三教(00:20:12): 喂我是。苗苗在长春他去跑马去了，他每每周都有新人？还是小？他们是回回回 hello？官方。不就是这玩意他。不发货了。跳吧跳。

三教(00:21:17): 上面转过。线上有同学能说个话，测试一下。喵喵。

三教(00:21:33): 可以。

f(00:21:41): 好像听不见，这边这个主持人讲话。

三教(00:21:47): 不能听见我说话?

f(00:21:48): 现在能听见了?

三教(00:21:50): 我没有说话，谢谢!

f(00:22:20): 那今天的贴纸是什么？今天贴纸什么?

三教(00:22:49): 你要用你的电脑吗？为什么要说让我用，也可以没问题，你你共享就行。就我这就放着。拿水果或者你直接就腾讯会议共享屏幕就可以不需要啥，你在这里就播放就行了，你要给我不让我用自己的电脑。对它儿锅。你直接共享过来，我就能看为什么不拆下来，自己还得改一下设置重新加入。

三教(00:23:53): 再大一点吗？我觉得挺好的，就这样，就这样，我们是 ios 的分辨率的问题，左右边上下左右都有黑边。上下的飞机可能是你，我这有个就比较神秘的关系，挂哪儿这衣服加，在家里领口，可以放口袋里，

三教(00:24:28): 主持人离开会议。你成为主持人阿里的语音录制吗？怎么回事，弄到哪去了，不知道我开一个云录制吧。说话人生三人，阿里云 QQ 空间里写的，你看我跟他讲这个不是我接触这个是初中的时候，我在抓包的时候 QQ 空间还是会的，你的初衷也太晚了。哪一年可能是，但是我第一次听说就是因为我空间，你你你现在是几年级？打打开。好像是可能的，八年前是的，但是六元你还是一不是初中的事情是16年的，没有就没回。妈的再聊一会。

三教(00:26:01): 不是技术空间，为什么需要配置，不可能是因为 qq 放弃他了，男孩子进入了。不知道现在我觉得好像可以放一些炒饭了？他有他有。肯定没有谁可能会考证体验最好不卖是不是？那你不知道?

三教(00:26:48): 看不到。看过图的。可以帮我不是我说的不方便，他不是我可能改任何自己三分吧，看来你差不多开始喂。看一下视频流是咋样？行看电视看看。我应该从去哪看视频？在这从这儿放这儿产品还在吗？不知道现在是不是还在。

三教(00:28:21): 那就这样放，我觉得我不是很大，我把自动隐藏。还是之类的就好，就这样的喂那个对我们差不多已经重新定义到七点整了，各位。对这是 UTC 加八，不是 UTC 加七小时54分，不知道不重要，这个各位线上线下同学大家好，然后今天因为会长去为组会健康工作了，然后我又返聘上岗了，然后来对。host1下这次的，然后这次我们有一个网页的董文冲同学来给我们带来一些关于这个先进的传输层和应用层的一些分享，然后也希望。大家能从中学到一些东西，我就不多讲，然后等结束之后，大家可以来拿一些新的 sticker 都在最前面线上的同学就看看吧，谢谢!

三教(00:29:33): 大家好，我是开始鼓掌好紧张，然后我是董文冲，我是网研院的，大家都不知道知不知道网研院是小众养老院系非常养老，非常养生。可能也不太好毕业。然后婉媛媛的唯11个缺点是男女比不太理想，但是我一看比在座的这个情况理想一点。

三教(00:29:55): ok 那今天我讲我跟大家分享一下我们的相关的一些东西，但是这些整体出发点可能偏科研一点，就不太偏应用，当然也会有一些应用的成分在内，但主要还是可能也跟大家也讲一下科研方面可能。

三教(00:30:12): quick 的最新进展之类的争取涵盖以下内容就争取涵盖无道，就涵盖不到就是首先是对 quick 做一个简单的介绍，然后我们讲一下就是在。在互联网扫描里面，一个比较重要的就是怎么去发现互联网上配合部署。然后第三个就是 quick 它可能面临一些放大攻击的问题，我们讲一下跟放大攻击相关的一些情况，然后最后一个可能就是不能明说的就是我们可能讲一下 quick 的阻断，然后基于 quick 的代理之类的这些乱七八糟的东西。我们争取不要被不要看到一半，然后挂一个人说你们不要再讲了，我希望不会这样，或者腾讯会议被掐了之类的也没有那么敏感。

三教(00:30:57): 第一点就是我们最开始是为什么要搞一个 quick 这样的新的传输层协议。大家都知道这个 TCP 和 UDP 他们最开始是那种非

常原始的一种有连接和无连接的状态，就是 UDP 大家都知道就是那种很原始，除了什么目标，IP 目标端口，IP 端口，后面就全是配帽子，就他没有什么。甚至没有什么结构可言。当然在最开始设立这些协议的时候，就是当时的互联网还是一个大家都互相信任的状态，就是当时谁在用互联网，大家都知道你你你发现有人在搞，你可以直接过于线下真实的，现在肯定不能这样了，所以当时大家都互相彼此互相信任，最开始设计的时候就没有，也没有考虑。

三教(00:31:40): 面向的问题，然后 TCP 它也是一个很好的协议，它最开始设计的时候就没有考虑到互联网会发展成现在这个样子，所以它最开始设计的时候，它我们后来发现它有很多的性能问题，比如说就是没有多路复用，就是一个动物复用，使用多个来传输不同的东西，这样就没有这个。

三教(00:31:59): 海涛的问题，第二个就是三次握手进行连接，当然这个问题其实比较小，因为 quick 其实也没有完美的解决，说我能不能在任何情况下都比较快速的进行连接，甚至我们也观察到有一些情况下。quick 使用，甚至四次五次，甚至是用两个 RPG 能建立这样的连接，它甚至比如九州 CP。那第三个就是当时的设备基本上都是固定 IP 的就是他没有考虑到我这个 IP 用着。突然我这个设备用着，突然从一个 IP 切换到另外一个 IP。所以在当时的场景下，这种情况几乎不可能发生，所以当时也没有考虑说，能把一个已有的连接迁移到一个新的 IP 上。

三教(00:32:41): 然后最后一个就是也没有完全解决这个其实也没有完全解决，就是用测控制文物探测这个这部分其实在 F c9002里面，然后现在现有的一些工作也证明了在一个恶意的客户端的情况下，可以去恶意的干扰服务端的用测控制的策略。

三教(00:32:59): 当然不是我们今天讨论的内容，因为我也不懂。这个我也不太懂，反正他自己也是这么说的，他说他能他说他比 TCB 好，但是实际上他也没有比 PCB 好多少，就反正肯定是有改进的。但是改进是有一定的代价，就是它变得很雍容，变得很大，以至于现在你一个人想要部署整套配合，其实是较为困难的，也没有什么一件一件的好用的文字来帮你解决这个事情。

三教(00:33:30): 反正就是最开始是为了解决这些问题，所以。The Google. 牵头说，决定设立1个新的协议，用来取代 TCP 最开始内部叫机会在我们自己内部支持或者怎么样，反正大概12年前后吧，然后大概陆陆续续改了大概89年到21年前后发了一个正式的 quick V 一。那 quick，它是一个基于 UDP 的传输层协议，虽然它基于 UDP，但是它还是一个传输层协议，这主要是为了保证这个中间件的支持就是有的中间件，他看到你这个 IP IP 报文，一看你这个是我不认识的东西。他就直接把这个报文丢掉。所以为了考虑，他就把这个东西把 quick 建立在 UDP 上，但是你想 UDP 反正它的结构也很简单，就是除了前面一些暴露就是整个配的，所以预期再设立一个新的协议的版本，不如就直接用用，用现成的就放在 UDP 上。它是一种类似于 TCP，它实际上类似于 TCP，它是保证可靠，然后有一些建立的过程就有流传输，甚至可以有多种流，而且它内存的这个 TRS 点三，为了防止这个。

三教(00:34:41): 为了防止这个不安全的传输就是它内置1.3就要求你强行使用 trs 123，这其实也是一个他对个人用户不是那么友好的，不是那么友点就是现在的情况下，就是虽然大家都有什么，就是那个叫什么。之类的东西可以帮你自己弄一点证书，但是这种东西反正不是一个人就很好，搞的东西，反正 quick 它内存这个 T r4.3题外的话就是这个 T r1.3其实它也类似于这个 quick quick 跑在 UDP 上就是 tr s1.3跑在 tr s1.2上就是 t1.3的报文。他自己说，这个版本号是 tr s1.2就是为了防止中间件把它不小心给丢掉。因为在 tr s1.2到 tr s2.3中间隔了很长时间，那个时候很多中间件就假设说不会有更新的版本号，所以他们就会把所有他们不认识的版本号都丢掉，最开始就是这个考虑导致 tr s1.3的版本号是 tr s1.2。

三教(00:35:38): 有单独一个拓展里边声明说我其实是123这个和 quick 跑在 UDP 上有点类似的就是为了防止中间件把他不认识的东西丢掉。差不多一个意思也是也简单，你看又有什么样，又有这张那什么什么什么远远远景都买，好像不知道什么东西。它最开始设计是服务于这个 thtt p3 htt p3也是一个和 ht p2比较类似的东西，但是我们可能不在今天可能主要还是 quick。然后他最开始是不给机会的，然后最后还是在 f8999九千九千零一，甚至9002什么九千九千三百多局，反而还有一大堆的有大压的 FC 来定义这个配合，那最重要的其实是就是九千九千零一这样子一个是定义这个唯一的一些传输的情况。

三教(00:36:33): 学员名义是钉钉要在 quick 上怎么怎么进行 TRS 协商怎么进行握手之类的？就他的这个连接建立的过程和 TCB 是比较类似的，但是它有一个单独的结构叫 initial，这个 initial 包就是用来专门用来握手的，这样一个结构就是客户端服务端就是提供包发起握手的时候回应握手，然后服务端会用一个包就发送这个客户端加密信息，这个过程是如果你客户端没有。服务端的缓存的东西，那这个过程是一个 RGT 的就是也是一个 RTT 可以建立这样的连接，但是它还有一种。我把这个调一下。不对我应该用这个数据。我就挑一。

三教(00:37:23): ok 就是它有一个另外的策略就是如果你客户端本地缓存一些关于服务端的信息，比如说我们刚刚说有 TRCR 客户端可能它本地缓存这个证书乱七八糟的情况下全缓存下来，那这个情况下，它就可以直接跟服务端发加密后的信息。这个过程我们称为0R TT 握手就是他就不用再进行 TRS 协商，直接用已有的东西去给他跟他进行，然后进行数据的传输。那边一解密的话，我这个能正常的解密就知道这是一个 MRT 的。

三教(00:37:53): RT 的这样一个过程，就是在这个情况下，就不需要进行一个 RTD 的一个 RT 的握手流程，就是他宣称说他可以把对已知的这个。服务器的访问从一个 ID 已经降降降低到0R DD，就是你不需要额外一个 IDD 来进行这个握手的流程，但是我们先展示说这个情况其实不是那么的。普遍就是甚至需要两个 IDE 才能进行这样的握手，为什么就是他在设计的时候有一个机制，就是这个地址验证，就是为了防止这个对 IP 进行伪造导致的这样的。攻击的情况就是 IP 进行伪造，是一种典型的进行反射式方法攻击的策略。然后他说那我为了防止这个反射式方法攻击的风险，你你如果想跟我握手，我会先返回给你一个 token 然后你把这个 token 的原样给我发过来，然后我通过确认我收到的 token 给我发的 token。同一个我都知道我确实是在跟你宣称的这个 IP 通信。那这个过程中你想你你给发发过去，大家把它给发过来，这个过程把连接建立从一个 RP 变成两个 ID，就是为什么很多时候 quick 的性能在连接建立这方面的性能还不如 TCP 因。

三教(00:39:06): 很多服务端的实现会积极的使用这种策略，比如说 nginx nginx1定会给你回回一个 token，我不知道为什么，但是有时候它的配置可能有问题，反正他一定会给你回一个 token，就这种情况下，quick 所宣称的一个 RT 甚至一个 RT 的握手，有时候往往是两个 RD 才能进行这样的握手。虽然这个机制不是就是这个机制，大部分服务端都有实现，它使用的不是很广泛，但是你一旦遇到了那它的性能比 TCB 不仅没有改进，反而还有倒退。

三教(00:39:36): 然后这边展示的是一个盈利包的结构就是前面这些乱七八糟的版本，它的版本非常长的版本有32位。不知道为什么，大家说怎么会出这么多版本，我不知道他可能很激进的，用了很长的这个版本号。不知道为什么，反正他就搞了这里边这个版本好像又有一个很神秘的地方，比如说就是他会有大概32分之一的版本，是用来协商的。就有32分之一的空间，它是不能用来通信。我不知道算对没有，反正大概这个数。

f(00:40:16): 刚才就是下面提问的好像不太清楚下面提问的什么问题。

三教(00:40:28): 下边没有问题提问，下边跟着我吐槽了一下。

f(00:40:38): 下面人说话不太能听得清的。

三教(00:40:35): R 二。还在海南。

三教(00:40:58): 然后在这个方面，他又用了一种标带i的表示他用了一种他我也不知道到底是哪提出的，但是他们都在用的一种，就是把这个长度编码到这个字段里边的一种，一种做法就是他用最高的两位来告诉大家说这个字段有多长，最高的两位可能是零零，就说明这个东西的长度，字段长度是。确实是意思，我不知道这个东西可能确实是从 UDF 里面吸取的。看上去信息量有点大，这个是 uint64，但是他能编的这个他配置比现场整数它最大的是262台。对它最高22次方就是它最多，比如说最高分是11的话，它就告诉你说这个东西的长度是八，就是它这个不是1234或者什么它是1248。所以可能你后边可能其实是有很大程度的浪费的。他就这么干了。等于有对齐的思想。

三教(00:42:17): token 可以这么长，可以这么讲，就是 token 的，就是这个更前面说这个是 token 的长度是用这个东西编码的，不是 token 是用这个编码的长度是用编码的。就你你自己考虑那你不想这么大可能吗？不可能吧？token1般来说，我们观察到一般是可能64，128，256什么？然后 quick 的就是它的传输的基本单元就是帧，就是它可能就是这部分，可能你听着可能有点的意思就是它传说最好单元是真这边演示了几个帧，比如说这个申请就是用用于填充的帧，但是它是零零零零零，这个 frame 就是用来就是听 frame，用来。就明天是不是活着之类的，然后就是用来承载这个 TLS 信息的就是他理论上可以承载别的东西在里面，它就是放 TRS123就是它的这个就是告诉你说它是一个大的数据可能被骗成好几片，然后每每片填到一个。里面。但是一般来说其实不需要分片，就在 tr s123里面，就这个 C-Data 就是 tr s123的这个卡的行后，它承载一个整个卡的行后，或者是 Server 分类的，然后他们就用这种方式来互相传递自己的这个握手的数据。

三教(00:43:51): 这种流程巡检和这个 HTTPS TRS 协商，这是基基本上一样的，他们会协商一些结果用于后续的传递数据的加密，然后在很多帧之后就是一个拍一个数据包，它可能它配可能存在若干个帧。我把所有帧都合到一块，然后进行刚刚说的这个 TRS 的加密，这样。包头也有一些加密，它包包头身上有一些混淆，就是从他们要传递的这些数据里边取最前面的一部分就是 sample，然后把用这个 sample 去加去把包头的一些。

三教(00:44:23): 字段给掩盖起来，比如什么包包，包括长度什么乱七八糟，什么 D CI D 的长度之类的会把我们俩掩盖起来。在9369是23年的一个标准，就是他为了避免我们之前说的那个中间件，导致把包乱丢的情况，他预先放了协议栈位，然后这个协议就只是换了一些一些常量的值，其他什么都没。这个术语叫什么叫 ocification 或者中文翻译叫僵化，意思可能对于他不支持的东西，他都直接丢掉，或者怎么样，就这种行为就是我们说是中间件导致的一个互联网的僵化就是它的目的是为了防止僵化，但 QUIC VR 现在的支持就基本上不普遍，因为 VR。唯一基本上是一个东西，只是它改了一些常量，就是包括这什么 S 之类的和一些特殊字段的这个数可能就是什么枚举值从一变成零，从零变成一之类的。就是他基本上是一个占位的协议。qu 传输的单元性流就是每每每个实际的数据是，一般来说是封装在一个流，或者是。

三教(00:45:34): 或者是几个数据在一个流里面进行的，就是这个理由是一个可以是单向的，也可以是双向的，就是他们要解决一个什么问题，它为了解决 HTP 它的承载在应用它在应用层，它对传输层是不可见的。比如说如果你传输层丢了某一个包，你不知道它丢的是哪个的话。传输层会传输层，不知道它到底是哪一个文件的组成部分，它可能会阻碍其他障碍传输的文件。但是如果你把这个文文件的单位给下下，放到这个传输层的话，传输层知道你到底是哪一个，哪理由里面丢掉了这个数据。都可以只重传某一个旅游的某一部分就不会理论上说不会发生那种应该确实有这种改进，但是也没有人具体的测过，说这个改进有多么有效，反正它的最小的传输。

三教(00:46:29): 单元是真，然后它每它的数据是在这个流中进行的，它允许任意数量的牛并发，但是这个牛肯定是有一个上限的。它是用一些特殊的针来进行流的控制，比如说它有这样的 stream stream 就是专门传输它的数据的，它有一个 ID，然后这个 type 可能就是一些乱七八糟的东西，比如说可能是它会。控制最大的数据量会有多少最大的旅游数据量就是每一个理由上面最多能存多少数据，然后在哪个流上有没有发生数据的阻塞，或者是在哪一个理由上，我不要再继续发生数据之类的。可能会把多个帧拼到一个包里，然后拼好之后再进行用 TRS 这样的加密。

三教(00:47:19): quick 有一个新的特性叫连接迁移，它就是为了解决 TCP 或者 UDP 的其中一个问题。就是我切换 IP 我怎么保证保持我之前的连接就是我们都知道它互联网它分层它是希望。各层之间保持一种独立的状态，那这样的话，如果我切换到我的这个 IP 对上层来说，它他知道不知道这是一个大家，没有想好的问题，但是在这种情况下，我们说他 quick 是希望让网络层和这个传输层进行某种程度的交互，使得我可以把一个连接从一个旧的 IP 切换到一个新的 IP。比如说我现在正在用数据流量，然后我现在连上校园网，那对于传统的这个 TCP 或者 UTP 来说，我其实就是重新建立一个连接，或者另外一个场景，就是我服务端有多个 IP，然后在多个 IP 之中，它有一个偏好，比如说它偏好用 ip v4，或者它偏好一个 I p6那这样的话，它就提供一种方法，能够让你把旧的连接迁移到一个迁移到新的 IP 上。他的想法，反正其实是非常简单的，就是通过一个 pass-through 认真在新的 IP 上发送一个加密后的随机的数据，然后接通，通过这个 pass response 把这个数据发过去确认确实是他收到了。然后这样的话，他们就可以协商把这个链接迁移到新的 IP 上，他们甚至可以写上一个新的连接 I。

三教(00:48:45): 我们刚刚讲的，现在其实有两个版本，其实它有一大堆草案版本，它是用于有一个版本形象的机制，用来确认服务端它到底支持什么版本。它有一个固定的格式，就是这样的格式，这个问号表示一个任意的任意寻求进去，然后这个 a 就是写死，就是所有这样的，它都是被认为是可以强制开启一个版本协商的就是它的规定里面写说你只要把这个版本号填成这样的你你给你给别人发过去，他就应该给你返回一个。它支持的版本列表，但我们发现很多实验，根本不管你发现他这个包他们不回。大家都知道实践各有各的毛病，然后直到 rf c9368里边他才完整的规范了这个版本形象的流程，然后它的这个版本箱包反正就大概长这样，这个 Version 就一定写死是零，然后后边后边是后面这个 on support version 就是一大堆，一大堆，就每每一个32位是一个它支持的版本的。

三教(00:49:46): 这个版本号。比如说。那你你算的应该比我算的，对我当时凭人印象瞎说的。反正有它的版本号，你看它有32位，还有32次方，反正多的很，他用六千六千五百六五三六分之一用来标识这是一个版本上的包，你也不知道他为什么。我也不知道为什么使用。就是这种形式来强制开启百万级，你用一个我觉得就行，我要用一个就行，反正我不知道到底是出于一种什么样的考虑。

三教(00:50:25): 接下来可以讲一些已有的关于 quick 的科研方面的工作，就是在21年的时候，就是21年可能刚发布或者怎么样，就是他就是就这帮人对 QUICK V 一和他们之前的各种协议草案在互联网上的部署情况实现了上扬。然后他们的这个测量就是他们的这个客户端的，他们使用客户端就是基于这个 quick 的 go 的实现。他们使用一些方法去查找这个潜在的这个 QUIC的实现。比如说他们想说 quick 的最大的应用这个 APP 三。然后说，IP 有域名解析到它，就是如果一个有如果有或多个域名对应这个 IP 的话，那相较于一个没有绑定域名的 IP，那他就更有可能部署 quick。同时，他们使用这种 HH ACP 和这个 HBS 的 DNS 记录，就当时还是草案，就现在这个这两个东西已经进

入标准了，就是使用这个 application 就是应用层的这个 protocol 就是应用层的这个协议的这样的记录。这样的 DNS 的信物用来进行这样的这样查找就是 LPN 等于 h2h3就是告诉大家说这个地方，这个 IP 是支持 h2和 h3的 hp rhd p3。

三教(00:51:34): 然后另外他们使用这个 AAA 来提供这个潜在的 ip v6信息，就是大家都知道 ip v6地址空间大得很，你不可能进行一个大大范围的普查，所以他们就选大大批域名，然后对这些域名进行 IPU 的解析。如果说我 AA 命中，他说这也可能存在一个，这也可能存在亏额不足。当然可能不存在，当然可能就是一个普通的 APP 没有什么 ap p3，然后其他就是他们进行了全网这个 APS 的服务，或者怎么查找，然后 APP 有一个 head out service 这个是用来说我这个服务器上有没有其他的。ATP 的 Alternative. 服务就比如说我通过 H p2进行了这样的通信。可能会有海外版的快客服务跑在80端口，刚刚有人在聊天里说腾讯会议的海外服务跑在80端口，那他开心就好11般来说 H DPS 它跑在443。所以 ht b443它如果开一个 UDP 的话，可能就是代表着它有这样一个 at p3的 quick。他跑来报名的就跑来报名吧，他开心就好。如果即使没有这个 HP 的 a service header，它也是更有可能就是如果你存在 ATP 服务，就相当于一个没有 ATP 服务的 IP，它就更有可能存在 at p3服务就是比如说你，你一个服务器上，你都看了 ap p3你没，没理由不开什么 ap p1.1之类的。

三教(00:53:20): 最后，他们在 QQ 购基础上好想1个工具叫 Q scanner 和他们就试图以不同的这个草案版本和这个正式的 quick V 一去在不同的 IP 上进行这样的连接的建立。最后，他们对比了这个 zap zap Z 有几个 problem 可以。看色馈和他们对比了，在外部当时的使用版本协商的方法和他们的方法的区别，那么这个认为 Z web 的这个版本形象方法不够精确，为什么，因为我们刚刚说有很多时间，他你你给他强制开启版本形象，他不理你。反正他们的结果他们在不同的时间里去统计到不同的草案的版本所占的比例。

三教(00:54:00): 我之后也是他们在23年，他对这个互联网这个实现就是你这个 quick 到底是什么，到底是谁写的，到底是基于一个什么样的实现，对这个事情做好一个测量。这个测量是基于他们之前的继续开发的，他们的思路有两个，第一个思路就是不同的实现，对于不同的错误给出不同反应，就是换句话说就是他们去蓄意的。制造一个报错，然后通过分析这个报错信息来进行这个 quick quick 实践的查找。比如说这个 ios quick 如果你给他发了一个无效的 T RSA P，它可能会回回这两个之一，然后它的错误码是固定的，对它可能会回这样一个。协议里没有明确的规定，这是 TRS 的就是它 TRS 层面的包。协议里面没有对的明确规定，但是它对错误号其实有一些规定，你看什么那么错误号都是一样的，这就是错误号是什么?

三教(00:55:02): 说实话，我不太记得这个1178和15年到底对应什么错误，但是它这个报错它是可以涵盖一个 message 的这个 message 可以就是大多数情况下是 human readable。基本上你可以通过报错的这个 message 和这个报错扣单来进行这样的匹配就是只要你匹配上，你就知道它到底是基本上大多数情况下，你就知道它是什么实现。当然有很多东西它其实不回你一个特定的 message，所以他们使用另外一个就是不同实现的 TRTRS 的参数的顺序是不一样的。就是 TRS 它是有拓展的，然后 TRS 拓展里面有一些专门的参数是给这个 quick 的，所以他们就敬我这个 T OS 的这个拓展的顺序和这个 transport parameter 就是他们这个通信的这个参数的这个顺序，通过这两点，就可以进行一些匹配，比如说他们，他的顺序先五一后43然后。它这一块就可能和 IOS 的匹配上，对于主要部署实验来说，他们的这个 TRS 的这个参数是长时间保持不变的，不会说你更新几个版本之后，这个东西就变了，你这个 TRS 模块可能写好了之后，有很多人一辈子都不会去动它。

三教(00:56:19): 所以他们就是根据这两个生物去进行互联网上的 quick 的实现的测量，我就知道每个 quick 它到底跑在什么样的东西上，使用最多的就是这个阿曼的这个就是阿大家都知道的这个 CD 包括后边什么什么。

三教(00:56:36): 不过 quick 它其实也是 google 自己的 CD 或者什么 RSR SB，它也是他自己的 CDN，就是绝大多数这些东西其实都是提供某种程度的 CDN 服务，再往后包括什么？什么什么什么这种乱七八糟，这个数量很少，基本上都是个人在用，然后有其实也有很多，大概有很多是他们自己识别不出来的。

三教(00:57:01): 这个是不是癌的，就是很多原因其实是有很多人在跑他们的闭源的，比如说腾讯自己，他其实腾讯自己内部想两个必然的配置我都不知道，同学最最想干的事情就是自己养股写几个一样的东西，然后他们让他们大逃杀。腾腾讯自己就有两个配合实现我不知道两个都是闭源的，特别讨厌。我还有开源，踢会格，我不要就是腾讯内部有一个叫什么叫高高高 server 是他们的 CDN 然后还有一个叫什么爱也是他们自己的可能是 CD 或者什么样的就是他们的 server server 端是不一样的。参数不太一样，感觉不是一个实践，但是他们内部反正做了一大堆差不多的东西，感觉当然除了这些实验，除了这些方法之外。

f(00:57:50): 散装腾讯。

三教(00:57:56): 请说。

f(00:57:57): 我散装腾讯。

三教(00:58:00): 确实什么腾讯微软这边也是散装的，我感觉大家大公司都是各玩各的，然后有内部养股，内部竞争。但除了这两个。

f(00:58:08): 感觉微软散装成散散装程度比腾讯还要高一点。

三教(00:58:14): 确实内部打架最喜欢干的事情，有微软特别喜欢干的几个事情，一个是改名，一个是买买来之后解散。我最后一个就是内部养股太太太喜欢了。用得好卡。不是先买来，然后改名改完名之后砍掉?

f(00:58:43): 就但是从反正那个东西改名改了好几次了，都把都感觉改改名了。

三教(00:58:50): ok 那我回到正题上来不了，微软就是除了正常松散之外，其实还有一个很明显的松，就是我跟他进行这个 ATP 通信就是我只要能给他建 ap p3通讯，我看他的骚扰，我就知道它到底是什么，到底什么东西。但有一个问题就是很多东西他不会告诉你他自己的 server 是什么，或者有的东西其实根本没法跟他那个通信就是他虽然表面上好像支持 APP，但是你看你给他通信，他会给你报错。

三教(00:59:27): 刚刚主要是现在已有的一些对 quick 的实践和部署的测量，一个比较搞笑的现象就是这个东西是他妈23年做的，然后从23年到现在已经过了两年，所以大多数东西都基本上进行一波大换血。虽然这里边的很多都是没怎么变的，但是除了这些之外，很多新的实践，有一大批的新的实践，其实他们都没测量过什么什么，阿里自己有一个什么之类的，就是他们就是那个参数东西没记，就是他们进了，但是他那个参数顺序可能变了，反正就是以他们的工作，你现在想要在。

三教(01:00:06): 再复现一下，或者在现有互联网再搞一下，你会发现这个东西其实已经不怎么能用，就是发文章的一个非常有趣的现象，就是发完文章之后，很多东西就没人管，没人维护，可能过了几年之后，你你你再想去复现在哪里发现根本复现没有。

三教(01:00:22): 发文章为道，然后第二部分是我们稍微讲一下关于 quick 的一些放大攻击的情况，就是这部分是我做的比较多的，就是一会也跟大家介绍一下我们的一些新的发现。就是大家应该都知道，就是我攻击方向，某个服务器去发送数据包，然后把我 IP 伪装一下就是 IP IP 整个报文都是伪装过的，然后你发给某个东西，然后某个东西帮帮你转发过去。

三教(01:00:49): 现在互联网部署了很多那种原地址验证的东西，就是对于某一个局域网或者它的出口会有一个地址验证，然后他把所有认为你你这里边不该出现的东西，它都它会滤掉，然后甚至可能会触发各种告警。我听说三大运营商在国内是有各种指标的就是它要保证说你，你的某些出口上面不能有来自就出口之外的流量。如果没有这个情况，他可能会进行线下的真人快打。那只是听说，但是大家还在都在做这个方法攻击的事情就是为什么，因为发法攻击可以发文章，就是你只要假装没有原地时间就可以发文章。

三教(01:01:32): 原地的原料多归多，但是你总总有一些地方，它是没有原地验证，比如荷兰是荷兰大多数机房好像都没有这种原地址验证，反正你就是通过这种方法，你就可以发发一些 IP 伪装的包，然后你就可以触发某种大大规模的。

三教(01:01:50): D. Os. 河南国外跟井盖没有关系。

f(01:02:00): 是欧洲阿姆斯特丹荷兰吗?

三教(01:02:02): 对阿姆斯坦不知道是他们把回传当生意在做或者怎么样。国外的很多运营商就是根本不管这个 IP 这个报文的到底是从哪来的，就是反正说他就转发他根本他们根本不管是内网还是外网。

f(01:02:25): 运营商，他是有责任吗?

三教(01:02:30): 我不好说，有可能是他们的技术实际上做不到，或者是他们觉得需要加设备需要加钱，他们没法自负盈亏。我觉得运营商其实是有这部分的责任在的，但是他们反正这个东西很难杜绝，然后你如果真的杜绝了，我们也发过文章了。万尼亚版权都不管，然后大家不知道看不看什么什么一些神秘的动画片之类，什么神神秘的漫画之类的，那边很多都是现在欧洲的一些没人管的地方。

三教(01:03:05): 我们使用这个 AF 就是这个放大比来衡量这个放大工具引入程度，比如说这个就是我发一个包，然后他回到六，比如说我发了60字节的包，然后他回了一个6666000字节的包，那这种情况下就是我使用60字节就能达到6000字节的这样的攻击的。工具的作用，那这样的情况下，一般来说，我们就把这个放大比叫100倍的发压比。

三教(01:03:29): 我在讨论之前，我们今天讨论一个很有意思的事情，就是能不能进行这个 TCP 上的反反射的方向攻击，其实这个大多数情

况下都大家都认为这个是做不了的，因为在 TCP 之下，你没有办法进行这个请求的来源，你没有办法完成这样的三个握手。因为你你自己不知道这个 TCP 那边的给你发回的话，你是没有办法去跟他进行后续的通信的，所以在这种情况下，其实你进行了一些一些简单的伪造，你可能也触发一些，只能触发一些 sequence 就是只能之间这个 S 或者是 ACK 的包，你你这样发发一个包，可能只能发发发100次，也可能只能回120，130或者200。在这种情况下，这个被道攻击的效率很低，所以大多数的大规模的 DD OS 都是放在反正是 DD OS 都是放在 UDP 的协议上，什么 NNDP 什么什么之类的这种协议，它可能达到几百甚至几千 Mbps 的在一些有问题的情况下，甚至能达到几万的放大，就是发发一字节的包，它可以回5万字节的包。这种情况下，它的效益是非常高的，因为有这些高效率的存在，所以导致 TCP 下虽然你也能达到可能23的放大比，但是这相当于几万的放大比太不经济了。所以一般来说，大家不用 TCP 来进行这样的放大。

三教(01:04:47): 就对这边就展示说这个 TCP 它有连续性的过程联系他同时进行了这个地址验证，所以我如果给到 ID 的话，只能进行极少量应答，因为大大头的通信都在地址验证和连接进去之后。对于来说，它协议上限制它最最多发送三倍于接入的数据，就是在进行这个连接建立和地址验证之前，我发1500字的包，他最多回我就是协议，说他最多回我3600字。不是这样，协议上说是说，不是这样，这就使得它和 TCP 命令一样的问题，就是你你用 quick 来进行攻击，它的性价比不高。

三教(01:05:25): 比如说这边展示了一些一些已有的工作展示的量，比如 NDP 在一些情况下可以达到500个方案，然后我们开始在一些情况下可以达到5万个方案。在这种情况下，他就是很没有性价比就做到三的，他们理论上说，协议上说最高三的发比就写的很不吸引人，所以大家。最开始对这个东西也没有什么研究。其实有也不仅是实现大多数情况下实现实验有问题，但是也有一些是不得不这么做，就是有些情况是不得不引入一个比较高发比我们一会讲。

三教(01:06:04): 我们先讲这个 TCP 的一个比较好笑的事情，这是我的一些工作的一个想法的来源就是 TCP 的中间件，会对一些情况做出敏感的回应，或者我们直接签，比如说就是 GFW 在一些情况下，会对单项的 tcp T tcp 通信做出这样的回应。

三教(01:06:24): 打个比方说，我们都知道这个互联网的其中一个特性就是通信往往是非对称的就是我从路径一达到某某个目标，然后它回回我某个流量量可能是从另一个物件回来的，那这样的情况下，对于一些中间件它。检查到了一个 TCP 流量，它不会去分辨这个 TCP 流量是不是来源于一个 TCP 连接，就是你如果不建议连接，直接往外去发，这样的 TCP 流量也有可能被这样的中间件捕获的。那这个中间件仔细一看。这个 TCP 的 pay 是一个 ADP 报文。作为一个报文里边这个域名就是一个不该放过去的域名，他会给你发发发三个 reset 回来或者怎么样，大家都知道 GF 是怎么干的，那这种情况下，我只要就是进行一个单项的容量就是单项，比如说这边展示了他们所使用的一些，这是已有的工作就是发票到21上。他们使用这个就是在 TCP 就是发发单向的 TCP 报文，然后这个 TCP P 的是一个 htt p1.1的请求，而这个请求的 host 是一个我认为应该滤掉的这样的 host。

三教(01:07:32): 有他使用这些域名就可以触发 GL 其他的中间件的这样的审查，然后他就会给你返回大规模的。大规模的这样的豹纹，然后它就会导致说从从从你正常用户就是如果你去数据触发这个他给你返回的东西，可能是或者是空白页面或者怎么样的。但是如果你去伪造 IP 以后发这样一个包，那你返回来的东西反正就一大堆，比你发发过去的东西要大得多。

三教(01:08:06): 然后其实他们也不只是加了有一些其他的东西在做，比如说美国在某某个大学，它有更敏感的，这个审查机制，或者英国也部署了一些类似的审查。然后他们反正他们做法使用不同的域名进行测量，然后他们返回的情况进行一个分类。现在某些情况下可以增大这样的反射，反正他们可能达到100多200多的放大比就是在 TCP 的单向流量上，通过触发中间件的审查来进行这样放大的增大。他们的方法，反正就是他们自己想一个东西来进行，这样通过收集哪些东西被审查，然后通过他们自己的内部的一个预算算法来进行尝试，对这个反馈的增大都使用 Z map 进行了这个 pro 进行这样扫描。去进行整个互联网扫描发，那么最后就是使用这种方法检测全球哪些地方有中间件，哪些地方中间件可以进行这样的单向的连接的攻击。

三教(01:09:15): 有这个东西，你想一下就可以进行阻断的观察，所以在他们的方法之下，就24年就是去年有一群人对他们的方法进行改进去进行全网的这个 GFW 阻断的观测，比如这边有一些比较简单的 GFL 阻断的。

三教(01:09:33): 场景，比如说第一种就是你，你进行这样的 DNS 查询的时候，DNS 查询被某个地方滤掉了，但它反反映一个假的 IP 就是 DNS 污染。然后第二种就是我们刚刚讲的就是 HTP 的 filter。你你发的某个东西有这个 APP，就是这种情况下，他可能就返回给你三个 reset，这种情况下他就会把你这个阻断掉了。然后最后一种就是这个 HTTPS 的阻断，就是大家都知道 HTP 握手时候会有一个 T OS TLS 里面有一个。字段是 SNISNI 是 TR 的一个拓展，就它的设计目的是什么？比如说我要跟某个 IP 进行通信。然后这个下面可能有好几个域名

，有每个域名所用的证书是不一样的。通过这种方式，你你通过指定 SNI 的方式，他知道你到底该用哪个证书来给你进行这样的回应。这个东西是明文传输的，所以这个东西对这个家伙可见他一看你这个 SNI 不对，他就直接把你这个流量给埋下来了。这就是比较简单的，就比较经典的三种 GIF 和阻断的方式。

三教(01:10:38): 你的哪种？厨房不？这个就基本上是就是被动阻断，或者是它有时候会有时候在这个阶段它会返回给你一些空白页面之类的就是你你以为你访问了没有。这个 TSSI 可见的情况其实是对 quick 也是成立的，就是虽然 quick 的密钥是加密了的，但是它加密所使用的密钥是固定的，因此和明明文没区别，就是大家都知道这个 GFW 有一个开源的模模仿的时间叫 open gfw w。现的 SNI 是直接可见的，但是现在的我们的结果表明就是。还没有对 quick quick 下手，你使用 quick app 端连接，只要你 DNS 北京上网已经实现了**。是这样的吗？我去年反正去年没有，去年我们做都还没有，还可能也没发，我也不知道。确实，那这边就我们说一下之前的情况，之前虽然 SNI 是可见的，但是还没有被下手。虽然它 SNIGFO 可见 ONGFLO 有一个项目也是可以把强调的，但当时我自己没有去抢这个东西。

三教(01:12:21): 同时，其实还有一个东西叫，就是为了防止第三方注入一个 reset 它是要求如果你想进行 reset 的话，需要服务端给客户端颁发一个 token，就只有 token 能用来 reset，其他情况下不能进行 reset。通过这种方式来避免进行 reset 的注入。当然我们知道有这个卡或者就是这是牵头的就是为了把这个所有东西都加密做这样一个情况，就是比如说你使用后这个 TRS 后就是被加密过的这种情况下。你你你用什么进行加密，就是你要进行一个 dns sdn ss，它也是一种加密过的 DNS 做过 DNS 查询，你去查到你要访问的那个网站的公钥，然后你再用那个公钥加密的行后或者加密这个 SNI。但是这个东西，它也是写在这个 tr s1个拓展里的就是你读一遍 TRS 你发现有这个拓展，你就可以直接把这个流量量给强调，就是建设单位目前对这个卡卡和 SI 的策略就是全都丢掉你，你只要发这个，我就不给你过谁你别用。

三教(01:13:32): 我刚刚讲那么多，你只是一些跟课没什么关系的讨论，我们回到这个话题。我们说其中一个风险就是证书和导致的犯大风险，就是这个人做的。然后我们刚刚讲说，quick 它的包有最小1200字节的限制，然后在企业有规定说在进行链接建议之前最多回。3600字节就是说，他把协议把他的这个方案被限制到三之内，当然勇敢。他们发现两个情况，第一个是证书链过长或者证书过大的情况下。服务端必须把所有的 TRS 协商都限制在三千六百三千六百字节之内，这种情况下很多时候是不够的，如果这中间特别长的话，他根本没办法把所有东西都塞到这三千三千六百字节里边。当时 OpenSSL OPEN SSL 没有这个 compression，当时 OpenSSL 不支持现在已经支持了，所以这个情况更好一点，就是它通过把这个东西压缩一下，就可以把这个问题解决。另外一个就是在输不动客户端响应的时候，服务端可能主动重新发送，比如说我给服务端发一个，然后他回我之后我不理他。ECC 证书的话对他们也他们对这个协议证书的。证书的协议组也进行了研究，就发现有一些更长的证书就更容易受到这样的攻击，可能用比较短的证书就好一点，或者是他们把这个证书压缩引入进来，就可以解决这个问题。

三教(01:15:03): 就正常说在 SSI 里边可能存在好几年就是好几年之前就有人想把这个东西加进去，但是也是到最近可能是去年或者前年才把这个东西终于默认进去。外一点就是在收不到客户端响应的时候，服务端可能主动重新发送，就比如说我给服务端发一个包，然后他返回我一个药包之后我不回答。之后只要我一直不回他就一直会假，他就觉得我没收到，他会主动重新发一遍，或者再主动重新发一遍，所以这个会陆陆续续。我去重新发，可能三四十个，你要不就他一直觉得我没收到，但是我就是刻意不响应，在这种情况下，它就会导致一个比较大的方案比是跟实践有关的，就有实践会主动的。

三教(01:15:52): 反复发送同样的包来避免你收不到，其中最最受影响的是那个 facebook 他们有现在他们有一个时间，他们那个时间会有大概40多倍的这个存放就是你给他发一个包，他回你包，然后你只要不给他，就会一直给你回这个包就直到回40多遍。这个问题跟证书没关系，但是咱们应该加一些策略，就是如果一直受到这样的情况，他可能说有人在故意攻击我们，那他反而那么一加。这个它是原因是因为贵的要求就是协议里面要求说我必须是不是说必须是可靠的一个对协议要求是必须是可靠的，那这个也不可能完全可靠，因为你回多少个都可能对可靠，它前提是两方都是善意的。不是协议里面说的是说在你完成了地址，就是完成了客户端的验证，你的 server 最高是一个 hard 限制是一个单倍，对，就是你超过三倍的人，你可以把这个链接直接扔掉，就是。你可以直接效果一下，你要等待客户端，手机上你要等待客户端重新建就是理论上它不应该无限制的，还是实现，有就是就在这一部分还是时间有问题，但是我们说这个可靠的时候。

三教(01:17:10): 我都讲了，这两两边的两边是一个良性的人，不会说我刻意不回，你刻刻意不回，不在这个协议的考虑的范围之内。这个对于前面这个问题，就是当时没有压缩这个问题，它是不得不超过三，就是它的证书面就是过长，它没有办法限制在三之内，这种情况我觉得还是可以理解的。那后边这个45倍的回应，其实就是他自己实现时间不知道在搞。那么另外，工作就是他们使用进行伪造对。300多

个数，那就是治标不治本，你你现在有正式的情况下，其实基本上可以控制在三之内，但是我觉得协议它可以有一个硬的上限来控制这个。这个地址验证之前的这个数据的传输，大家在你你改成数的其实也只不过是把一些东西从变得不把一些实验变得从从不符合协议改成不那么符合协议，没有本质上来解决这个方法工具的问题，只是一个话术，我觉得。

三教(01:18:20): 然后是他们进行了各种情况的伪造，其中比如说连接迁移的伪造，让他们发现这个情况比较严重的一个情况，就是地址验证只在你跟它建立连接的这个过程中有然后之后，你把它迁移到一个新的连接理论上，你再进行一次验证。这种验证方法就是我们刚刚讲过有。challenge pass response 来确定你之前都是同一个 IP 在控制，但是很多实践它在你迁移之后的处理有问题，就是有时候你没有通过他的后续验证，它也会在新的 IP 上给你发这样的。这样的包。就是他们发现就是在伪造版本包，伪造这个迁移包，在这三种情况下，其实是存在各种各样的方法风险的，当然这个图其实是为了他们问了他们的这个。

三教(01:19:09): 连接失败之后，服务端就直接把连接，一般来说，如果你联系失败，你就直接把这个联系三次就重新开一个。

三教(01:19:21): 这个图也是他们为了说是好看，所以他们这个图有很大的水分，这个300多是他们的这个好看标的数字就是他们在建立这个。你你之前那些报文都没算到这个通信里边就是只用了最后几个部分，就是为了好看，黑线是他们实际的结果，就你看300多实际结果其实是18但他们标300多，为什么？因为300多好听又好看。

三教(01:19:50): 玩家的小技巧，大家不要学。就是他我们衡量发的攻击的情况下，就是我们会说就是用整整个的链接就是我发给你多少，然后你发给受害者多少用你说发给受害者数据的总量就属于我发给你的数据的总量，这是我实际发工具的效果。他不是他，只是他这个300多是只考虑最后一个包，就前面其他东西我全都不算，只算我最后一个包一定要得到回应，那不是那可以很高，但是就不对，因为你既然给你你是发放数据的。就大家懂的都懂，就是发发文章的必要的牺牲，那冰箱也是热门配件，事情上就有，就他们还比较好，就是他们至少标一下说就懂行的看一下，知道这个实际结果是多少。

三教(01:20:51): 对要搞成指数的坐标都搞指数的坐标，对，就是这种情况是比较常见的，因为你你你很难在一个图里边给他全全。全全展示状态，因为有时候你看，比如说三三300多，你不可能真的画一个300。特别是有的发动机，它发射比能到5万多和三，你放在一个图上，你只能用一些对数的方法。

三教(01:21:20): 我们有一些新结果就是一些错误的配置导致的，其中第一个就是 cos 他自己的一个问题，就是广播地址有问题，就是你给他广播地址发一个包，他会在这个广播地址上回回128个包。这个文章是他们想一个报告来给我们讲讲我们的这个工作就是 C 它有一个功能叫 any cast 的就是 cast 的。就是他的想法就是我有一系列的服务器，每个都可以处理到其中到某个 IP 的请求，所以他们每一个服务器都可以宣称自己认哪个 IP 最最近的都可以把自己当哪个 IP 使用，就是最近的某个某物有可能把它定向到。可能把它对应到某一个主机，就是有一系列主题，他们每一个都可以处理到某一个 IP 的请求说你到哪个主机是你的最近的路径而变化的。然后这个使得他们有一个误配值，导致他们把外部发进来的目标是广播地址的包也进行了一个错误的处理。所以你给他广播地址发发一个包，他会把这个或者把包错误的转发给64个服务器，然后这64个服务器每一个会混两个包，所以你给他发一个包，他会回回你128个包，这而且这是这可以稳定复现，但是他们现在修复了，现在修复了就没有这个问题。

三教(01:22:41): 反反而非常搞笑他们就可能去年年终给他们报的，今年年终年底给他们报的，他们可能去年就休掉了，但是他们可能花了半年时间来写博客，就是大家非常喜欢的这个磨洋工环节。

f(01:22:53): 对了，我这边有个问题按照传统的 TCPIP 的。

三教(01:22:53): 就摸一下。

f(01:23:04): 这个网络的定义的话，你给一个网络的这个255就是应该严格来说是11111这个广播地址发发包的话是网络里面所有的，按照这个按照 TCPIP 的这个标准定义的话，它网网络里面的每一台主机都会收到吗?

三教(01:23:13): 这个钱也挺好。

三教(01:23:28): 一般来说，这个情况就是广播只应该用于内网，就是只有内网发到这个权权益的这个地址，大家会广播到每一个服务器上。而不说外部给他发一个255，他也是广播对每一个主席上就不应该外部给他发一个外部到广播地址发一个包，理论上网关应该把这个。

f(01:23:41): 是每一个主机?

三教(01:23:51): 这个包直接丢掉，而不应该把它转转发到每一个地方都发一份。

f(01:23:56): 这个就是看防火墙策略，但是如果说是从如果，但是如果说是从就是说最最传统，最就是很久以前那种很纯真的网络时代，就是说没有坏人的那种。

三教(01:24:07): 那对。

f(01:24:10): 那种没有坏人的那个网络时代的话，那按照这个标准实现的话，它就是你给一个网，你给一个广播地址，你给一个网络的广播地址发一个。

三教(01:24:11): 那种情况下可能确实是对。

f(01:24:22): 发一个包，网络里面所有的机器都会收到。

三教(01:24:26): 对我这个具体的标准我不知道，反正你具体标准也更新过好多次，我觉得可能在比较纯真的时候，大家比比比较有信任的时候，可能你你从来不发广播地址，它也会广播到内网的每一台主机吧，但是现在就。现在你外部发的广播地址它，你如果真的转化到每一个主机的话，可能就有问题。

f(01:24:48): 现在的广播地址这个定义有什么用?

三教(01:24:52): 我不知道这个现在广播地址一般是内内部在用，就是某个内网就是我内网发到广播地址，这个网络地址会广播到每一个主机上。

f(01:25:03): 那行吧，那会不会有的内网也会在第二层或者第三层的交换机上面就直接把这个包给丢掉了。

三教(01:25:13): 也不是没可能就是它内网可能也根本不属于广播地址就广播和多播在 UG 在 iii ip v4设计里边是比较失败的。大家都知道这个东西是比较失败的，因为整个把这套东西整个都去掉，就用了一道别的算。

f(01:25:28): ip v6i P v6的主播是干吗的?

三教(01:25:33): 对 ip v6，它重新设计了就是它设计上就可能是只能发给某一些木游或者某一些主机之类的，就不允许就从从策略上就从设计上就不允许外部的广播的访问。这个不在我们这个讨论的范围之内，我就这部分，就我们简单聊一下，就是他们当时会有一个就是当时有问题就是他们的 any cast 的配置有问题导致。外部发现的目标是广播地址的请求会发发给每一台主机，这个东西不仅是对他对这个 quick 有这个反应，你给他发发那个包都会被错过的广播到所有主机上，这个问题只是我们在处理 quick 的时候偶然发现的。问题他们认为很大，我第二个问题是我们刚刚讲过腾讯3500多。我不知道，但对我来说挺多的，再加上他们也不乐意再加上你，我还当什么博士生专心挖互动是吧，他是发发发。那其实还是很厉害。100多个128倍对你，你可能在成三个。我不知道我没具体去试到底能达到多少就是500那挺好，那其实还是要贷款。还是很大的，挺大的一百一百多位。

三教(01:27:14): 这个是我们第二个问题是我们发现有一些 quick 实现有问题，就是我们刚刚讲 quick 的包，它有一个最小12001200字节的限制，就是如果不到这个字节的话，你就要在后面加，就把它形容到它有1200字节那么大。然后我们发现有一些各个实验，它没有这个1200字节的限制，其中最典型的是腾讯自己的内部的实验，有这个问题，我跟他们报了他们假装这个问题不存在。腾腾讯大家都知道我们跟他们报过很多问题，他们就装死就是可能他们内部有什么指标之类的，你可以跟他讲了，他说我们知道我们的评估，我认为这个问题没什么危害，这个问题对于腾讯内部当然没有危害，只是外部有危害，所以我们就不管了。

三教(01:27:59): 就这个就是演示，就是这个558是这个，这是我给他们发的包，这个东西是没有填充的，它小于这个1200自己，然后他你看他又给我回回了一大堆。它这个话比最高，可以到接近40倍。就是腾讯，自己有问题，但是我跟宝马他们到现在可能没修。那也是我被抓进来。我解决不了我们自己的内内部的漏洞，我也解决不了你。对中心也不好说，因为国家运营中心很多时候他不会就是他对这个发放工具的处理，他很多时候不当漏洞处理的。

三教(01:28:42): 大丰，你属于一个不太有地位的地方，很多时候你去报这种问题，他们都认为这是一个不大的问题或者。他也不觉得过动，对他也不觉得这活动就是你跟他说这个不符合协议限制，他们说反正也不是不能用，就算不能用他们自己也不是受害者。这不是他，你不是可以拿他们去打别人吗？也算是受害者吗？那就不知道了，他反正他自己不觉得有问题，他们可能还发现我怀疑他们内部有指标，就

是这种波动率之类的，是有年度指标，就一年只能有几个过渡，一年只能有几个活动。我怀疑是这个原因导致他很多问题，就他们就知道了，也不管。不是没可能。

三教(01:29:36): 这个问题相对于第三个问题是一个很小的问题就是第三个问题是我们是今年年初发现的就是我们刚刚讲有一个版本协商的情况，就把你让他使用这个零零零这个特殊的版本号，就是这个版本号，在协议里边明确规定说你收到这个版本号，你就你什么都不做。但是一些实践会把这个当成不支持的版本号，待会再返回给你一个版本镜像包就是你给他发一个版本镜像包，他会给你回一个版本形象包。

三教(01:30:03): 那这个大家一讲就知道说这个东西可以就可以直接拿来构建，就是你只要伪造 IP 你可以直接崩掉这两两个有这种服务器是吧，你可以一次崩掉两个。

f(01:30:14): 我感觉散布整个 quick 生态。

三教(01:30:17): 整个过程就是各玩各的。这个 quick 协议本身也确实复杂，就是它的复杂程度，我感觉比我见过的绝大多数协议都大，然后就是它虽然是一个传输层协议，但是它表现出来的感觉已经很像一个复杂的应用层协议。

f(01:30:45): 那是不是我在想是不是稳定安全和性能，这就是不可能三角?

三教(01:30:53): 我不知道我觉得你再加一个资金投入，就是没有钱，大家不给你好好干活，就这个东西属于是。大大厂都在推 quick 的一个问题是，它确实比比这个 TCP 就是 hd p3，特别是在 CDN 方面，它比 hd b2性能有很大的改进，有带宽什么的都有很大的改进，就是或者他们运运运营 CDN 的改 quick 之后，据说能使用30%。真的假的，我不知道，据说能省这么多，那大家肯定很积极的推这个东西，然后上次他们做 CDN 的那群人，他们写的也是安全性最好的就是最最没问题的。各种功能支持也是最多的，就是 google 他们自己写的肯定是最好的，那我自己提的还有其次就是考文雅他们写的那个 QSHE 写的也非常好。

f(01:31:43): 我这里我刚才说的稳定指的是对于长连接来说的稳定就是指的是能稳定保持一个长连接。

三教(01:31:53): 那这个大多数市面上的东西都可以维持超长这个长链接的稳定。特别是比如说配合作是社区实现，他也做的很好。

f(01:32:01): 但是像这些稳定维持长连接的这种特性，像刚才说的感觉很容易被用于放大攻击或者地到死。

三教(01:32:14): 你你能跟他建立长链接，就说明你不是要用来去提长链接和 D，它是就是其实不怎么挨着，就是你如果跟他建立长连接，你是跟他实际的在通信的，那你如果。

f(01:32:18): 一半。那刚才说的，比如说你换了一个 IP 出口这种情况就是说两换了一个 IP 这种情况的，这个就是换 IP，然后重新建，重新就是说迁移连接这种特性感觉有可能被利用吗?

三教(01:32:41): 就是他刚刚讲过，就是一些时间有问题会导致你这个哪些迁移之后的地址没有完全验证，他就给你发东西，这可能有几十倍的范围，一会我们会讲另外一个给哪些迁移带来的问题。

三教(01:32:56): 我先说就是我们当时进行全网的扫描，就是他们大概我们发现有大概十个左右，有问题的时间，就包括什么，就是这个 nginx 和 O nginx 下游的，然后还有阿里巴巴的这个差额，它和它下游的。包括其他包括什么金山内部有一个 nx 时间的改版和字节跳动有一个实验的改版就是各个企业自己做实验的改版。而因为他们其实事实上在下游，所以他们当时没处理好的问题，他们也一定。

三教(01:33:29): 今天继承了其他的包括什么什么乱七八糟的 ERP，反正我们大概发现有将近十个这样的这样时间有问题，其中点名批评 ngx nginx 他们之前的这个。就之前他们的这个侧面里面写说我们会什么什么情况下发 CAE，然后我们给他们报了这个问题之后，他们连连夜把他们的策略改了，不故意不给我们发 CAE 点名这边这个字。就是他们之前的策略是导致这个 worker 崩溃的情况，不给 CVE，然后我跟他们报了这个问题之后，他们直接一刀切，所有实验性模块都不给发 CVE。然后 quick 是他们的实验性模块，所以他们就不给。

f(01:34:15): 然后不给 CVE 是哪边不给 CVE?

三教(01:34:15): Why. 考尔福亚的就是 f5上就是 CV 的颁发，它是各个对他们是深深地就是一般的他都会找他们。就你对就是它是就是最后解决方案，就是你只有没有，只有一个。对 f5认为实验性模块就全都不给 CVE，当然这个策略是他们新加的之前不是这么说的，就跟帮我问题之后，他们为啥不给我发 CVE 加的? 不给发邮件，他只是回得很慢，他最后确实会给而已，我不知道，我知道，反正这个电影就

很搞。就很多情况，你你你找的话，他可能就告诉你，你找 f5没有，我就说我之前已经找到我指定的 CD，但是他不给我，然后就我把这个跟他的这个 communication 都给你，然后让他去确认了，过了好几个，好好，你说的有道理，我会试试，现在这么干还行不行？可能不行，我不好说，包括什么什么，反正一大堆东西，反正最后就一个 CV 也没捞到。然后我们再讲一些其他一些乱七八糟的，就是你刚刚讲的其实有一个问题，就是就可以迅速的在不同端口上切换我的连接。然后如果我这有一个 NAD 设备，然后我本地在不同的端口上迅速进行这样。

三教(01:35:48): 我可以很快的把这个 NAD 表里边的所有空间全都跑掉，而且我们的研究表明就是一些地址，虽然它不是广播地址，但是你给他发一个包，他也可以，他也给你回很多个包，就这些包的这个连接的 ID 是不一样的，这也意味着就是可能一些 IP 上所部署的 NT，它也有类似于广播地址。那样的问题我们刚刚讲第一个问题。

三教(01:36:14): 另外就是。

f(01:36:14): 这个特性是用来干的，迅速就是对 quick。Connection.

三教(01:36:19): 靠什么说事情?

三教(01:36:27): 就连接迁移，就他最开始的想法是很好的，我如果有多个 IP 的话，我可以从从一个 IP 上无缝切换到另一个 IP 上。我可以使用，就无论是服务端还是客户端，我都可以把这个已有的连接几乎保留的情况下，把它从一个端口，一个 IP 上切换到另一个端口，另一个 IP 上，那这样。

f(01:36:48): 但是，但这个问题，但是这个问题它说的是说的感觉像是你从同一个出口出去，就假假如假如说。假如说我这个网，我这个电脑只有一个网卡，只有一个 IP，然后它在这个 IP 上面不断的切换这个端口，感觉这个第一条说的是这个问题。

三教(01:37:07): 它只它可以只切换端口。这没问题，因为我们连接追踪都是独立组的不同的端口，对他来说就是不同的连接。

f(01:37:17): 它为什么一个连接要不同的切换这么多端口?

三教(01:37:21): 他故意这么干，给你到那个 NIT 设备，就比如说我想把悠悠上的带来对，只希望就是为什么？因为 NAT 它的表它不会维护会合，就是它会对于很多 AD 设备来说，他是不知道的，他只知道 UDP。这样的话，我进行一个敏捷迁移，这个 NAT 他知道他可以把旧的表表上销毁，我就不再用了。但是很多 NA 它观察不到的情况下，它会把一个 UDP 它就算没有通信，它也会保留一段时间，比如30秒。这样的话，我很快的在不同的端口之间切换，我就可以把 NAP 表整个全部耗尽。

f(01:38:03): 第一个说的就是恶意的客户端的情况?

三教(01:38:08): 一个恶意的客户，他可以通过这种方式去用这种方式就恶意的把 NET 表里边的所有空间全部消耗掉。不是一个恶意设备我直接对着某一个任何 IP 的某一个 UDP 端口用用我自己的，其他的我乱发不是一样的。

f(01:38:34): 不过第一条感觉也不是那么容易的去实现，因为因为很多运营商的话，他对这个单单账户单宽带账户的话，它有连接数量限制。

三教(01:38:49): 好不好实现，我不知道，因为这个有人在提这样的铭文，可能这个是一个文章题目而已。

f(01:38:53): 因为现在有人说现在某些地方的运营商好像单宽带就是一条宽带，只给3000个连接，这个确实有点坑了感觉。

三教(01:39:05): 是有主要是我觉得对第一个意义不大，因为我你是不是 QSH 他说了，我自己开一个给你有什么去，我专门干这个事，对我觉得这个意义不大，就这个东西就是为了发文章去专门构造的，我感觉。

三教(01:39:20): 第二点比较有意思，就是他们讲说 QQ 通讯里边有一个就是同一篇文章讲的就是 QQ 通信里面存在一个资源比特位，它用用于 RT 就我把这个设你把这个生一再发给我，或者怎么样，就通过这种方式去测量这个。连接之前的 IDT，那这个功能是可选的，所以通信双方可以不使用咨询，在这种情况下，协议说应该把这个资源位忽略掉，所以就可以把这个资源位当做一个隐蔽信道，就是往往里边填一些东西，然后这个填的东西其实就对大家不可信。我感觉还是稍微有一点意思，比比上面那个有意思。ok 那我们其实讲了很多，我们冲刺一下最后一个环节，这是 quick proxy 大家都知道有个东西就是它是那免费的就你下载下来，然后你一打开你就可以用它免费代理。但是这个东西它在客户情况下，它就是你你你打不开它，为什么，因为因为因为它会因为它那个 DNS 之类的强了，就是你你需要开代

理的才能开，你打开之后你就可以把它给关掉。

三教(01:40:31): 非常神秘，但是他不要钱，你你要是没事干的话可以玩一下，然后他的这个实践也是考自己提了11系列的实践叫 mask 就是动物复复用的这个 Q quick quick 代理之类的东西。然后他很直面于这个 HTP。所以他们这个代理甚至是基于 HTTP 的，他要把数据包包裹在这个 HTTP 的 data 中，然后 HP D 是他们单单独提的一个新概念，就是把数据封装在 HTTP 报中，然后再用 ADB 来承载代理。

三教(01:41:04): 这么一搞，我觉得很奇怪，就是你明明可以直接在这个上面跑代理你为什么不要再要把它包裹在 APP 里边，不是不知道为什么他很喜欢 ATP。然后这个就是一个前向代理就是它不会像后边之后的连接，就包括这个真实的 IP，然后使用这样的 method 就是 connect 这样 method，然后进进行这样的升级就是把它从一个简单的这个 A PHP 连接，然后升级到一个。

三教(01:41:31): mask 代理连接这个东西其实是它从 ap p1.1到 ap p3都定义了这样的一套规范，就是你你你可以在 ap p1.1上跑这样的代理，但是好像没有人这么干。

f(01:41:34): Thank.

三教(01:41:46): 我除了这个之外，还有一个社区实践，大家都知道这个 history 就是他的 VR 是跑在这个 quick 上就，但就比刚刚那个放屁好一点，就是它虽然也有 at p3，但是 at p3是他伪装的一部分，就是他假装自己这个一个服务器，然后你需要通过一个特殊的 pass 就是，然后把后台设置一个特定的值，然后把然后。衔接一些，填一些你拉好的值，然后你然后之后你跟他自己就不是 ht p3了，就是他的，他自己代理的这种单独的协议以后，你就可以在这个协议里边进行这个 proxy。这个东西，你你之前的流量其实是就是在家你来握手的这个阶段就是你看看上去就是普通的建立连接，然后你你真的去访问它，你发现它其实它只有充当自己是一个阶端服务器，只有这个特殊的情况会告诉他，其实不是。但是你只要他验证，不过他就会正常的回应，你就是可能大家忽略或报错怎么样，他就不会暴露自己这一个这种代理。

三教(01:42:53): 他会假装自己是一个 ap p3，但是实际上我们的研究也发现它不是那么假装就是它存在一些情况可以帮助它可能不是不只是 ap p3。它有三种伪装的方法，它分别是这种 file proxy stream 就是 file，就是当成一个静态文件服务器，这种情况下，你甚至可以当那种之类的这些东西用你你提供静态。

三教(01:43:21): 然后，然后。这用你们有人跑这个吗？对你你你甚至是你，你甚至可以把它当什么当代理，然后好，但是没伪装好。然后它是基于这个构的，然后他然后第二就是对第二种返回一个固定的字符串，这种情况下，比如说你你你无论访问什么，他会给你说404这种情况下，就是你看上去就像一个404的服务器一样。但是在这种情况下，就是在搞反向代理的情况下，它是它就会，它是可能在某些情况下，他会暴露自己。

三教(01:44:02): 不是一个简单的就是比如说他默认是这个他默认他建议你是什么是 newscombiner 就是那个 news 他那个服务器是 nginx 的。但是如果你去你你你访问一个使用 proxy 的这个函数。它是会够的，就是在这种情况下，你你的你的字段，实际的 server 字段和它的这个和服务器它不一样，就是你看到的情况下，你这个会提供它为。被 reverse 那个服务器的 server 字段，比如说你配置到 news combinator 的时候，你去访问，给你的这个 server 字段是 NG。但实际上我们刚刚讲过，有一些技术可以采集它的这个 TRS 拓展顺序，这个顺序对应的是 go。那这种情况下，你你你提供的 server name 和你这个实际的 server 不一样，这个情况下就有可能揭示说这是一个 history。所以我建议大家如果真在搞这个的话，就最最最好。就是开。就建建议说最好不要就除了这个之外，他还有一些其他的特征，比如他很贴心，他会告诉你他一定会提供，就是默认会提供这个，那你，你可以改一下源码把它关掉。

三教(01:45:20): 但是相较于一个普通的服务器服务和服务器，它不会提供，就是说只能说你你协议支持的太好，有可能会暴露一些不该暴露的信息，其他包括如果你看到这个 traffic states 你你你去扫一遍。这个 traffic states 是一个简单的鉴别。你你一扫就能扫出来，在某个高端口上可能有一个可以 go on authorize。然后你发现这个服务叫443是跑跑一个 nginx，然后一个高端的那这也可能报大，但是这些都不是你通过这个 DP。叫什么 inspection 就是你不是通过 GPI 能检测出来的，就你需要主动扫描，虽然现在也有一些主动扫描的东西给他可以主动扫描之类的。但是这些就是它不是简单的怎么出来的，而且它其实会有误判，所以在这种情况下，就是他我也不好说会不会以这个思路对他下手，只是跟大家说一下我们的一些一些新的发现，就是他在一些情况下就可能会被识别出来。

三教(01:46:26): 为什么这个这玩意就是那个默认还给一个 news 是可访问的吗？我不知道，我不是你就我你你你这么用的我还真不知道，因为大多数情况下我都是开着代理，不可能说。不可能说宽带看一下能不能。感觉给这个默认值就好，好像当年说给一个默认值用完的，然

后次日被被什么被被那个很兴奋的人。

三教(01:47:01): 要两有。ok 然后最后一点就是 quick 其实它除了 ad b3之外，它有一些其他的服务。这是我们最后一个部分。我简简单提一下，比如说 DNS DNS 就是 DOQ 它现在是一个比较成熟的标准了，就是现在已经有东西在跑这个。然后有一些工作在进行全网的这个 DOQ 的识别，比如说222年的一个叫，就是当时还在草案就上，这个是我们组今年刚发的。现在可能还不能看，现在可能还没好，我这个文章也是讲的是就去年到今年做了一个关于。关于这个 CDO Q 成为标准之后的一些一些情况就是它部署什么，然后实现分别是什么，然后它的证证书的生态是怎么样，这是我们组。这第二个是我们组做的。然后他大概意思就是说传统的 DNS 它有各种各样的问题，然后虽然用了 DOT 就是 DNS TRS 进行这样加密或者 dns R dps da 这样加密，但是这会儿 DNS S 但是他们仍然受底层协议的限制。所以他们就专门说我们把这个跑在 quick 上，能不能对它进行一些性能上或者各种各样的改进，然后他们主动测量了一下，发现这个 DLOQ 的采用率在逐步的增加，当时还在这个标准化进行还在推动，现在基本上已经完完成了就做一点，我刚刚讲的就是握手时间比预期长，它有的时间会就会一定会给你回一个 token。

三教(01:48:32): 这种情况下，你跟他握手是2R TT 的瞄，没发现这个强制执行。我们最后一个就是说，DODOQ 它比现有的这个加密 DNS 要好，是一个更好的选择。

三教(01:48:45): 所以当然我现在试一下，我感觉，反正国内好像 D OK 的部署就没有太入网环境，没感觉到 TLS 的优势，就是现在的一个技术的发展的一个倾向，就是大家会越来越倾向于觉得网络是好的。就包括什么新新能源的控制之类的，他们会假设这个网络是好的，所以弱网的情况下，可能它的优势确实不大，但是在网络情况比较好的情况下，他会他的表现可能就比比以往鞋要好好一些，我好很多，我也不知道。

三教(01:49:16): 这个图展示了当时测量的是221年22年测量的这个 DQ 的这种情况，比如他们跑在这些端口上，就当时也是不同组合的这个占比是一直在变化的，当时也是这个当时部署的这些，我当时你看中国还是有一些部署的，但是中国。交给的部署我印象也和世界不太一样，但是我不记得是比世界更好玩，比世界更差，可能是可能比世界更好。但是国内这个 AB 三的部署其实就很少，而且 AP3其实一个问题就是它和现有的这个代理不兼容，就是你你想跑什么，你你用什么卡之类的，你没有办法在上面跑一 B 三。其他什么都是 TRS 都是 TCP 的。所以国内的无论是使用上还是这个部署上，其实都比较少。你如果想使用你首首先你得访问一些有。你要使用你你你要使用 APP T，你首先你访问有就部署的东西，然后部署 ATTP 的东西绝大部分都被强了，所以国内可能就根本没有这个使用的需要。

三教(01:50:22): 那以上就是今年的主要内容就这些没有大家如果有什么问题的话。交流一下。我这没接受的很有人问 TLS 有没有上海AP3的计划？那回答是为什么要上没有必要，但是上海之三的话是不是能减少一些带宽或者怎么样?

f(01:51:02): 然后怎么说，刚才下面感觉下面同学的那个声音好小，都听不清。收音是基本上都收不进去，什么声音?

三教(01:51:19): 我先看一下，就先生聊天，里面有人问说这个丘纳米尔有没有上 a3的计划，我下边的回答是为什么要上 a g3，我不知道，就是如果没有什么镜像站的大部分用户其实都不是那个都不是这个浏览器大部分都是一些就是下载工具，或者说都不是下载工具，一些简单的。这个包管理器他们通常不支持部分先进的协议，通常来说有 HTTPS 已经不错了。如果哪天我们能看到的，能够原生的比较好的支持。quick 我们可能可以打开，但是这个我觉得也不是特别必要的事情。对然后现在这个 AP3的最大的用户还是 CDN 因为你 CDN 是 CDN 分发，你可以用自己的工具。自己工具去搞，比如说什么 switch 之类的不是 Switch Stream 之类的，他们可能就已经上 CDN 了算一个。第三就是他们确实能省省这个容量，而且他们这个东西他客户端是他们自己管的，他们可以自己决定。在客户端里面决定说用户用什么，在这种情况下，他们可以进行双方的适配，但是经销站增压站其实管不了，大家在用什么来下载我们的。

三教(01:52:47): 我还有个两年，其实你说资金环境面面对面对的用户，就我们华人的白。这其实还是刚刚那个题就是你你刚刚说 CDN 能提到30%我觉得是不是很大的因素，其实是因为就 CDN 面对的这种浏览器客户它可能是比如说多个小文件，比如说加载一个 JS，然后还有可能有 C 之类的。但是你像我们传统，不管是进站。那你前面举的这个 steam 的例子，其实都是 pillow data 就是你只要上一个正确的压缩算法，我觉得可能是不是？对就是你的大文件 ISO DEB 之类的都是压缩过的就是我的意思对于有这以数据为主导的，而不是说各种什么标题头这些东西为主导的，是不是可能，其实它的那个压缩比例没有那么大，但是我不太清楚你那个30%的数据是什么情况。

三教(01:53:46): 得到了，就当时你提的时候，我就很好奇这个它有没有对比不同场景下的这个压缩比例，那个版本也是我凭印象的就是我不知道具体是多少，就是我印象中之前看过一个宣称说使用这个之后有多大的改 2007年有一篇文章是讲这个的，但是具体内容我不太记

得。反正我记得其中几个比较重要的数据，可能说他们现在有百，就当时17年可能就有25%左右的流量是跑在规格上的，就他们说这个降低了多少的延迟，然后市场多少的带宽。是一个内部视角提的这样一个情况。具体内容我不太记得了，如果你想知道，我可以找一下一起。这是我们17年那个？

f(01:54:28): 我这边还有个问题，就是说以后的 quick 有没有可能成为一个真正的跑在这个传输层的一个协议，就是和真正的和 UDP TCP 跑在同一层上面，而不是一个基于。基于 UDP 的传输层协议？

三教(01:54:48): 不太可能因为中间件你你没有办法确认换中间件，看到你，它的传输层协议是什么？是写在 IP 层的？就是 IP 层有一个 IP 层面，有一个字段，它那个字段会有一个标号码给每个协议分一个标号，然后很多中间件对于非 UDP TCP 或者他们认识的那个标号。就直接丢掉了，你没有办法劝大家换成员。

三教(01:55:19): 这就和 tt t21123不可能宣称自己是 tr s123了一样。他想清楚自己是 TRC。

f(01:55:23): 就是很多那个很多第三层的那些就是第你你就怎么说。很多那些运营商的那些网络设备会识别第四层甚至更高层级的一些一些一些数据，一些数据不好使，然后。不认识的就直接丢了。

三教(01:55:45): 他还是这个他的传输层协议是什么？是写在 IP 层的之前网络层就在这一层有的中间件就会把它丢掉就。甚至没有更往上的事。

f(01:56:00): 那不能进行 GDP IP 那边不能进行一些相关协议或者一些更新。

三教(01:56:11): 它更新，如果想把它写成一个新的协议号的话，肯定是能写的，但是问题在于，很多中间件你不可能去更新，他们就是很多运营商不一定是国内可能国外的很多运营商也是的，就很多关键组件它可能都是。欠维护也没人敢动这种中间件当时设计的时候可能就设计的不那么有长远考虑，他会丢掉他不认识的。IP 段它所标的那些传输层的包就是他一看这个协议他不认识，他就直接丢掉。

f(01:56:43): 那长期来看的话，以后有可能换吗？说以后就是永远只会跑在 UDP 上面。

三教(01:56:49): 不可能有一天突然全瘫痪，所以就所有设备都会取成一点点坏，最后就是这件事，永远办法。对就不太可能就是很多人家他就不愿意画，就是就大家难道不就一个东西能跑就不要动的，运营商也是这么想的。

三教(01:57:13): 我问一下关于那个就是他其实我感觉这个设计了一个非常的这样一种机制，但是就它的它真正带来的收益有那么高吗？就是它已经考虑了，说我把建立连接的这个 RP 定的降低，但是我这种情况下，我还是在设计一个非常复杂的。机制去让他去 migration 是何必就是为什么不直接就让他重新建立，或者说他这不是难道不是干了把相当于取代上层该干的事，他去想方设法去承包上一层下一层的事都把它给搞进去，到底是怎么？

三教(01:57:54): 我想这事对我也不知道你问这个问题，我确实不是我能回答的就是他们当时设计的时候是怎么想的，我没有办法知道，包括我们刚开始就说他可以有64位的这个64位。长的 token 的长，就这些设计到底是出于一个怎么样的考虑，我觉得我们现在不太好知道我们发邮件问一问他也不是没可能。我先把这个聊天就打开一下，就不是没可能，但是就这个 connection 就是他当时想的是非常好的，就是我如果要 CI P 或者怎么样，这个 CI P 它其实是双向的，就不仅客户端可以进行 connection 服务端也可以要求进行 connection。

三教(01:58:41): 就把它可以让1台机器切到另一台机器是有可能的，就是如果两个机器它们之间的共享状态，它可能从从一个一个一个比较远的他就他们 I P 一样，就是他们都可以成，或者他们 IP 不一样，或者怎么样，就是他们或者 IP 是 IP。就他可以允许服务端，也可以允许服务端去，从一个他认为你最好不要从这个 IP 走，换到命令你，你最好从这个 IP 走，但是设计归设计实现归实现它的设计想法肯定是好的，但是实际就是这个特性。我现在我都不知道有没有人用。这个数据我们也拿不到底现在有多少东西搞做出来这个东西到底带来了多大的，这有没有改进，肯定是有的，但是这个东西我觉得做实验也做不出来，然后要有多少，migration 我们现在也不可能知道。我们也吃不动了，我觉得这个事儿是你 quick 不去擦这个屁股下面人家链路层早就擦了几十年的屁股了，你从北京坐个高铁到上海，一路上 IP 动都不动一下，你想这下面有多少的东西在替你擦屁股，就我觉得网上转移是一个正确的，你不可能一步转移到应用层，那就只能由传输层再给你兜。

三教(01:59:54): 真的可以做到 APP 可以天怎么做到，那就基站一路看到它最后就是一个 tunnel，你连到任何一个基站上，它都是给你打一个 tunnel 和你的那个 home server 的。所以。你人在国外为为什么还是上不了 google1个道理吗？电话卡就真是就是这。

f(02:00:16): 但是国内的这个移动网络它都是那个接入地，他都是接入地漫游，就是说流量都不转发回你你的那个开户审了。

三教(02:00:27): 不，你这个是你，它有几种不同的方式，如果你1个连接一直不断，比如你一个电话 vol T 一直在打，那显然它是不可能。就是他不会给你断开，因为没有这样的 handover 的机制，所以它只能保证你的连接一直是一直同一个对一直是那个 IP 你你下次坐高铁。

f(02:00:47): 但你你坐高铁的话，你你基本你几乎不可能保证在跑350的情况下，然后不掉话。

三教(02:00:53): 你下次坐一坐京沪高铁，它一路上都不断的对他们现在练车上都有那个 RU 和 BU，所以能够帮你做一些 mesh 那样的东西，所以挺高级的。

f(02:01:04): 不是车上车厢就是车上它都有天线了吗?

三教(02:01:08): 不会因为你接入的是本地累积有一些这样的实现，那这个就扯远了。后面这个事情，大家想做很多年了。ok ok 我们还是一个看这个 quick。

f(02:01:25): 但是电话的话，现在电话不也都是 IM S 承载了吗？那好像。接入地变一下，然后但是也连到之前的，但是也连到之前的那个 I M S 服务器也不是不可能吧。

三教(02:01:48): 是，但是它本质就是一个 S IP 的连接你的 TCP，你的源地址改了，你就没有协议上的支持，那直接就掉了。就是这个问题就是解决了也不叫解决，就是其中一种机制来缓解你这个问题的。

f(02:01:48): 你。你。

三教(02:02:03): 这要求你同时可以在两个 ID 上通信这个东西来说还是不太现实的，他可以就是在旧的台机上发一个20。

f(02:02:05): vo lte vor 它没有 vo ltv onr 的这个应用层，没有考虑到这个设备端的这个就是终端的这个 IP 地址切换导致的这个导致的掉线。他是说他直接就是说。

三教(02:02:24): 就他当时考虑很多协议设计的时候，不会考虑你 IP 有变化的，因为它底层这个后面的这些机制，我理解它就比较上面就比较随意。

f(02:02:33): 但是我觉得像那个 IM S，特别是通话特。

三教(02:02:36): 我觉得这个问题我们可以再讨论，因为这个问题我们再讨论一些哲学的问题，不讨论具体的这个协议细节，可以说就是有一个草案就是，但是那个草案已经死定了没。

f(02:02:53): 但是在通话语音会话这种工况下的话，他是会移动的，现在移动的情况还是很频繁的。

三教(02:02:53): 感觉怎么可能是有已经 excel 了。这个问题我们以后再讨论，先讨论一下其他同学的问题，我们先聊。然后第一个，有人说这个 quick 就是防火墙，对确实因为因为这个问题的根源是你没有办法区分，就没有办法简单的区分这个东西是 UDP 还是 quick？我觉得主要这个问题根源就是这部分如果能把一些中间件更新换代一下，让他去检测 quick 的那个版本的部分，就是但凡他做到 quick 版本那个部分，然后对一些特定的字段，对这个版本去放行的话，这个问题可能就好了。当然我们之前说。中间件你一旦部署，你可能就不太能换，特别是有个中间件，它是纯硬件的中间件。对于纯意这件事，你你你这句话可能原原厂商都捧，我就没有人了。

三教(02:03:52): 就这个就是中间要替换是很困难的，就会有一个一个问题，就是他最最开始的一个一个问题，就是他跑在一个他跑在 APP 上都就这个问题也是妥协的结果，但是它也带来了很多，我们现在看就是。认为他跑在 APP 上才会导致出现的问题，有可能实现多进，就是有一个的 draft。那个 drop 的我估计可能明年或者后年可能应该能成为 RFC，就是他肯定能成为的，只是时间问题就是一样，他总会成为的，只是时间问题。

三教(02:04:29): ISD 希不希望可以推广这个事情，我没法知道，我觉得可能 SP 自己也有一些考虑吧，就是如果在某个时候，他需要把配合作为某个政绩的话，他会大力推广。如果他需要把作为政绩的话，我们之前提的什么什么问题都会解决这些事全都换。如果他希望把这个东西作为政绩，但是我估计那天可能反正不会那么轻易的到来。

f(02:04:57): 你说的就像 ip v6现在的 ip v6这样的吗?

三教(02:05:01): ip v6现在是作为政绩的推广。

f(02:05:04): 对我的意思就是说，你你的意思像现在的 IPV 应用的现状一样。

三教(02:05:10): 这个我个人觉得他把 quick 作为一个政绩推广是不太现实的，还是要好歹他解决一些比较根本的一个问题，我觉得亏的就是 TCP TCP 也不是不能用，而且我感觉带来了很多新的问题，似乎不知道还是。就是从从从你大公司的角度来讲，你如果是什么 CDN 分发商之类的，然后你你又能控制用户的用户那一端的一些东西，你可以控制浏览结果，你可以控制一下 app 那这个情况下你可以搞会，就是这种情况下，他肯定是对你好的。那对于个人用户或者是你，无论你的个人用户是希望使用还是个人用户，希望部署这个都不是那么友好你，而且会带来很多新的问题，刚才说会带来很多新的问题。

三教(02:06:00): quick 它本身一个很复杂的协议，就是它的复杂程度就是我还记得我前年就开始研究这个东西。我当时我看了半个月，一个月的英文文档，我还是没看出来。我当时给那个本科生的网员出了一个 quick 的题，然后被吴老板逼了。吴老板说太难了，太太不会真的太复杂，它甚至不像一个应用层协议，他甚至不像小层协议，他感觉。感觉是很像应用东西。

三教(02:06:33): 这就写一个对确实 TTRS 也很复杂的 T tct rst rs 本身就够复杂，然后它有压力拓展，然后每个拓展反应，然后又有一个专属于 quick 的拓展。有 EECH，反正又很复杂，现在我就没提，因为因为我因为我也看了，我看不懂，我只知道这会直接办掉 ECH 你两天发 ECH 直接给你丢掉。牌的非常不符合那个皮肤最高 quick，为什么要给 SNI 做一个假加密，我也不知道这是一个你如果能直接上 CF 的网站那。那可能他还没有，就是这个 GFO 会丢掉 ECH 也是我之前在哪看，就是国国外有一个专门研究 GFO 的组织，我忘了叫什么。反正他们还有一个网网网站，反正就是他那个网网站会经过他们对 TF 就对加的新发现之类的，对这样对。我印象是在那上面看了说 EECH 是给办了。对我还看看那个时候为什么我也觉得 EC 也是难以办，但是你你默认在这个拓展，你如果让他们给中国用户单独写一个写中中国定制版本，这种事情也不是。没出现过，那可能就是国国内不用国外用之类的。

三教(02:08:13): 我们还是还要就是他为什么要给 QUICK SI 做一个假加密就 quick 它其实不是给 SNI 做一个加，它是给你找包的整个 P 的重要加加密就是为为为什么他要给 quick quick，为什么要给整个做一个假加密，现在也不可靠，他可能就是为了宣传我们这个东西有多么安全，我们加密是端到端全全过程加密。全过程民主一样，全过程加密有什么区别，对就没什么意义。就是首首先它能不能做一个正确加密，它是可以做的，但是你做一个正确加密有一个问题就是你，你的公钥从哪来?

三教(02:08:57): ECH 尝尝试解决那个问题，我用 DNS 获取一个公钥，就可以加密整个 ECH。my socket。他可能。他的宝宝到底所有东西再加一个？你给我发了就有一些有一些知识最多咱们洗涤洗洗还有什么作用？那就快压力也可能是为了救自我觉得你说的是有一定可能。

f(02:09:40): 抱歉，刚才提问的，我没听清。

三教(02:09:45): 刚刚下边的一位同学提供了一个我觉得我要重复哪部分没事儿，这个同学你不用听清每一个问题，对他刚才说的是。

三教(02:09:58): 类似这样的协议，为了防止自己的协议被误认为其他的协议，所以他就用简单的方式混淆一下，基本上就是这么一个想法，但是主要现在的想法是他这个 quick 能和什么东西混淆的感觉，好像也不太能和谐，为了把明。就是不直接以明文呈现，对我感觉也是有可能的，但是如果你为了不被简单的不是匹配，你也可以用一些掩码的方法来盖，而不是你什么都弄进来，这边又有什么，我看一下我这边有一个过程，我这边有一个完整的过程，我直接贴在这。你没必要用什么做什么什么这个什么什么，然后又有什么什么 AS 又什么 GCM 什么乱七八糟，什么玩意都来了。这是为什么？如果是为了避免简单的简单模式匹配，好像没有必要整这么复杂。我这一点，我也没有搞明白，就是我不知道他当初是出于一个什么考虑做这个东西，我只是我写这个东西很难受，就把这个过程实践了一遍，因为我在复现别人工作，我自己做一些工作，我这个东西复现了一遍，更想死。他感觉只要给开发者千万或者怎么样，我也不知道可以写邮件问问 I ETF 的人在干什么，但也有可能是为了他把整个过程给做成一种可以批量处理的模式，就是一个函数写完所有事对也有可能，但是现在反正我觉得这个不可考，主任你给他们发邮件问。

三教(02:11:39): 我们只能做一些猜测。我感觉反正我们的所有猜测说风力都不是那么像。反正他就撞。还聊起来了？ quick 是不是也有可能这个是为了方便后续的实现，可以把这个值的 thought 换成一个位置，也有可能关于这个 quick 我其实想我觉得这事我有点想吐槽，就是 quick 这个事本身就是为了把你的流量控制这些东西从这个 TTP 站里解放出来，然后先回这回再给你实现进去。虽然说课里面确实可能有一些需求，但是我觉得。我是觉得有基本的 T OS 之类的，也就差不多了。他到底为什么要就是他如果想 offload 的话，那应该 offload 到别的东西上去，而不是 kernel 内内核里边这个我觉得还挺奇怪的，就是因为有的这些。但是映客，你就可以直接一个 copy 的这个题了，

很神秘。

三教(02:12:49): 很神秘我他是。就等于是拿你自己穿一个 APP 不放。举报什么什么国庆？很神秘，就是 quick 当时专门把他的协议就是控制什么的单独灭了一个 r9002。当然他也没有，就是在恶意的情况下，我们刚才讲了恶意用户可以恶意的控制发包送给收包送进来。就假装丢包之类的，通过这种方式来搅乱整个 server 的策略就是网网站有人做过这方面的工作。

三教(02:13:36): 但是我还是觉得这个东西就能进内核，就像什么，你就像什么之前，大概五年前十年前，大家在用什么什么之类的时候，大家要换一个 BBR 的控制，要把这个内核换掉一下。

三教(02:14:09): 他是怎么的?

三教(02:14:15): 那我们这个线上的环节差不多就到这里结束了。

f(02:14:19): 我还想问个问题。

f(02:14:28): 就是前面提到就是说 quick quick 的话，至少服务端那边有非常多的实现，我这些实现能相互兼容吗？因为感觉这些实现的话，里面有很多细节就相差过大。

三教(02:14:47): 这个服务端不需要相互兼容，是服务端不会和服务端建立联系，100，190，190，190。就是你你你就是做了一个服务端和另外一个服务端建立他也是把其中一部分当做客户端独立出来，然后客户端就你只需要保证客户端和服务端11匹配。

f(02:15:07): 或者说这个服务端这么服务端有一大堆这些东西，然后客户端的话，可能我想会不会实现，也实现的话，客户端的这个 quick 实现也有很多，然后。相互排练。

三教(02:15:26): 现在客户端主流的实现只有一个。

f(02:15:31): 行吧。那挺好的。

三教(02:15:35): 你说的这个问题，不只是网上各种东西上都是一样的，其实就包括现在各种 HTTP 就不用带 S 的 HTTP 服务器之间互相。差异都很大，所以说更不用说这类复杂的协议就是其中几个可以说一下，就比如说这个是既有服务端有客户端的，比如说这个 python，它也是既有客户端有服务端的，但是基本上没有人把他们当。客户端很少有人把他们当客户端就是做研究的话，你可能会需要用用这些，但对于大多数用户来说，他你你亏本还只是跑在一比三上面，你用浏览器就说到这里，我突然就是建立管理。建立完了之后，两边不完全是几乎是就是他们都他们能起到作用几乎是一样的，但是有一个例外就是 reset 只能从服务端发到客户端。

三教(02:16:29): 我印象中是什么阻碍，就是大家只实现服务端，而不用一个抽象全做出来，我不知道，就有的是全做好的，比如说这个客户或者是对他是全做好，有的我怀疑可能是因为对于很多大公司而言，他们的目标就是写一个服务的，他们就没考虑客户端浏览器。可能是这样，比如说像那句话，最最大的用户，像什么 ios google 什么这些都是 CDN，确实有一个项目。还有好多上坡的块，不过他去买，他也就去买，所以那个矩阵长啥样子好看吗？我们看看能不能访问我们不知道有没有被禁。你你你是电脑在分享屏幕，你可以用你的电脑直接访问我，还是用这个吧，是这样，我们能用 google 搜索吗？不能吧？我们毕业了。是哪个肯定打不开，这看着很像。打开不知道他的静态资源都哪来的，我们有。是打不开。吃干饭的。他几乎打开不行，我们还是分享屏幕！我要分享屏幕，我要分享我这边的屏幕。反正就大概这样吧，你可以看到他们。

三教(02:18:48): 我家有一个脚本，因为可能没有服务端。对。至少你应该认定你认定是觉得我不给人几个字，就只有我直接这里面这一个。是他支持的东西，一个是每一个块就是一个不是每小画画是一个优越的太阳？多久没跑过，后面就没跑过。我看一下这个 chrome 就是关键，没考就这种东西感觉就31比三对酸奶是酸。这个数列不是不知道是在不可能，我不知道。点点开需要的，我不好说可能是会有的情况下可能就你说有东西没有。这怎么还可以画画画笔线，怎么这么高级，我不知道，可能他们的知识没那么好，或者他们无头魔人家不支持。

三教(02:20:23): 我不知道刚刚你要问他。看上去只有一个客户。我不好说，其实基本上 QQ 它是有客户端的，它可能是有没有他自己跑，还有他自己跑，自己还能跑挂的好面粉。做个车子吧，不知道有没有一个全都是绿的，是没有，我看着没有一半一半红，就是这个是我们新新新的伟大协议，你们有没有什么伟大的协议?

三教(02:21:12): 那我们线上环节大概就这样还真有。先把直播看一下。你讲一下你的封面，我们我带你喵喵做一个证书颁发，虽然上面好

像印的有点小问题，那个金枪鱼的鱼不见了。这个金枪之夜听着像什么欧洲骑士对决，这是还有一个眼睛，然后有两个棋。就没有了，这人能给我一会儿你在听到的那个机器群吗？我一会我在群里说一下，因为你们如果要这个 PPT，我还可以把 PPT 发给你，但是这个名字就不写我们就是你可以分享里面，我们会挂主页，你把不开放的时候就好。

三教(02:22:07): 发给那边好的我来放，有没有人帮忙拍个吉祥之旅，一只眼都金枪之夜是没有。你不是本来的话，他回来晒太阳那就是你的眼睛。快点还有啥用，那这边还有别的款吗？有点亮我假装控制我不知道不，不要。可以再去。他说你说比较到时候也。

三教(02:23:33): 当然也是这样，还能说大家都放心吧，超超绝错版证书。我也不知道是不是错的，他就百分之多的讲解一个错。你有没有这样的，你看这是谁。