而实现容器逃逸。

接着，我们需要布置一下入侵场景。场景很简单——为集群设置 Pod 安全策略，只允许 Pod 在创建时挂载宿主机 /tmp 路径下的目录或文件。结合官方文档 [10] 及网上技术分享 [13][14]，首先创建策略：

```
apiVersion: extensions/v1beta1
kind: PodSecurityPolicy
metadata:
  name: privileged
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  allowedHostPaths:
  - pathPrefix: /tmp/
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
```

接着打通 RBAC：


```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
 name: privileged-psp
rules:
 - apiGroups:
   - policy
   resourceNames:
   - privileged
   resources:
   - podsecuritypolicies
   verbs:
   - use
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
 name: kube-system-psp
 namespace: kube-system
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: ClusterRole
 name: privileged-psp
subjects:
 - apiGroup: rbac.authorization.k8s.io
   kind: Group
   name: system:nodes
 - apiGroup: rbac.authorization.k8s.io
   kind: Group
   name: system:serviceaccounts:kube-system
```

然后为 API Server 配置 Pod Security Policy 插件。编辑 API Server 的配置文件（一般是 /etc/kubernetes/manifests/kube-apiserver.yaml），在 --admission-control 命令行选项后加上 Pod Security Policy，然后等待 API Server 重启服务(如果长时间没有重启可以尝试手动执行 service kubelet restart 重启 Kubelet 服务)，直到能够看到 API Server 进程启动参数中包含 Pod Security Policy。

```
root# ps aux | grep kube-apiserver | grep -v grep
root     26141  4.5 12.9 377460 264384 ?       Ssl  11:51  11:37 kube-apiserver
--tls-private-key-file=/etc/kubernetes/pki/apiserver.key --proxy-client-cert-
file=/etc/kubernetes/pki/front-proxy-client.crt --proxy-client-key-file=/etc/
kubernetes/pki/front-proxy-client.key --enable-bootstrap-token-auth=true --
service-cluster-ip-range=10.96.0.0/12 --tls-cert-file=/etc/kubernetes/pki/
apiserver.crt --client-ca-file=/etc/kubernetes/pki/ca.crt --kubelet-client-
key=/etc/kubernetes/pki/apiserver-kubelet-client.key --requestheader-client-ca-
file=/etc/kubernetes/pki/front-proxy-ca.crt --insecure-port=0 --allow-
privileged=true --requestheader-group-headers=X-Remote-Group --service-account-
key-file=/etc/kubernetes/pki/sa.pub --kubelet-client-certificate=/etc/
kubernetes/pki/apiserver-kubelet-client.crt --requestheader-username-headers=X-
Remote-User --requestheader-extra-headers-prefix=X-Remote-Extra- --
requestheader-allowed-names=front-proxy-client --secure-port=6443 --admission-
control=Initializers,NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStora
geClass,DefaultTolerationSeconds,NodeRestriction,ResourceQuota,PodSecurityPolic
y --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname --advertise-
address=xxx.xxx.xxx.xxx --authorization-mode=Node,RBAC --etcd-servers=http://
127.0.0.1:2379
```

上述输出说明 Pod 安全策略设置成功。我们来测试一下，尝试创建一个挂载宿主机根目录的 Pod：

```
root# kubectl apply -f - <<EOF
# stage-1-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - image: ubuntu
```

```
    name: test
    volumeMounts:
    - mountPath: /vuln
      name: vuln-vol
    command: ["sleep"]
    args: ["10000"]
  volumes:
  - name: vuln-vol
    hostPath:
      path: /
EOF
Error from server (Forbidden): error when creating "STDIN": pods "test" is
forbidden: unable to validate against any pod security policy:
[spec.volumes[0].hostPath.pathPrefix: Invalid value: "/": is not allowed to be
used]
```

可以发现，由于安全策略的存在，Pod 创建失败。另外，一些朋友可能会想到用相对路径 .. 来绕过，事实上 /tmp/../ 这种形式也会报错：

```
The Pod "test" is invalid:
* spec.volumes[0].hostPath.path: Invalid value: "/tmp/../": must not contain
'..'
* spec.containers[0].volumeMounts[0].name: Not found: "vuln-vol"
```

至此，环境准备完成。

3.2.2 漏洞利用

目标很明确：在文件系统层面实现容器逃逸。一旦实现了文件系统层面的容器逃逸，黑客就像是穿越了结界，很容易继续扩大战果，实施更有杀伤性的入侵。

结合前文的分析，在黑客的视角下，我们要做的事情实际非常简单：

(1) 创建一个 Pod，以 hostPath 类型挂载宿主机 /tmp/test 目录；

(2) 在上一步的 Pod 中执行命令，在宿主机 /tmp/test 目录下创建指向 / 的符号链接 xxx；

(3) 创建第二个 Pod，以 hostPath 类型挂载宿主机 /tmp/test 目录，在容器中以 subPath 类型挂载 xxx；

(4) 在第二个 Pod 的 shell 中，执行 chroot 将根目录切换到 xxx，实现容器逃逸。

3.2.3 注意事项

在实践过程中我们发现，为了顺利复现漏洞，需要注意：

(1) 前后创建的两个 Pod 要在同一个宿主机节点上（如果是多节点集群环境）；

(2) 不同版本 Kubernetes 环境下 Admission Controller 的 Pod Security Policy 插件的配置方式有一些小差异，具体步骤请参考官方文档。

3.3 漏洞修复

v1.9.x 系列的 Kubernetes 在 v1.9.4 版本中修复了 CVE-2017-1002101 漏洞 [15]。

漏洞的根源在于，subPath 指向的宿主机文件系统路径不受控，在符号链接的辅助下，可以是任何位置。

修复方案需要考虑两点：

(1) 解析后的文件系统路径必须是在 Pod 基础路径之内；

(2) 在检查环节和绑定挂载环节之间不允许用户更改（避免引入 TOCTOU 问题 [16]）。

Kubernetes 产品安全团队曾提出了几种不同版本的安全方案 [12]，这些方案能帮助我们更好地理解即将出场的 CVE-2021-25741 漏洞的成因。接下来，我们一起来解读一下这些方案。

3.3.1 方案一（基础方案）

基础方案是：

(1) 在宿主机上对所有的 subPath 解析符号链接；

(2) 判断符号链接解析后的指向目标是否位于卷内部；

(3) 只把第 (2) 步中判定为卷内部的解析后路径传递给 Runtime。

这个方案很简单，但是存在 TOCTOU 的风险 [16]。黑客可以先给一个合法符号链接，使第 (2) 步判断通过，再将其替换为恶意符号链接即可。因此，如果要采取这个思路，就需要为目标路径加上某种形式的锁，避免其在第 (2) 步和第 (3) 步之间被黑客更改。

后续的所有方案都采用一种临时绑定挂载的方式去实现上述“锁”的概念，这基于绑定挂载的特性——绑定挂载生效后，挂载源就不可改变了。

3.3.2 方案二

方案二在方案一的基础上做了加强：

(1) 在 Kubelet 的 Pod 目录下创建一个子目录，比如 dir1；

(2) 将卷绑定挂载到上述子目录中，比如挂载点为 dir1/volume；

(3) 使用 chroot 切换根目录到 dir1；

(4) 在切换后的根目录内，将 volume/subpath 绑定挂载为 subpath。这样一来，任何符号链接都是在 chroot 后的环境中解析；

(5) 退出 chroot 环境；

(6) 在宿主机上，将经过绑定挂载的 dir1/subPath 传递给 Runtime。

这种方案有效，但完整实现过于复杂，官方团队没有采用。

3.3.3 方案三

将方案一和方案二进行了整合：

(1) 将 subPath 路径绑定挂载到 Kubelet 的 Pod 目录下的一个子目录；

(2) 判断绑定挂载的挂载源是否位于卷内部；

(3) 只把第 (2) 步中判定为卷内部的绑定挂载传递给 Runtime。

这个方案看起来有效、简单，但是第(2) 步实际上非常难实现，因为现实中要考虑的情况实在太多了（比如 Volume 类型差异带来的影响）。

3.3.4 最终解决方案

最终，安全团队针对 CVE-2017-1002101 给出的修复方案是：

(1) 在宿主机上对所有的 subPath 解析符号链接；

(2) 对解析后的路径，从卷的根路径开始，使用 openat( ) 系统调用依次打开每一个路径段（即路径被分割符 / 分开的各部分），在

这个过程中禁用符号链接。对于每个段，确保当前路径位于在卷内部；

(3) 将 /proc/<kubelet pid>/fd/<final fd> 绑定挂载到 Kubelet 的 Pod 目录下的一个子目录。该文件是指向打开文件的链接（文件描述符）。如果源文件在被 Kubelet 打开的时候被替换，那么链接依然指向原始文件；

(4) 关闭文件描述符 fd，将绑定挂载传递给 Runtime。

详细方案讨论见官方博客 [12]。实际的修复代码过多，限于篇幅，这里不再论述。

我们在新版本的 Kubernetes 集群中重试前文的漏洞利用步骤，发现 stage-2-container 将无法创建成功：

```
root# kubectl get pods
NAME                READY   STATUS                       RESTARTS   AGE
stage-1-container   1/1     Running                      0          110s
stage-2-container   0/1     CreateContainerConfigError   0          17s
```

此时，stage-2-container 的事件日志如下：

```
root# kubectl describe -n test pods stage-2-container | tail -n 7
Events:
  Type     Reason     Age                  From                  Message
  ----     ------     ----                 ----                  -------
  Normal   Scheduled  2m59s                default-scheduler     Successfully
assigned test/stage-2-container to ctnsec-master
  Normal   Pulled     26s (x7 over 2m50s)  kubelet, ctnsec-master  Successfully
pulled image "ubuntu"
  Warning  Failed     26s (x7 over 2m50s)  kubelet, ctnsec-master  Error:
failed to prepare subPath for volumeMount "vuln-vol" of container "stage-2-
container"
```

从 Kubelet 的日志中，我们能够查看到更详细的信息：

failed to prepare subPath for volumeMount "vuln-vol" of container "stage-2-container": subpath "/" not within volume path "/tmp/test"

可以看到，日志明确指出了 / 路径并不在 /tmp/test 路径下，因此 Pod 建立失败。

最终方案看似完美无缺。然而，一个未曾考虑到的特性让安全团队为避免 TOCTOU 问题做出的以上所有复杂设计如千里长堤般溃于蚁穴。四年之后，CVE-2021-25741 出场。

4. CVE-2021-25741：百密一疏

4.1 漏洞分析

CVE-2021-25741 漏洞的成因与 CVE-2017-1002101 漏洞的最终修复方案密切相关。因此，如果您对上一节的最终修复方案只是匆匆略过，并希望明白 CVE-2021-25741 的原理，建议再回过头了解 CVE-2017-1002101 到底是如何修复的。

事实上，CVE-2017-1002101 漏洞的最终修复方案的确达到了预期目的——确保挂载路径位于卷内部，同时避免竞态条件入侵。我们结合 1.17.1 版本的 Kubernetes 代码简单看一下是怎么做的（如前所述，所用代码过多，不再赘述）。在 subpath_linux.go 中：

```
func doBindSubPath(mounter mount.Interface, subpath Subpath) (hostPath string,
err error) {
  // 1. 在宿主机上对所有的subPath解析符号链接
    newVolumePath, err := filepath.EvalSymlinks(subpath.VolumePath)
    if err != nil // ... 出错返回
  newPath, err := filepath.EvalSymlinks(subpath.Path)
  if err != nil // ... 出错返回
    // ... 省略
    // 2. 依次打开每一个路径段，确保当前路径位于在卷内部
  fd, err := safeOpenSubPath(mounter, subpath)
  if err != nil // ... 出错返回
    // ... 省略

  kubeletPid := os.Getpid()
  mountSource := fmt.Sprintf("/proc/%d/fd/%v", kubeletPid, fd)
  // Do the bind mount
  options := []string{"bind"}
  klog.V(5).Infof("bind mounting %q at %q", mountSource, bindPathTarget)
    // 3. 绑定挂载subPath到Pod内
  if err = mounter.Mount(mountSource, bindPathTarget, "" /*fstype*/, options);
err != nil // ... 出错返回
  // ... 省略
}
```

以上就是修复方案给出的步骤。新的问题到底在哪儿呢？

在 mounter.Mount 上。该函数会调用 doMount 函数，doMount 函数最终是通过执行系统上的 mount 工具来实现挂载的：

command := exec.Command(mountCmd, mountArgs...)

然而，根据 Linux 手册 [17]，mount 工具默认情况下是解析符号链接的。因此，虽然前述补丁过程中黑客无法做些什么，但他可以在 mount 工具解析符号链接后和挂载操作执行前制造竞态条件入侵，从而绕过前述补丁的防御措施。

4.2 漏洞复现

在特定的环境下，一旦成功触发漏洞，黑客能够实现容器逃逸，如下图所示：

注：Metarget 已经支持 CVE-2021-25741 漏洞环境搭建。在安装 Metarget 后，执行以下命令，即可部署存在漏洞的 Kubernetes 集群：

./metarget cnv install cve-2021-25741 --domestic

4.3 漏洞修复

这一次的修复 [18] 很简单，在调用 mount 时传递了 --no-canonicalize 参数，命令 mount 不再解析符号链接。

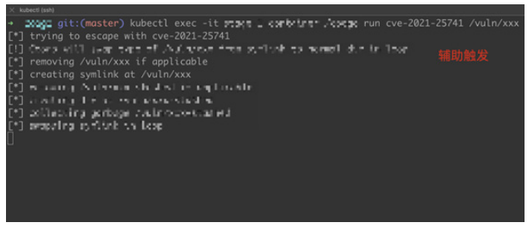
5. 结语

CVE-2017-1002101 和 CVE-2021-25741 都是符号链接处理不当引起的安全问题。事实上，符号链接引起的安全问题并不少见。我们曾不止一次提到过，成熟复杂系统（譬如 Linux）的魅力在于其能够提供强大的功能和机制，而问题则往往出现在这些功能与机制同时或交替生效的场景中。

思路再拓展一下：Windows 上的“快捷方式”与 Linux 上的符号链接的功能非常相像。而“快捷方式”也曾被曝出许多严重安全漏洞。例如，CVE-2010-2568——Windows 快捷方式文件存在缺陷导致的任意代码执行漏洞，据称曾被应用在针对伊朗核设施的“震网病毒”[19] [20] 中；再如 CVE-2017-8464——另一基于 Windows 快捷方式的任意代码执行漏洞，由于其漏洞原理上与 CVE-2010-2568 的相似性，被戏称为“震网三代”。

在云计算世界，我们尤其擅长将各种基础机制打包起来，创造出新的事物，这种新事物也许能够极大地提高生产力，甚至促进产业变革——容器便是典例。然而，结合前文所述，这也意味着以往不曾出现过的机制交叠带来的逻辑漏洞或许会在云环境陆续产生。

云原生时代，安全不可缺席。我们将持续输出云原生安全研究成果，最新成果直接赋能绿盟科技云原生安全产品 NCSS-C，为您的云原生业务保驾护航。



最后，由绿盟科技星云实验室编写的《云原生安全：攻防实践与体系构建》一书已经出版，该书从云原生技术的风险分析入手，介绍了云原生安全防护思路和体系以及云原生的可观测性，并重点介绍容器基础设施安全、容器编排平台安全与云原生应用安全。如果读者想要了解更多、更深入的云原生相关漏洞知识，可以参阅该书的第二部分。欢迎您对书的内容提出宝贵建议，并与我们进行交流。

参考文献

[1] https://seclists.org/oss-sec/2021/q3/172.

[2] https://nvd.nist.gov/vuln/detail/CVE-2021-25741.

[3] https://github.com/kubernetes/kubernetes/issues/104980.

[4] https://nvd.nist.gov/vuln/detail/cve-2017-1002101.

[5] https://github.com/kubernetes/kubernetes/issues/60813.

[6] https://github.com/Metarget/metarget.

[7] https://en.wikipedia.org/wiki/Symbolic_link.

[8] https://kubernetes.io/docs/concepts/storage/volumes/.

[9] https://kubernetes.io/docs/concepts/storage/volumes/#using-subpath.

[10] https://kubernetes.io/docs/concepts/policy/pod-security-policy/.

[11] https://en.wikipedia.org/wiki/Symlink_race.

[12] https://kubernetes.io/blog/2018/04/04/fixing-subpath-volume-vulnerability/.

[13] https://medium.com/@makocchi/kubernetes-cve-2017-1002101-en-5a30bf701a3e.

[14] https://stackoverflow.com/questions/59054407/how-to-enable-admission-controller-plugin-on-k8s-where-api-server-is-deployed-as.

[15]https://github.com/kubernetes/kubernetes/pull/61045/commits/16caae31f9e1c4dc74158a9aa79dbce177122c7e.

[16] https://en.wikipedia.org/wiki/Time-of-check_to_time-of-use.

[17] https://man7.org/linux/man-pages/man8/mount.8.html.

[18]https://github.com/kubernetes/kubernetes/pull/104253/commits/296b30f14367a42d43f25ad0774d10be55b49f4d.

[19] https://en.wikipedia.org/wiki/Stuxnet.

[20] https://www.cs.utexas.edu/~shmat/courses/cs361s/stuxnet.pdf.

人工智能赋能网络靶场创新发展

绿盟科技 安全管理产品部 孙翔

摘要：近年来，网络空间对抗形势日趋严峻，世界各国均高度重视网络靶场的建设。人工智能作为新一轮产业变革的核心驱动力，可以推动网络靶场技术升级，实现网络靶场能力的整体提升。本文主要对人工智能在网络靶场领域的应用创新方向进行探讨。

关键词：人工智能 网络靶场 创新

1. 前言

随着物联网、大数据、并行计算和深度学习算法等技术的突破，人工智能近年来取得了突飞猛进的发展，在智能机器人、无人驾驶、图像语音识别等众多领域展现出令人期待的发展前景，并得到了广泛关注。网络靶场作为网络空间安全研究、学习、测试、验证、演练等必不可少的重要基础设施，借助人工智能技术能够实现产品的快速创新发展。下面，我们将简要介绍人工智能赋能靶场的多个优势。

2. “人工智能 + 网络编排” —— 更快的仿真环境构建

网络靶场主要使用虚拟化和网络编排等技术来构建各类业务仿真环境。其中网络编排通常分为高层业务编排和底层网络资源编排。业务编排和网络资源编排相互依赖，形成一套闭环的网络编排系统，实现自动化、定制化环境构建。已经有很多研究表明人工智能在 SDN 管理（网络资源）、NFV 编排（业务）等方面都表现出优异的性能。通过人工智能技术和网络编排的结合，可构建人工智能的网络编排架构，由“自动化”向“智能化”的转变，实现高效、稳定、快速的目标网络环境仿真。

某人工智能编排架构如图 1 所示，可见现有的网络编排系统大多采用设计态和运行态分离的架构，实现了闭环自动化。为了将人工智能技术应用到网络编排中，实现从“自动化”到“智能化”的转变，在设计态和运行态架构的基础上，设计了基于人工智能的编排系统架构 [1]。

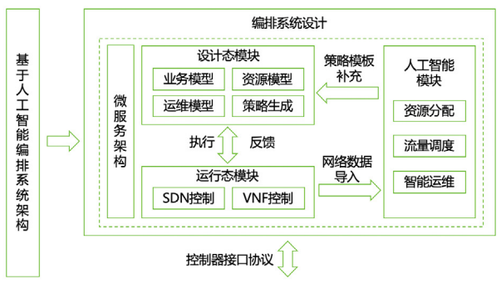


图 1 某人工智能编排系统架构

3. “人工智能 + 自动化渗透” —— 更高的安全测试效率

网络靶场在仿真环境构建完成后，还需要仿真攻防行为来对目标网络进行渗透测试，发现安全问题进而提高真实环境的安全性。现有人工渗透方式对人的能力要求较高，在交付方式、工作效率、标准化程度和数据可控性等方面都有很多不足。而借助人工智能技术可以将白帽子在大量渗透过程中积累的实战经验转化为机器可存储、识别、处理的结构化经验，并且在自动化测试过程中借助人工智能算法不断进行“智力”成长和逻辑推理决策，以贴近实际人工渗透的方式，对靶场指定目标进行从信息收集到漏洞利用的完整测试过程，从而提高安全测试的效率和准确性。

例如某人工智能漏洞挖掘研究路线如图 2 所示，第一步，从公共漏洞库中广泛搜集漏洞样本，基于补丁信息和漏洞已知信息来半自动化地标注数据，建立漏洞样本数据库。第二步，提取大量漏洞代码的特征。利用程序切片技术去除无关代码后，利用三种方式提取特征，分别为基于敏感函数切出代码片段、转换成中间代码提取控制流图、运用 Pin 提取指令序列。第三步，运用相关算法将特征转化成向量。针对控制流图，采用 strcture2vec 算法，将结构图的特征特性抽象成向量；针对源代码、中间代码和指令序列，使用词向量模型 word2vec，将“词”转化成向量。第四步，运用相应的深度学习算法进行特征的学习与判断。由于代码属于序列模型，故采用递归神经网络来学习特征 [2]。

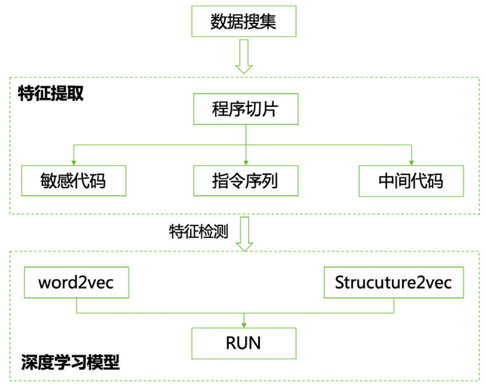


图 2 某人工智能漏洞挖掘研究路线

4. “人工智能 + 效果评估” —— 更准确的效果评估

网络靶场在对攻防行为进行评估的时候主要涉及预测态势算法和指标体系两部分。预测态势主要是指利用感知系统对当前信息现状的调查，对所预测内容的主要有关因素进行分析，并结合历史资料、预测经验以及科学的方法理论，对未来一定时期内可能出现的安全态势变化进行预测。基于人工智能技术，可以在专家系统预测和人工神经网络预测两种方法上提升预测准确率。在指标体系方面，人工智能可以基于基础

运行指标、网络威胁指标和网络脆弱性指标的检测结果为态势感知提供大量参考，既能够将效果评估难度降低，又能够更加直接地显示入侵效果。基于人工智能，利用人工智能高效的信息收集与处理能力以及高精度的判断能力，能够实现更准确的效果评估。

例如某基于人工智能的信息网络安全态势感知具体技术，如图3所示。

(1) 数据采集阶段，对防火墙日志、Web 服务日志等信息进行采集能够为态势分析提供基础数据，要求是所收集数据能够被系统所识别并借助云服务器实现数据更新。主要技术包括：端口匹配技术、流量特征检测技术、自动连接关联技术。

(2) 数据预处理阶段，由于此感知技术基于人工智能而发展，所以能够运用大数据对所采集信息进行预处理，降低数据的后续处理难度。此技术主要运用了大数据技术中的 Stream 框架，此框架具备数据处理速度较快、扩展性较高与并发处理能力较强的优势。在具体的预处理活动中，将涉及内容数据归一、情报知识库的关联和数据归并。

(3) 数据存储与检索阶段，由于信息网络中所存储的信息量异常庞大，所以系统在对大量数据进行检索时一般可以借助搜索引擎来完成，比如 Elastic Search 引擎，此搜索引擎能够实现分布式全文搜索，与企业内的云计算环境非常契合。具体搜索模式为在系统平台对信息进行处理与计算等操作后，将数据保存至分布式搜索引擎的索引文件中，再将各类型数据以时间、名称、内容等为标准进行分类存储，并提供出索引字段，以提供数据快速检索功能。另外，将索引以多个分片和多个副本的形式存储于分布式文件系统中，既能够有效实现对近期录入数据的近似值查询，通过相类似信息佐证所查询信息的真实性，保障数据可靠性，又能够使系统中的 TB 级数据索引时间缩短至秒级，大幅度提升索引性能。

(4) 在经过上述阶段处理后，仍旧需要通过多种技术对数据进行深入分析与挖掘，发现其中的潜在风险。主要应用技术为：恶意代码智能检测技术、广谱反病毒查杀技术、机器学习技术、自动化数据处理技术。

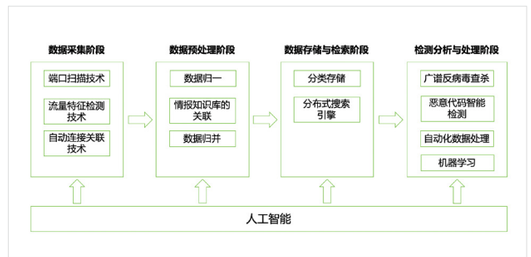


图 3 人工智能态势感知涉及的技术

5. “人工智能 + 决策平台” —— 更强的防御体系

网络靶场的目标之一就是复现企业网络环境，支持各角色用户在此环境上进行攻防实战，进而制定企业的安全防御体系或进行优化。而通过与人工智能技术的结合可搭建智能决策平台，在资源有限的情况下，针对遇到的安全问题，输出处置的行动方案，再转成脚本化的策略，下发到防护设备上。智能决策平台构建通用的 AI 协同决策支持系统框架，支持安全专家或其他系统的信息输入，并根据专家系统或其他系统的输入的变化进行决策的调准和优化，提供有效人机协同能力，实现闭环决策。

例如，如图 4 所示，某智能决策平台架构流程如下：底部的网络靶场通过仿真入侵设备、终端靶机、网络设备和安全设备四类设备的多个节点，构建完整的虚拟网络拓扑。其中入侵设备类型节点接收入侵调度引擎的指令，安全设备类型节点接收策略编排引擎的调度，网络设备通过捕获虚拟网络拓扑的流量向多个引擎发送相应日志。智能决策平台通过威胁情报平台更新知识和规则，使用入侵调度引擎让入侵设备在虚拟拓扑内执行入侵操作，接收各个引擎的日志分析执行效果，再使用策略编排引擎让安全设备执行防护策略，通过引擎日志验证防御效果 [3]。

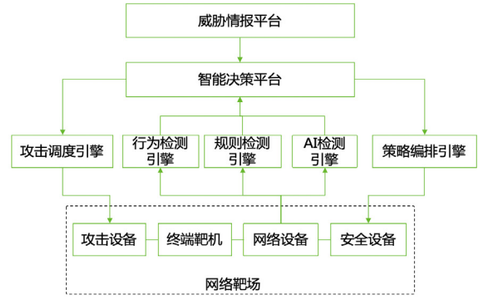


图 4 某智能决策平台架构

6. 结语

赋能人工智能安全是绿盟科技在技术创新和产品研发中一以贯之的创新构想和实践方向。绿盟科技作为深耕网络安全二十余载的安全企业，拥有由众多优秀的专家组成的独立安全研究机构，致力于跟踪国内外最新网络靶场研究方向，为用户打造更加智能的网络靶场产品。

参考文献

[1] 陈天骄，刘江，黄韬 . 人工智能在网络编排系统中的应用 [J]. 电信科学，2019, 35(5):9-16.

[2] 夏之阳 . 基于人工智能的漏洞挖掘 [D]. 上海交通大学，2019.

[3] 胡庆伟 . 对基于人工智能的信息网络安全态势感知技术分析 [J]. 网络安全技术与应用，2020(5):149-150.

个人信息安全法律保护伞

——《中华人民共和国个人信息保护法》解读

绿盟科技 咨询设计部 曾令平

摘要：随着《中华人民共和国个人信息保护法》（以下简称《个人信息保护法》）在 2021 年 8 月 20 日颁发，并于 2021 年 11 月 1 日起正式施行，全社会正掀起一场以“个人信息保护”为主题的保卫战。作为企业，在这场保卫战中该如何履行责任与义务，如何保护和利用个人信息；作为公民，又当如何维护自身合法权利？由此，本文将从《个人信息保护法》的背景介绍、具体内容、问题思考、关注要点等多方面进行解读，梳理整部法律的个人信息保护框架内容，包括对个人信息处理规则、个人信息跨境传输、个人信息处理活动的权利、信息处理者的义务、监管部门职责以及罚则等，同时也提出了个人信息保护工作落地建设路径。

关键词：个人信息保护法 信息泄露 信息处理

1. 背景介绍

1.1 发布背景

数据经济潜力不断突显，同时也产生了一系列的安全问题，如 2017 年“支付宝年度账单事件”、2020 年“圆通内鬼泄露 40 万条个人信息事件”等与公民个人信息保护相关的泄露事件和诉讼案件开始频繁涌现。与此同时，数据垄断、数据滥用、数据权属、数据流通等新型问题也进入研究和管理视野。这些与公民个人信息保护相关的议题或事件，不断成为舆论热点。而最新数据显示：2020 年中国网民总体规模已占全球网民规模的五分之一，2020 年中国网民规模为 9.89 亿人。而互联网网站 443 万个，手机应用程序数量为 302 万款。2021 年以来，国家网信办对地图导航、运动健身、短视频等十多种类型的手机应用程序进行了检测。351 款 APP 因违法收集个人信息被通报，25 款因严重违法违规收集使用个人信息被下架。

“为及时回应广大人民群众的呼声和期待，落实党中央部署要求，制定一部个人信息保护方面的专门法律，将广大人民群众的个人信息权益实现好、维护好、发展好，具有重要意义。”全国人大常委会法工委副主任刘俊臣表示，制定个人信息保护法是进一步加强个人信息保护法制保障的客观要求，是维护网络空间良好生态的现实需要，也是促进数字经济健康发展的重要举措。

从现已颁布的法律来看，虽有部分内容与个人信息保护相关，但在社会实践中，这些法律的适用大多规定得较为原则性，并不能满足公民对个人信息保护的各类迫切需求。此外，综观其他法规及规范性文件，例如《关于加强网络信息保护的决定》《电信和互联网用户个人信息保护规定》《信息安全技术 个人信息安全规范》（GB/T 35273-2020）等规定，虽然在司法案例中起着极强的合规参考作用，但其同时也存在着一定的滞后性，并不能够适应各行业、各类企业的合规需要。其实，早在 2001 年，国家成立国家信息化领导小组，下设国务院信息化工作办公室，主要负责推动国家的信息化相关立法。受国信办委托，由周汉华老师领衔的个人数据保护法研究课题组承担《个人数据保护法》比较研究课题并草拟专家建议稿。接下来，将通过介绍《个人信息保护法》的发展历程进行详细讲解。

1.2 发展历程

自 2003 年起，我国就启动了保护个人信息的立法程序。经过了十几年不断摸索，个人信息保护立法才逐渐趋于完善。以下从几个重要时间节点进一步说明：

- 2003年，《个人信息保护法》专家建议稿开始起草，2005年初已经完成；
- 2009年，《刑法修正案(七)》第7条将非法提供与获取公民个人信息行为纳入刑法规制；
- 2013年，《电信和互联网用户个人信息保护规定》对“公民个人电子信息”做了界定，并明确了信息收集、使用的原则和相关规则；
- 2017年，《网络安全法》的实施，对“公民个人信息”进一步界定、对用户“知情同意”做出明确规定、对“网络运营者”提出明确要求；
- 2020年，《民法典》强调“以人为本”，加大了对公民隐私权和个人信息的保护力度；
- 2020年6月，全国人大常委会调整2020年度立法工作计划，个人信息保护法草案将提请审议。

《个人信息保护法》在 2018 年被列入全国人大常委会未来五年任期的立法议程中，经历了从 2020 年初次评审到 2021 年的二审、三审，《个人信息保护法》的具体内容也不断发生变化。

1.3 法律地图

本文从国家法律、行政法规、司法解释、部门规章、技术规范五个层面入手，梳理国内数据安全与个人信息保护相关制度，整理形成可直观查看的“中国数据新秩序的法律地图”，如图 1 所示。

国内安全工作坚持总体国家安全观，在不同领域均有相关文件指导安全工作。其中与数据安全和个人信息保护领域相关性较强的有：民事领域通过了《民法典》；在网络空间安全领域，有《网络安全法》、等保 2.0 系列标准、《网络安全审查办法》等；在数据安全领域，有刚刚出台的《数据安全法》；在个人信息保护领域，有《个人信息保护法》。在儿童个人信息、密码、网络犯罪、消费者权益保护、电子商务等领域也有专门立法。总体来说，《网络安全法》《数据安全法》与《个人信息保护法》在立法定位上坚持总体国家安全观，共同构成了我国数据新秩序下的“三驾马车”。

图 1 中国数据新秩序的法律地图

2. 内容解读

2.1 概述

在信息化时代，个人信息保护已成为广大人民群众最关心、最直接、最现实的利益问题之一。《个人信息保护法》坚持和贯彻以人民为中心的法治理念，牢牢把握保护人民群众个人信息权益的立法定位，聚焦个人信息保护领域的突出问题和人民群众的重大关切。

全文共八章七十四条，明确了法律适用范围，聚焦目前个人信息保护的突出问题，在有关法律的基础上，该法进一步细化、完善个人信息保护应遵循的原则和个人信息处理规则，明确个人信息处理活动中的权利义务边界，健全个人信息保护工作机制。确立以“告知—同意”为核心的个人信息处理规则，落实国家机关保护责任，加大对违法行为的惩处力度。

2.2 七大关键点

2.2.1 术语界定

《个人信息保护法》规定了三个术语定义和四个相关用语的含义，本文仅对“个人信息”“敏感个人信息”“个人信息的处理”和“自动化决策”的定义或含义做进一步解读：

- “个人信息”，其定义采取的是“识别”的方式，仅采取定义式的规定方式，已发布的《个人信息安全规范》进行了不完全列举。随着数据经济不断发展，本文大胆预测，有关个人信息的定义和范围也将再次被延申。
- “敏感个人信息”，该说法与《个人信息安全规范》中“个人敏感信息”措辞不同但所表示的内容基本一致，只不过本法中的“敏感个人信息”更加强调了“人格尊严”。值得关注的是，本法中也对列举的信息做了新增和调整：生物识别、宗教信仰、特定身份、医疗健康、金融账户、行踪轨迹等信息，以及不满十四周岁未成年人的个人信息。
- “个人信息的处理”，相较于一审稿中，主要变化在于删除了“活动”，强调了个人信息的处理动作或场景。
- “自动化决策”，主要变化在于主谓宾的顺序调整，强调了人工智能技术的重要应用对公民个人信息权益的影响。

2.2.2 适用范围

本法明确了“我国境内”和“境外管辖”两大适用范围，“境外管辖”同等回应了欧盟 GDPR、美国 CCPA 等国外立法的长臂管辖效力。

- 我国境内：在中华人民共和国境内处理自然人个人信息的活动，适用本法。
- 境外管辖：在中华人民共和国境外处理中华人民共和国境内自然人个人信息的活动，有下列情形之一的，也适用本法：

以向境内自然人提供产品或者服务为目的；

分析、评估境内自然人的行为；

法律、行政法规规定的其他情形。

2.2.3 基本原则

在“第一章 总则”部分，进一步明确了处理个人信息的基本原则，本文参考相关法律法规，结合企业实践，总结了以下六大基本原则：合法正当诚信原则、目的明确原则、最小必要原则、公开透明原则、信息准确原则、安全保障原则，如图 2 所示。

图 2 个人信息处理六大基本原则

2.2.4 “点”“面”“球”生态融合

本文从“点”“面”“球”构建个人信息保护生态融合体系，以达到相互影响、相互制约、相互信任、不断演变，并在一定时期内处于相对稳定的动态平衡状态。

- “点”：指全民守护，坚守基本底线。

任何组织、个人不得非法收集、使用、加工、传输他人个人信息，不得非法买卖、提供或者公开他人个人信息，不得从事危害国家安全、公共利益的个人信息处理活动。

- “面”：指共同参与，建设良性生态。

国家建立健全个人信息保护制度，预防和惩治侵害个人信息权益的行为，加强个人信息保护宣传教育，推动形成政府、企业、相关社会组织、公众共同参与个人信息保护的良好环境。

- “球”：指国际合作，推进生态互融。

国家积极参与个人信息保护国际规则的制定，促进个人信息保护方面的国际交流与合作，推动与其他国家、地区、国际组织之间的个人信息保护规则、标准等互认。

2.2.5 处理规则

个人信息处理规则：包括了一般规定、敏感个人信息的处理规则、国家机关处理个人信息的特别规定三个方面。需要注意的是，本法确立以“告知—同意”为核心的个人信息处理规则，同时也新增了同意的例外事由，如《个人信息保护法》第十三条中提到的“前款第二项至第七项规定情形的，不需取得个人同意”，如图 3 所示。

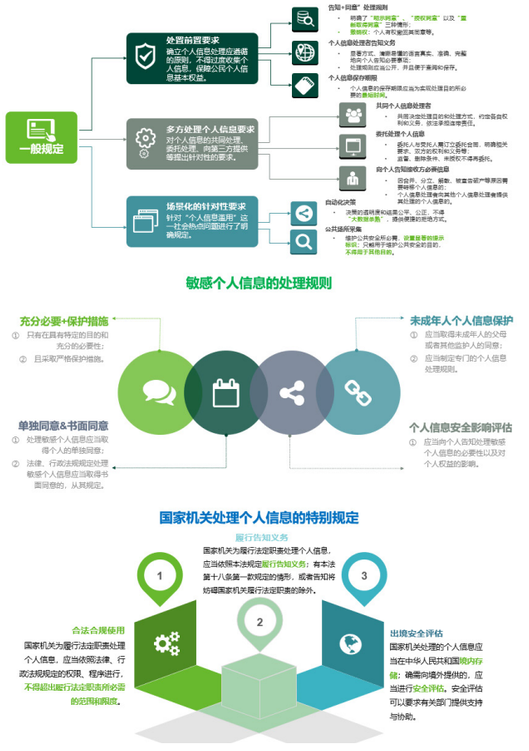


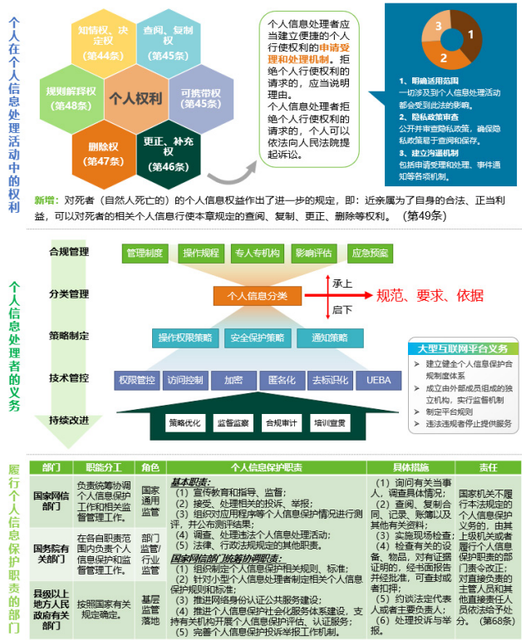
图 3 个人信息处理规则

个人信息跨境提供的规则：本法构建了一套清晰、系统的个人信息跨境流动规则，以满足保障个人信息权益和安全的客观要求，适应国际经贸往来的现实需要。关于跨境提供场景下的规则要求如图 4 所示。

图 4 个人信息跨境提供规则

2.2.6 相关主体

本法涉及个人、个人信息处理者、监管部门三大强相关的主体，如图 5 所示，分别就个人权利、个人信息处理者的义务、监管部门所履行的个人信息保护职责进行阐述。



部门	职能分工	角色	个人信息保护要素	具体措施	责任
国家网信部门	负责统筹协调个人信息保护工作和相关监督管理工作。	国家通用监管	***基本职责：*** （1）宣传教育和指导、监督； （2）接受、处理相关的投诉、举报； （3）组织对应用程序等个人信息保护情况进行测评，并公布测评结果； （4）调查、处理违法个人信息处理活动； （5）法律、行政法规规定的其他职责。 ***国家网信部门统筹协调职责：*** （1）组织制定个人信息保护相关规则、标准； （2）针对小型个人信息处理者制定相关个人信息保护规则和标准； （3）推进网络身份认证公共服务建设； （4）推进个人信息保护社会化服务体系建设，支持有关机构开展个人信息保护评估、认证服务； （5）完善个人信息保护投诉举报工作机制。	（1）询问有关当事人，调查具体情况； （2）查阅、复制合同、记录、账簿以及其他有关资料； （3）实施现场检查； （4）检查有关的设备、物品，对有证据证明的，经书面报告并经批准，可查封或者扣押； （5）约谈法定代表人或者主要负责人； （6）处理投诉与举报。	国家机关不履行本法规定的个人信息保护义务的，由其上级机关或者履行个人信息保护职责的部门责令改正，对直接负责的主管人员和其他直接责任人员依法给予处分。（第68条）
国务院有关部门	在各自职责范围内负责个人信息保护和监督管理工作。	部门监管/行业监管			
县级以上地方人民政府有关部门	按照国家有关规定确定。	基层监管落地			

图 5 三大相关主体权利、责任与义务

2.2.7 强监管和惩处力度

近年来，有关个人信息权益侵权案件逐渐增多，比如“告知—同意”的认定、人格权纠纷、人脸识别等与个人信息主体强相关的权益。因此，本法在这方面加强了监管并加大了惩处力度。本法规定了“一般的个人信息违法行为”和“情节严重的个人信息违法行为”，虽然对这两者没有严格的界定和说明，但可参照以往的司法案例或借鉴 GDPR 相关处罚案例。详细惩处要求如图 6 所示：

处罚力度	违规行为	处置情况	单位	直接负责的主管人员和其他直接责任人员
一般的个人信息违法行为	违反本法规定处理个人信息，或者处理个人信息未履行本法规定的个人信息保护义务的。	由履行个人信息保护职责的部门责令改正，给予警告，没收违法所得，对违法处理个人信息的应用程序，责令暂停或者终止提供服务。	拒不改正的：100万以下	1-10万
情节严重的个人信息违法行为	有前款规定的违法行为，情节严重的。	禁止其在一定期限内担任相关企业的董事、监事、高级管理人员和个人信息保护负责人。	五千万元或上一年度营业额5%以下	10-100万元
记入信用档案，并予以公示	有本法规定的违法行为的，依照有关法律、行政法规的规定记入信用档案，并予以公示。			
依法给予处分	国家机关不履行本法规定的个人信息保护义务的，责令改正；对直接负责的主管人员和其他直接责任人员依法给予处分。履行个人信息保护职责的部门的工作人员玩忽职守、滥用职权、徇私舞弊，尚不构成犯罪的，依法给予处分。			
民事责任	**私益诉讼：**处理个人信息侵害个人信息权益造成损害，个人信息处理者不能证明自己没有过错的，应当承担损害赔偿等侵权责任。 其通过确定的，由人民法院根据实际情况确定赔偿数额。			
	公益诉讼：个人信息处理者违反本法规定处理个人信息，侵害众多个人的权益的。 人民检察院、法律规定的消费者组织、和由国家网信部门确定的组织可以依法向人民法院提起诉讼。			
刑事责任	**刑事惩处：**违反本法规定，构成违反治安管理行为的，依法给予治安管理处罚。			
	治安处罚：违反本法规定，构成犯罪的，依法追究刑事责任。			

图 6 强监管下惩处要求

2.3 横向对比

为了便于对《个人信息保护法》进一步理解，本文通过列表的方式对国内强相关的几部法律法规进行横向对比，如图 7 所示。本文对照仅限于非法律专业视角进行对照，因此严格意义上来说，不能准确对比分析法律效力位阶的相关问题。

对比项	网络安全法	民法典	数据安全法	个人信息保护法	个人信息安全规范
个人信息定义	识别 采取了不完全列举方式。	识别 采取了不完全列举方式，扩大个人信息的范围。	对“数据”、“数据安全”等概念进行了定义，仅采取定义式的规定方式。	识别 仅采取定义式的规定方式，将匿名化的信息排除在个人信息之外。	识别+关联 采取了不完全列举方式，对个人信息和个人敏感信息进行例举。
侧重方向	方向：网络空间安全 • 维护网络空间良好生态； • 个人信息保护仅是出总体性原则和概括性要求，未进行颗粒度要求。	方向：个人信息权益 • 个人信息权益的定位其人格属性； • 确立了个人信息受法律保护的原则和设立了相关的民事权利。	方向：数据安全 • 数据领域的基础性法律，重点确立数据安全保护管理各项基本制度； • 明确规定了开展数据活动的组织和个人的责任和义务。	方向：个人信息安全 • 个人信息保护的专门法律； • 兼顾个人信息的安全和利用； • 明确边界，适用范围更为直接明确。	方向：个人信息安全 • 提供了具有可操作性的指引； • 规范了个人信息处理活动应遵循的原则和安全要求。
局限性	在社会实践中，将用户同意作为唯一的合法依据，其适用性有限且缺乏细节。	仅适用于民事领域，且未能向监管者进行个人信息保护提供实施细则或赋予其明确的权力。	立足数据安全工作实际，着力解决数据安全领域突出问题，不涉及具体个人信息保护范围。	仍采取国家主管部门监督、指导，有关部门对应执法的模式，并未确立集中监管机构。	不具有强制效力，但不排除实践中执法机构的参考借鉴。

图 7 国内法律法规横向对比

通过以上从定义、侧重方向、局限性三个方面进行横向对比后，本文得出如下参考结论：

（1）随着我国法律法规的不断完善，各行各业首先需要考虑的是“合规”问题，特别需要关注具体的、可落地的安全要求；

（2）各项法律法规间各有侧重点和存在一定的相互关联性，需要特别注意上位法的法律效力；在具体实践过程中，均须遵照执行；

（3）《数据安全法》和《个人信息保护法》都提出落实处理者的责任和义务，对企业而言，是否需要建立两套标准呢？答案是否定的，建议将其融合，组织建设、制度流程等可合二为一，人员能力、技术措施需要有针对性实施，在具体要求方面再进行细化和管控。

2.4 解读思考

2.4.1 典型问题Q&A

典型问题一：关于“告知 + 同意”。

依据：第 14 条将“充分知情”作为“同意”的前提条件，需要取得“单独同意”的情况：第 23 条、第 25 条、第 26 条、第 29 条、第 39 条。

解答：通过用户主动勾选、浏览隐私政策等获得个人信息的授权使用，并赋予用户撤回同意的权利。同时梳理“单独同意”的场景并进行对应功能调整。

典型问题二：关于生物特征等敏感个人信息。

依据：第 26 条规定的“所收集的个人图像、身份识别信息只能用于维护公共安全的目的”、第 28 条规定的“特定的目的和充分的必要性”的前提、第 29 规定的“单独同意”、第 30 条规定的“必要性以及对个人权益的影响”的告知。

解答：重视敏感个人信息的处理规则，并做好相关充分告知和影响评估等工作。

典型问题三：关于“个人信息保护负责人”。

依据：第 52 条规定“处理个人信息达到国家网信部门规定数量的个人信息处理者应当指定个人信息保护负责人”。

解答：其中“规定数量”在本法中未明确规定，但可参照《个人信息安全规范》的规定：从业人员规模大于 200 人、处理超过 100 万人的个人信息、处理超过 10 万人的个人敏感信息的。

典型问题四：关于影响评估。

依据：第 55 条规定“有下列情形之一的，个人信息处理者应当事前进行个人信息保护影响评估，并对处理情况进行记录”。

解答：结合《个人信息安全规范》和《影响评估指南》相关要求，进行个人信息安全影响评估落地执行。

以上思考的问题仅为冰山一角，建议组织结合自身实际情况，制定相应安全策略，落实个人信息保护责任。

2.4.2 主要关注点

（1）明确个人信息保护责任制，落实全生命周期管控责任。

内容：建立个人信息保护组织架构，明确岗位职责，制定对应的全流程管理规范、制度、流程等。

方案支撑：数据安全管理体系建设。

（2）通过个人信息分类（分级）管理，实现建设第一步。

内容：建立个人信息管理机制，明确保护对象及策略。

方案支撑：数据分类分级。

（3）发现企业个人信息安全隐患，降低信息泄露风险。

内容：利用风险评估手段识别发现企业的个人信息安全风险，协助企业进行整改，提升企业个人信息保护建设水平。

方案支撑：个人信息安全影响评估、APP 个人信息安全评估。

（4）识别个人信息处理活动，落实安全技术措施。

内容：梳理个人信息全生命周期处理活动，制定相对应的安全要求，对各风险点进行提示，包含可落地执行的机制等。

方案支撑：个人信息保护专项规划、数据安全管控平台。

（5）建立个人信息安全事件应急响应机制。

内容：建立个人信息安全应急预案，明确个人信息事件的应急方针、政策，应急组织结构及相关应急职责。

方案支撑：应急响应体系建设。

（6）组织开展个人信息安全培训教育。

内容：组织开展个人信息安全专业培训，提升企事业单位个人信息安全保护意识，促进个人信息安全人员专业能力提升。

方案支撑：个人信息安全专业教育培训。

（7）聚焦个人信息跨境提供，保障国家安全、公共利益及个人权益。

内容：建立个人信息跨境提供全流程管理规范、制度、流程等；明确合规路径，并征得用户的单独同意，确保个人信息安全流通。

方案支撑：遵循国家个人信息出境相关规定。

3. 结语

2021 年可谓是数据保护元年，《数据安全法》《个人信息保护法》《关键信息基础设施安全保护条例（国务院令第 745 号）》等一系列法律法规的颁布和实施，标志着我国在数据安全和个人信息保护方面正式进入 2.0 时代。而数据安全和个人信息保护密不可分，就像是一对孪生兄弟。关于个人信息保护的应对思路，可在数据安全建设的基础上进行专项设计和实施，形成个人信息保护体系的长效机制（IRCSS），如图 8 所示。该机制主要通过五个方面进行个人信息安全落地建设，帮助组织确立管理制度和操作流程，全面了解个人信息安全状况，制定个人信息安全监测与防护措施，通过优化改进与持续运营，实现持续自适应的个人信息安全防护能力。

图 8 个人信息保护体系的长效机制（IRCSS）

权利义务框架下的移动互联网APP个人信息保护

——《个人信息保护法》对APP个人信息保护影响分析

绿盟科技 战略规划部 林涛

摘要：2021 年 11 月 1 日，《个人信息保护法》正式生效施行，标志着我国个人信息保护新阶段的全面开启。当前数字经济大背景下，移动互联网应用程序（APP）的应用日益广泛，业已成为个人信息保护的重点领域之一。在新的法律框架下，开展 APP 个人信息保护应当重点关注什么？相关各方需要推进哪些工作？这些问题都值得我们深入探讨。

关键词：个人信息保护法 移动互联网 APP

1.APP 个人信息保护必要性日益突显

APP 数量随着手机用户的增加而持续高速增长。据 CNNIC 发布的第 48 次《中国互联网络发展状况统计报告》显示，截止到 2021 年 6 月，我国手机网民规模已达到 10.07 亿，网民使用手机上网的比例为 99.6%。可见，以手机为主要平台的各类 APP 面临更加广阔的流量基础和发展前景。另据工信部公布的《2020 年互联网和相关服务业运行情况》显示，截止到 2020 年底国内各类手机终端 APP 已达 345 万款。

与 APP 高速增长形成鲜明对比的是 APP 个人信息保护形势日益严峻。中国互联网协会《中国网民权益保护调查报告》显示，每年网民因个人信息泄露等原因遭受的经济损失为 800~1000 亿元。另据中国消费者协会开展的 APP 个人信息收集与隐私政策测评结果显示，在收集个人信息方面，纳入测评的 100 款 APP 普遍存在涉嫌过度收集个人信息的情况，包括“位置信息”“通讯录信息”“身份信息”“手机号码”等。

2.《个人信息保护法》确立了基本权利义务框架

为加强 APP 个人信息保护，国家先后采取了一系列举措，初步建立了相关的保护制度规范框架和监督管理机制。《个人信息保护法》的颁布实施，不仅进一步巩固了这些制度和管理成果，而且对于用户方权利、服务方义务做出了体系化提升。

2.1 用户方的四项权利

在 APP 个人信息保护的场景下，《个人信息保护法》对于个人信息权利的规定，可以大致概括为四项，即：知情权、决定权、安全权和选择权。

一是知情权。主要是指用户对与自己有关的自身个人信息和对与自身有利害关系影响的信息所享有的了解和知悉的权利，一般主要包括信息查询、复制等具体权利。《个人信息保护法》第四十五条（第一、二款）、第四十八条具体规定了个人信息查阅、复制以及要求个人信息处理者解释处理规则等具体知情权，明确了权利内容、权利行使条件和行使方式等。

二是决定权。主要是指用户对自身的个人信息处置的权利，一般主要包括发现个人信息有错误时提出异议的权利，以及要求个人信息处理者删除个人信息等具体权利。《个人信息保护法》第四十五条（第三款）、第四十七条具体规定了个人信息的指定转移权、请求删除权，并明确了该项具体权利的适用范围、行使流程等内容。

三是安全权。主要是指用户对于个人信息享有的维护信息完整、确保个人信息处于有效保护和合法利用状态的权利，一般主要包括保护信息完整权、排除风险权、消除侵害权等具体权利形式。《个人信息保护法》第四十六条具体规定了个人用户享有的要求个人信息处理者实施更正、补充的权利；此外法律还在总则部分规定了信息处理质量的要求（第八条）。

四是选择权。该权利相对于前三项权利而言是一种衍生权力，即基于个人信息权利而做出是否使用、或使用哪些 APP 服务的权利，其在权利形式上一般表现为确认权、退出权等。《个人信息保护法》对于个人用户选择权的规定，主要是通过对个人“同意”的规定来体现的，法律条款涉及第十三条、十四条、十五条、十六条，对个人同意的条件、做出方式、撤回同意的影响和效力等进行了明确。

2.2 服务方的三项义务

在 APP 个人信息保护的场景下，《个人信息保护法》通过明确规定服务提供方义务的方式，强化对个人用户权利的保护，并进一步推进理顺监管机制。法律对于服务方义务的规定，可以概括归纳为三大项：一般保护义务、特殊保护义务和基础保障义务。

一是一般保护义务。主要是指服务方承担的与用户方权利相对应的各项基本保护义务，包括对用户知情权、决定权、安全权、选择权及其各项具体权利的保护方式和制度等。《个人信息保护法》第五十一条规定了保护用户个人信息的具体五项措施要求，包括：建立内部制度和规程、实行分类管理要求、采取相应安全技术措施、从业人员定期教育培训、制定应急预案。

二是特殊保护义务。主要是指服务方承担的除了基本保护义务之外、需要额外承担的特别保护责任，主要是针对特殊用户群体、特殊保护对象或者服务方本身具有特殊影响的情况。《个人信息保护法》规定的特殊义务主要有：敏感个人信息处理义务、重要互联网平台及用户数量大和业务类型复杂的服务方义务两类。前者包括：取得书面同意、取得单独同意、进行额外告知等义务（第二十八条至第三十二条规定）；后者包括：建立外部独立监督机构、制定平台规则、发布社会责任报告等义务（第五十八条）。

三是基础保障义务。主要是指为确保实现保护义务而需要承担和开展的合规类建设职责，表现形式上一般是自行或委托第三方开展相关安全检测评估等。《个人信息保护法》规定的基础保障义务主要包括设立保护专人或专职机构、定期开展合规审计、开展前置影响评估、及时采取补救措施并报告等（第五十二条至第五十七条），法律通过明确适用主体类别、制定详细适用条件等方式，强化了此类义务的界限和可操作性。

3. 对 APP 个人信息保护影响的思考和展望

《个人信息保护法》的施行，无疑进一步提升了对 APP 用户个人信息的保护广度和深度，而其对于 APP 监管者、APP 信息处理

者、APP 合规保障者三方主体的影响则更具有全局性和深远价值。笔者将结合思考研判，分别加以简析。

首先，推进 APP 监管者优化监督管理。一是从监管主体来看，将进一步强化协同机制。《个人信息保护法》以专章 (第六章) 明确了个人信息保护的管理框架，即“国家网信部门负责统筹协调”“国务院有关部门依照本法和有关法律、行政法规的规定，在各自职责范围内负责个人信息保护和监督管理工作”，可见在 APP 个人信息保护领域，相关部门基于这一监管框架强化落实分工协同，将成为监管发展主基调。二是从监管依据来看，法规和标准规范将加速出台。在法规层面，除了此前已公开征求意见的《移动互联网应用程序个人信息保护管理暂行规定》将加速推进外，一批已生效实施的部门规章也将迎来修订完善的契机，包括《移动智能终端应用软件预置和分发管理暂行规定》等；在标准方面，与 APP 收集个人信息相关的基本要求、同意权认定方法、检测评估规范等也将成为重点立制方向。三是从监管手段来看，检查监督举措的统筹性将持续加强。当前针对 APP 个人信息保护的检查和反馈机制已初具规模且成效明显，而如何加强不同行业领域监督检查之间的统筹协调、最大限度减小被监管对象的负担、强化检查结果的互认等，或将是优化监管手段的一个重要方向。

其次，推进 APP 信息处理者强化合规落实。《个人信息保护法》对于 APP 信息处理者（包括平台运营者、分发者、智能移动终端生产企业、网络接入服务提供者等）而言，其影响将主要体现在三个方面：一是 APP 经营思路将加速收敛。法律的施行将进一步督促 APP 信息处理者加快树立保护个人信息的“红线”意识，避免过激的“唯流量论”发展思路。二是 APP 信息处理者在个人信息保护方面的建章立制将提速。受高压监管和处罚的影响，APP 服务提供者必然会强化健全法律要求的规章机制，在个人信息保护制度完善、机构人员设置、流程补足等多个方面优先发力。三是加强 APP 在个人信息保护上的技术改进和提升。改进提升 APP 自身的推荐算法、个人信息保护规则、信息核查机制等技术设置或将更具实质意义，在此方面 APP 信息处理者可选路径包括多种，如基于自身技术实力自行研发、引入第三方专业力量联合推进等。

再次，拉动 APP 合规保障市场加快创新发展。《个人信息保护法》在强化监管的同时，也为 APP 合规保障市场的发展带来新的发展动能。一是促进 APP 基础开发服务创新发展，包括为 APP 提供软件开发工具包 (SDK)、封装、加固、编译环境等，将迎来个人信息保护需求带来的更新、改进发展契机。二是促进 APP 个人信息保护检测技术和平台创新发展，检测技术将重点围绕隐私条款的内容合规性、APP 功能的个人信息需求梳理、对引用第三方嵌入工具个人信息保护的核实等。三是促进 APP 个人信息保护支撑服务创新发展，此类业务主要包括个人信息保护相关规划咨询、个人信息保护合规审计、个人信息保护影响评估、个人信息保护风险监测、个人信息保护教育培训等。

总之，加强 APP 个人信息保护是个人信息保护领域至关重要的一环，在当前推进数字经济发展的进程中具有十分直接的现实意义。在《个人信息保护法》生效施行之际，绿盟科技将依托自身在个人信息保护方面的丰富服务实践和技术产品积累，为加强个人信息保护事业持续赋能。

绿盟数据安全总体框架

THE EXPERT
BEHIND GIANTS
巨人背后的专家

多年以来，绿盟科技致力于安全攻防的研究，
为政府、金融、运营商、能源、交通、科教文卫等行业用户和各类型企业用户，
提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。
在这些巨人的后面，他们是备受信赖的专家。

客户支持热线：400-818-6868

NSFOCUS绿盟科技