CSA GCR cloud security alliance
GREATER CHINA REGION ®

交易所Top10安全风险

致谢

云安全联盟大中华区（简称：CSA GCR）区块链安全工作组在 2020 年 2 月份成立。包括 100 多位安全专家们，分别来自中国电子学会、耶鲁大学、北京大学、北京理工大学、武汉大学、世界银行、华为、腾讯、知道创宇、赛博英杰、元界 DNA、慢雾科技、安比实验室、启明星辰、天融信、联想、OPPO、零时科技、安永、阿斯利康等五十多家单位。

区块链安全工作组有 9 个项目小组，包括智能合约安全、数字钱包安全、共识算法安全、交易所安全、Dapp 安全、去中心化数字身份（DID）安全、网络层安全、数据层安全，AML 技术安全等方向。

本白皮书主要由交易所安全小组专家撰写，感谢以下专家的贡献：

区块链安全组组长：黄连金

交易所安全小组的领军人物：谭晓生

本文档原创作者：邓永凯、黄连金、谭晓生、叶振强、余晓光、余弦（按拼音字母排序）

本文档审核专家：陈大宏、赵勇

本文档贡献单位：华为

关于研究工作组的更多介绍，请在 CSA 大中华区官网（https://c-csa.cn/research/）上查看。

如本白皮书有不妥当之处，敬请读者联系 CSA GCR 秘书处给与雅正！联系邮箱：info@c-csa.cn；云安全联盟 CSA 公众号：

序言

CSA GCR 区块链安全工作组的交易所安全小组对于过去几年交易所发生的安全事件进行了分析，按照安全事件的发生频率和资金损失程度总结了主要的十个安全风险。对于每个风险进行解释和描述，给出有关的案例的文章链接供读者参考，并且给出应对措施和建议。

交易所管理用户的资金，因此所有交易所，不管大小，都需要金融级别的安全。金融系统的身份管理，两地三中心的高可用性和灾难备份的能力，安全风险感知，7×24 h 实时监测预警，数据安全和隐私保护等等安全控制是交易所必须建立的安全能力。CSA GCR 希望通过对于已经发生的交易所安全事故的分析，提出应对风险的可以落地的对策。希望交易所能够重视安全，避免为黑客打工。

李雨航 Yale Li

CSA 大中华区主席兼研究院院长

目录

1：高级长期威胁（APT：Advanced Persistent Threat）

风险描述：

高级长期威胁（英语：Advanced Persistent Threat，缩写：APT），又称高级持续性威胁、先进持续性威胁等，是指隐匿而持久的电脑入侵过程，通常由某些人员精心策划，针对特定的目标。其通常是出于商业或政治动机，针对特定组织或国家，并要求在长时间内保持高隐蔽性。高级长期威胁包含三个要素：高级、长期、威胁。高级强调的是使用复杂精密的恶意软件及技术以利用系统中的漏洞。长期暗指某个外部力量会持续监控特定目标，并从其获取数据。威胁则指人为参与策划的攻击。数字货币交易所的高级长期威胁一般是黑客在攻击之前对攻击对象的业务流程和目标系统进行精确的收集。在此收集的过程中，此攻击会主动挖掘被攻击对象身份管理系统和应用程序的漏洞，并利用电子邮件和其他钓鱼手段安装恶意软件潜伏等待成熟时机会，再利用 0 day 漏洞或者交易所流程方面的漏洞进行攻击。比较著名的针对数字货币交易所的 APT 黑客团队包括 CryptoCore（也被称呼为：Crypto-gang”，“Dangerous Password”，“Leery Turtle”大概成功盗取 2 亿美金）和 Lazarus（大概盗取 5 亿美金）。

相关案例链接：

1） 2019 年 5 月币安被盗 7000 多比特币: https://www.36kr.com/p/1723636465665

2） 美国交易所 Coinbase 在 2019 年 8 月描述他们如何处理 APT 攻击: https://www.coindesk.com/coinbase-says-it-foiled-a-sophisticated-hacking-attack

应对措施：

持续性的威胁需要交易所坚持不懈的防御。建议交易所的主要功能模块（交易，订单，资金管理，冷钱包，热钱包，衍生产品，Soft Stacking 等等功能模块 ）最好每季度进行第三方安全渗透测试，或者至少每年进行一次第三方的全方位的安全审计。每一次功能添加或者修改在上线以前必须通过第三方审计。交易所内部可以考虑成立一个红

队（Red Team)。交易所每一个工作人员必须通过安全培训，防止黑客钓鱼攻击，对于外部电子邮件的链接和附件没有经过安全部门检测不能打开。利用云服务（比如 AWS，阿里云等等）的系统，应该使用云服务商提供的身份管理和访问控制系统（IAM）和 API 安全服务。跟踪了解 APT 的黑客团队的新动向，特别是他们利用的 0 day 安全漏洞。对于交易所的各种服务器进行加固，公司的补丁管理必须作为公司的安全操作的标准流程。尽量利用冷钱包存储交易所的大部分资金。使用多签名的方法处理额度比较大的转账。

2：分布式拒绝服务（DDOS）

风险描述：

分布式拒绝服务攻击 DDoS 是一种基于拒绝服务攻击（DoS）的特殊形式。是一种分布的、协同的大规模攻击方式。单一的 DoS 攻击一般是采用一对一方式的，它利用网络协议和操作系统的一些缺陷，采用欺骗和伪装的策略来进行网络攻击，使网站服务器充斥大量要求回复的信息，消耗网络带宽或系统资源，导致网络或系统不胜负荷以至于瘫痪而停止提供正常的网络服务。与 DoS 攻击由单台主机发起攻击相比较，分布式拒绝服务攻击 DDoS 是借助数百、甚至数千台被入侵后安装了攻击进程的主机同时发起的集团行为。数字货币交易所经常受到 DDOS 攻击。

相关案例链接：

1）DDOS 攻击交易所 OKEX 和 Bitfinex

https://cointelegraph.com/news/ddos-attacks-on-okex-and-bitfinex-were-sophisticated-possibly-related

2）交易所 DDOS 攻击分析学术文章

https://ris.utwente.nl/ws/portalfiles/portal/92134801/Impact_of_DDoS_Attacks_on_Cryptocurrency_Exchange.pdf

3）DDOS 攻击交易所 Mt. Gox:

https://venturebeat.com/2013/04/04/mt-gox-outage-ddos-attack/

应对措施：

可以分为以下 6 点：

（1）制定拒绝服务响应计划

根据全面的安全评估，制定 DDoS 预防计划。与小型交易所不同，大型交易所可能需要复杂的基础架构，并需要多个团队参与 DDoS 规划。当 DDoS 出现时，没有时间考虑采取的最佳步骤。需要预先定义它们，以便迅速做出反应并避免任何影响。制定事件响应计划是迈向全面防御战略的关键第一步。根据基础架构，DDoS 响应计划可能会变得非常详尽。发生恶意攻击时所采取的第一步非常重要。需要确保数据中心已经准备好，团队知道他们的职责。这样可以最大程度地减少对业务的影响，并节省恢复时间。下面是关键要素：

系统清单。包括针对 DDOS 的工具（IPS， IDS）评估和筛选工具；组建响应小组；定义关键团队成员的职责，以确保对攻击进行有组织的反应；定义通知和升级流程；确保团队成员确切知道发生攻击时与谁联系，包括应告知攻击的内部和外部联系人列表；还应该制定与客户、云服务提供商和任何安全厂商的沟通策略。

（2）保护网络基础结构

只有采用多级保护策略，才能减轻网络安全威胁。这包括先进的入侵防御（IPS）和威胁管理系统，包括防火墙、VPN、反垃圾邮件、内容过滤、负载平衡和其他 DDoS 防御技术层。它们共同提供了持续、一致的网络保护，以防止 DDoS 攻击的发生。这包括以最高级别的精度来判断和阻止不正常的流量，以阻止攻击。大多数标准网络设备都带有有限的 DDoS 缓解选项，因此可能需要外包一些其他服务。借助基于云的 DDOS 解决方案，可以按使用量付费使用云安全服务提供的 DDOS 缓解和保护服务。除此之外，还应该确保系统是最新的，过时的系统通常是漏洞最多的系统。鉴于 DDoS 攻击的复杂性，如果没有适当的系统来识别流量异常并提供即时响应，几乎没有办法防御它们。在安全的基础架构和作战计划的支持下，此类系统可以最大程度地减少威胁。

（3）基本的网络安全：采取严格的安全措施可以防止业务网络受到损害。安全做法包括定期更改的复杂密码、反网络钓鱼方法以及允许很少的外部流量的安全防火墙。

仅这些措施并不能阻止 DDoS，但它们可以作为关键的安全基础。

（4）建立安全的网络架构。业务应创建冗余网络资源；如果一台服务器受到攻击，则其他服务器可以处理额外的网络流量。如果可能，服务器应在地理位置上位于不同的位置，因为攻击者更难以分散资源。

（5）利用云将 DDoS 防护外包给基于云的服务提供商具有许多优势。首先，与私有网络相比，云具有更大的带宽和资源。随着 DDoS 攻击程度的增加，仅依靠本地硬件可能会失败。其次，云的性质意味着它是一种分散的资源。基于云的应用程序可以在到达目标或目的地之前吸收有害或恶意的流量。第三，基于云的服务由软件工程师操作，他们的工作包括监视 Web 以获取最新的 DDoS 策略。在公司和行业之间，为数据和应用程序选择正确的环境将有所不同。混合环境可以方便地在安全性和灵活性之间实现适当的平衡，尤其是在供应商提供量身定制的解决方案的情况下。

（6）了解 DDOS 可能出现的警告标志： DDoS 攻击的一些症状包括网络速度降低，公司内部网路上的连接不正常或网站间歇性关闭。如果网路性能比平时减少，则该网络可能正在经历 DDoS，因此交易所应采取行动。

3：内鬼监守自盗（Insider Attack）

风险描述：

交易所内部人员利用公司内部安全流程的漏洞，监守自盗；或者在离开交易所以后利用流程和安全控制方面的漏洞发起攻击。

相关案例链接：

1）交易所 Coinsquare 的内鬼监守自盗攻击:
https://www.chainnews.com/news/150259287150.htm

2）交易所 Bithumb 的内鬼监守自盗攻击:
https://bcsec.org/index/detail/tag/1/id/533

3）数字钱包和交易应用 ShapeShift 的内鬼监守自盗攻击:

https://m.biben.com/article?id=611f0e4636c9d4f7330bc4f225228eca

应对措施：

交易所应该建立专业的安全团队，制定公司级别的安全策略（Security Policy)，培养全体工作人员的安全意识，进行定期的安全培训和安全上岗机制。做到职责分离（Separation of Duty)，最小权限（Least Privilege)。 对于高权限账号（比如数据库账号，系统根账号，资金管理账号等等）进行集中管理和监控，每一次使用应该通过流程批准，每一次使用以后需要更新口令，每一次使用都需要有时间限定和申请原因。员工离开交易所以后应该立即终止账号的使用。公司除了有安全策略以外，还需要指定标准的安全流程(Standard Security Procedures）和应急处理流程（Incident Response Procedures)。交易所可以利用第三方保险公司和安全储备资金的方法，应对安全事件发生以后对用户的损失进行赔偿。

交易所应当将审计程序作为交易所交易的一部分进行谨慎开发，确保在资金对账出现问题时第一时间由审计程序自动报警，并收紧提币流程。

交易所程序应当对用户相关的口令、API Key、充提地址等关键信息加密存储，避免 DBA 接触到明文。DBA 和运维权限要完全隔离。

4：API 安全风险问题

风险描述：

交易所一般都会公开订单查询、余额查询、市场价格交易、限价交易等等 API。API 的安全如果没有管理好，黑客可以利用 API 安全漏洞盗取资金。一般可能的 API 安全漏洞如下：

（1）没有身份验证的 API

API 必须有身份验证和授权机制。符合行业标准的身份验证和授权机制（例如 OAuth / OpenID Connect）以及传输层安全性（TLS）至关重要。

（2）代码注入

这种威胁有多种形式，但最典型的是 SQL，RegEx 和 XML 注入。在设计 API 时应了解这些威胁并为避免这些威胁而做出了努力，部署 API 后应进行持续的监控，以确认没有对生产环境造成任何漏洞。

（3）未加密的数据

仅仅依靠 HTTPS 或者 TLS 对于 API 的数据参数进行加密可能不够。对于个人隐私数据和资金有关的数据，有必要增加其他在应用层面的安全，比如 Data Masking， Data Tokenization, XML Encryption 等等。

（4）URI 中的数据

如果 API 密钥作为 URI 的一部分进行传输，则可能会受到黑客攻击。当 URI 详细信息出现在浏览器或系统日志中时，攻击者可能会访问包括 API 密钥和用户的敏感数据。最佳实践是将 API 密钥作为消息授权标头（Message Authorization Header）发送，因为这样做可以避免网关进行日志记录。

（5）API Token 和 API Secret 没有保护好

如果黑客能够获得客户甚至超级用户的 API Token 和 API Secret ，资金的安全就成为问题。

没有对于 API 的使用进行有效的检测，黑客可能利用 API 进行多账户、多笔的转账。API 的实时安全检测如果不能判断这种攻击，就会有损失。

相关案例链接

1）交易所 GateHub 的 API 被攻击导致 18,473 账号被黑：
https://gatehub.net/blog/gatehub-update-investigation-continues/

2）交易所币安部分用户 API token 被盗:
https://www.wired.com/story/hack-binance-cryptocurrency-exchange/

应对措施：

最常见的加强 API 安全性的方法：

（1）使用令牌。建立可信的身份，再通过使用分配给这些身份的令牌来控制对服务和资源的访问。

（2）使用加密和签名。通过 TLS， XML Encryption， 零知识证明等方法加密数据。要求使用数字签名，确保只有拥有权限的用户才能解密和修改数据。

（3）识别漏洞。确保操作系统、网络、驱动程序和 API 组件保持最新状态。了解如何全面实现协同工作，识别会被用于侵入 API 的薄弱之处。利用持续监控来检测安全问题并跟踪数据泄露。

（4）使用配额和限流。对 API 的调用频率设置限额，并跟踪其使用记录。如果 API 调用数量增多，表明它可能正被滥用，也可能是编程出了错，例如在无限循环中调用 API。指定限流规则，防止 API 出现调用激增和拒绝服务攻击。

（5）使用 API 安全网关。API 安全网关担当 API 流量策略执行点。好的网关既能帮助验证流量的使用者身份，也能控制和分析 API 使用情况。如果交易所服务器是部署在云上的，大部分头部的云服务提供商都有 API 安全网关解决方案或者第三方的 Market Place 上可以找到的 API 安全服务网关。

（6）交易所对 API 使用应加上 IP 限制，并识别同一 IP 使用多个 API 可能存在黑客风险，要特别注意防止重放攻击，关键 API 不允许重复提交调用。

5： 假充值问题（False Top-up）

风险描述：

假充值是指链上逻辑错误或交易所链上链下对接的时候，对交易的检验不够严谨导致的错误入账的问题

相关案例链接：

1）EOS 假充值：

https://www.anquanke.com/post/id/173315

2）以太坊代币假充值：

https://zhuanlan.zhihu.com/p/39499902

3）USDT 假充值：

https://mp.weixin.qq.com/s/CtAKLNe0MOKDyUFaod4_hw

4）以太坊以太币假充值

https://t.zsxq.com/YNbMFIa

5）XRP 假充值

https://developers.ripple.com/partial-payments.html

6）门罗币假充值

https://hackerone.com/reports/364904

应对措施：

EOS 假充值修复：

使用默认的 history plugin 配置，除此之外检查 EOS 转账的交易状态，确保交易执行状态为 “executed”。同时，也需要判断以下几点防止其他类型的 “假充值”：

判断 action 是否为 transfer；判断合约账号是否为 eosio.token 或其它 token 的官方合约账号；判断代币名称及精度；判断金额；判断 to 是否是自己平台的充币账号。

以太坊代币假充值修复：

除了判断交易事务 success 之外，还应二次判断充值钱包地址的 balance 是否准确的增加。其实这个二次判断可以通过 Event 事件日志来进行。很多中心化交易所、钱包等服务平台会通过 Event 事件日志来获取转账额度，以此判断转账的准确性。但这里就需要特别注意合约作恶情况，因为 Event 是可以任意编写的，不是强制默认不可篡改的选项：

emit Transfer(from, to, value); // value 等参数可以任意定义

作为平台方，在对接新上线的代币合约之前，应该做好严格的安全审计，这种安全审计必须强制代币合约方执行最佳安全实践。

作为代币合约方，在编码上，应该严格执行最佳安全实践，并请第三方职业安全审计机构完成严谨完备的安全审计。

USDT 假充值修复：

交易所自查 USDT 处理逻辑，立即安排功能下线修正并且排查历史 USDT 交易记录；

引入专业代码审计，提升代码的健壮性；

提升开发人员对于区块链技术的基本认知，避免错误的认知导致错误的结果；

提升交易所整体风险控制流程，对于疑似风险交易予以拦截。

以太坊以太币假充值修复：

针对使用合约进行 ETH 充值时，需要判断内联交易中是否有 revert 的交易，如果存在 revert 的交易，则拒绝入账。针对使用合约进行 ETH 充值时，需要判断内联交易中是否有 Out of gas 的交易，如果存在 Out of gas 的交易，则拒绝入账。针对使用合约进行 ETH 充值时，需要判断内联交易中是否有 Error 字段的交易，如果存在 Error 字段的交易，则拒绝入账。采用人工入账的方式处理合约入账，确认充值地址到账后才进行人工入账。针对使用合约进行 ETH 假充值时，除了 revert 和 Out of gas 的手法外，不排除未来有新的手法，安全团队需要持续保持关注和研究。

XRP 假充值修复：

交易所充值 XRP 到账应该判断 deliver_amount 字段，如果只判断了 Amount 则会造成假充值。

门罗币假充值修复：

升级到最新的版本

https://github.com/monero-project/monero/issues/3983

6：交易所热钱包存储过多资金，成为黑客目标

风险描述：

交易所热钱包存储过多资金，成为黑客目标，这个风险与交易所热钱包有关的 IT 系统的漏洞、采用不安全的存储方式对私钥进行存储、安全意识较低有关。黑客采用包括但不限于以下的方式进行攻击：

恶意链接钓鱼收集用户信息。黑客投放恶意链接引导用户点击，借此收集用户的登陆凭据。

数据库被攻击导致私钥泄露。交易所数据库中存放其热钱包私钥，黑客对数据库进行攻击，获取到数据库数据后通过数据库存放的私钥进行转账。

IT 系统漏洞。交易所自身系统存在漏洞，黑客通过其自由漏洞获取 IT 系统控制权后，直接通过 IT 系统进行转账。

员工监守自盗 。前雇员在离职后通过在职时留下的后门进行资产转移。

相关案例链接：

1）交易所 UPbit 的热钱包被盗 5 千万美金

https://www.welivesecurity.com/2019/11/27/upbit-cryptocurrency-exchange-hack/

2）2019 年交易所 Cryptopia，CoinBene，BitPoint， Binance，都因为热钱包存储过多资金被黑客盗取。

https://cointelegraph.com/news/most-significant-hacks-of-2019-new-record-of-twelve-in-one-year

应对措施：

对于交易所热钱包的安全问题有以下应对措施：

1）交易所单个热钱包内不应存储过多资金。以防因某钱包私钥泄露导致全部资金

被盗取。采用热钱包池等方式将结构化资金分散提高安全性。

2）采用体系化的钱包资金储存及权限管理策略。对不同钱包的权限做好合理鉴权。

3）不应将钱包私钥明文、编码或或弱加密后直接存储到数据库中。应将私钥加密对应 KEY 存储在不同数据库中。防止由于某一数据库泄露后导致其中钱包全被控制。

4）制定公司级别的安全策略。对于前/现内部人员可能导致的安全问题按照内鬼监守自盗的防御方式来进行防御。

5）可以进一步参考 CSA GCR 数字钱包安全工作小组发表的文献。

7: 51% 攻击 （也可以称为硬分叉攻击，或者双花攻击）

风险描述：

51%攻击，又被称为 Majority attack。这种攻击是通过控制网络算力实现双花。如果攻击者控制了网络中 50%以上的算力，那么在他控制算力的这段时间，他可以将区块逆转，进行反向交易，实现双花。对同一笔交易进行双重花费甚至回滚以往的历史交易。

相关案例链接：

1）BTG 双花攻击:

https://www.zdnet.com/article/bitcoin-gold-hit-with-double-spend-attacks-18-million-lost/

https://forum.bitcoingold.org/t/double-spend-attacks-on-exchanges/1362

2）ETC 双花攻击 ：

https://cointelegraph.com/news/coinbase-ethereum-classic-double-spending-involved-more-than-11-million-in-crypto

应对措施：

根据以往发生的 51%攻击案例，51% 攻击一旦成为真实场景下的成熟攻击方法，各个公链都需要小心，虽然通过 51%攻击需要很大代价，但在小币种公链网络中情况就发生了变化。况且，没有 50%以上的算力，还是有机会成功的，只是概率低而已。

无论工作量证明的 PoW，还是权益证明 PoS，还是委托权益证明 DPoS，只要在共识问题里面，理论上讲都无法避免出现 51% 攻击情况，在不同共识机制的实现中还可能存在各种其他问题，而且在熊市的这段时间里，算力下降，币价大跌，主网相对更加脆弱的，更容易出现 51%攻击问题。

目前，针对 51%攻击的防御主要有如下几种：

- 提高确认次数至 500 个以上。
- 改善共识机制。例如，由原先的 PoW 改为 PoW + PoS。
- 升级新的共识算法。比如 Bitcoin Gold 遭受 51%攻击之后表示，将开发新的 PoW 算法以替代原有的 Equihash 算法。
- 与数字资产交易平台合作，同步相关信息，阻断黑客的套现渠道。

8: 不安全的文件处理

风险描述：

这种风险与文件的不安全处理有关系。包括下载外部电子邮件的链接或者附件，也就是传统意义上的钓鱼攻击；也包括对于交易所用户上载的 KYC（实名验证）文件没有经过安全处理。恶意代码隐藏图像中，这种方式也称呼为隐写术（Steganography），攻击者将恶意代码与指令隐藏在看似无害的图像之中伺机执行，这种风险与 APT 风险有一定的关系。一般来说，单单一封邮件无法对你实施攻击，一定要以邮件为基础，在此之上产生别的交互才可以，比如说点击链接后输入内容，运行/打开文件，当需要以上动作时，便存在风险。

相关案例链接：

1）交易所 BitStamp 文件不安全处理被黑客攻击：

https://www.coindesk.com/unconfirmed-report-5-million-bitstamp-bitcoin-exchange

2）CoinCheck 文件不安全处理被黑客攻击：

https://cointelegraph.com/news/report-record-breaking-coincheck-hack-perpetrated-by-virus-tied-to-russian-hackers

3）LocalBitcoins 文件不安全处理被黑客攻击：

https://www.coindesk.com/localbitcoins-user-funds-stolen-chat-client-hack

4)下面这篇文章介绍隐写术：

https://securityintelligence.com/steganography-a-safe-haven-for-malware/

应对措施：

对于不安全的文件的防御措施一般有：

不要随意安装未知来源的软件，在正常功能之外添加恶意远程控制代码且不被发现，是非常容易做到的。某知名交易所就是因为客服从陌生人处获取并打开了安装包后，黑客获取权限进入内网才导致了重大损失；

在非必要情况下禁用 word 中的宏，word 文档远不如看起来无害。宏病毒可能在你打开那个文档时恶意宏命令已经开始执行。

对 KYC 中上传的文件的 HEX 内容进行校验，其中可能存在隐写的控制代码或恶意命令，隐藏在文件中或文件尾。

如果必要的话对于 LSB 隐写进行识别。其中图像隐写中肉眼难以分辨，但广泛存在的隐写方式为 LSB 隐写，通过将秘密信息嵌入到载体文件的最低有效位，对载体文件做很小的改变就能隐藏大量的秘密信息。

用户上传的图片需统一处理，并转换为常用的 JPG 等格式。要禁止用户上传 word、pdf 等文档，此类文件在恶意构造的情况下可用于攻击审核人员或客服的办公计算机。

9： DNS 域名劫持 （DNS domain name hijacking）

风险描述：

DNS 服务是互联网的基础服务，在 DNS 查询中，需要有多个服务器之间交互，所有的交互的过程依赖于服务器得到正确的信息，在这个过程中可能导致访问需求被劫持。

劫持访问需求有多种方式：

利用路由协议漏洞，在网络上进行 DNS 域名劫持。如 BGP 协议漏洞（BGP 协议对于两个已经成功建立 BGP 连接的 AS 来说，基本会无条件的相信对方 AS 所传来的信息，包括对方声称所拥有的 IP 地址范围），将受害者的流量截获，并返回错误的 DNS 地址和证书。

劫持者控制域名的一台或多台权威服务器，并返回错误信息。

递归服务器缓存投毒，将大量有毒数据注入递归服务器，导致域名对应信息被篡改。

入侵域名注册系统，篡改域名数据，误导用户的访问。

上述的攻击行为都会将用户的访问重定向至劫持者控制的一个地址。使用一个假冒的证书让不明真相的用户登陆，如果用户无视浏览器的证书无效风险警告，继续开始交易，就会导致钱包里的资金被盗。

相关案例链接：

1）交易所 CoinCheck 域名被黑：

https://www.zdnet.com/article/hackers-hijack-one-of-coinchecks-domains-for-spear-phishing-attacks/

2）去中心化交易所 Etherdelta 域名被黑:

http://www.leilu.com/archives/1746

应对措施:

交易所应该采用技术手段，防止 DNS 劫持：

选择安全技术实力强的大型域名注册商，并且给自己的域名权威数据上锁，防止域名权威数据被篡改。

选择支持 DNSSEC 的域名解析服务商，并且给自己的域名实施 DNSSEC。DNSSEC 能够保证递归 DNS 服务器和权威 DNS 服务器之间的通信不被篡改。

在客户端和递归DNS服务器通信的最后一英里使用DNS加密技术，如DNS-over-TLS，DNS-over-HTTPS 等。

用户应重视网络安全，不做不安全的操作，确保钱包安全。

提高安全意识，不要越过 HTTPS 证书强制访问。

开启 HSTS 功能。HSTS（HTTP Strict Transport Security）是浏览器支持的一个 Web 安全策略，如果开启了这个配置，浏览器发现 HTTPS 证书错误后就会强制不让用户继续访问。

使用安全性更高的硬件钱包。

10：第三方安全

风险描述：

使用第三方服务的时候：

因为交易所使用第三方服务自行配置错误导致被黑；

因为第三方服务自身漏洞导致交易所被黑；

因为第三方服务被利用来钓鱼投毒投马导致交易所被黑；

因为第三方服务被黑导致交易所被黑。

相关案例链接：

1）MyEtherWallet DNS 劫持事件深度分析

https://www.freebuf.com/articles/blockchain-articles/169773.html

2）一个通杀绝大多数交易平台的 XSS 0 day 漏洞

https://mp.weixin.qq.com/s/yfbKf_5Nk2NXFl2-xlFqKg

3）全球知名第三方 js 服务被劫持

https://wx.zsxq.com/dweb2/index/group/225441212851?from=mweb&type=detail

4）门罗币钱包之“狸猫换太子”

https://mp.weixin.qq.com/s/PeIgmHDEgy0k8OU9R0q02A

5）PyPI 官方仓库遭遇 request 恶意包投毒

https://security.tencent.com/index.php/blog/msg/160

应对措施：

对待第三方服务和组件安全，在日常开发和运维中，需要尽可能做到及时捕获和处理"黑天鹅"事件：

1、接入第三方服务的时候需要进行初步的安全评估，尽可能的列出所有潜在的安全风险，并输出安全性需求分析；列风险，出需求 ；

2、对于组件应要做到及时发现捕获安全漏洞情报，并第一时间安排处理；早发现，早处理；

3、尽可能不采用灰名单的第三方服务，需要接入第三方服务可以优先选择"慢雾精选推荐"，不在"慢雾精选推荐"里的，请进行第 1, 2 点。

对开发人员的钓鱼攻击防范：

针对 npm 包，pip 包，RubyGems 包，vscode 插件对开发人员进行钓鱼的手法越来越多，防不胜防。开发人员在安装各种依赖的时候，务必要去官方网站上检查，务必确认安装的包的名字与官方的包一致。有条件的可以考虑自建包管理仓库，并且关注对应

包的更新动态，及时跟进安全更新，要求开发人员从自建源下载安装对应的包。

日常防护：

慢雾安全团队对交易所常用的第三方服务进行初步安全调查发现，部分第三方服务存在安全风险，应该纳入灰名单，如果有更好的选择，尽量不要使用灰名单中的服务，交易所在使用第三方服务的时候也需要事先进行安全调研。

邮箱：info@c-csa.cn
官网：https://c-csa.cn