单数据包授权技术（SPA）是SDP框架中极为重要的一项技术，重要到什么程度呢，可以说如果没有SPA技术做支撑整个SDP架构便会土崩瓦解。这么重要的技术说是SDP的核心技术也不为过，那么关于SPA到底是什么，它究竟起到了什么作用，它的工作流程是怎样，为何SDP架构如此的依赖于它等等问题，美创安全实验室将为大家一一解答。

01

什么是SPA

这里的SPA可不是水疗，这里的SPA英文全称是Single packet authorization 中文译为单数据包授权。顾名思义，SPA这项技术就是对单个数据包进行授权的过程。虽然字面意义上的过程很简单，但其实SPA包含了很多很复杂的技术要点。

SPA作用于SDP客户端与SDP网关之间，以实现强大的服务隐蔽性。有了SPA技术，才能够实现SDP中的资产隐藏功能。所谓的资产隐藏功能，就是未经过身份验证的SDP客户端无法“看见”敏感资产，既然看不见就更谈不上访问请求了。而这个功能实现的基础就是SPA技术。SPA会根据内置算法生成一个单数据包，在经过加密分组等等一些手段发送到SDP网关之上，若是这个数据包通过了SDP网关的验证，这样才会同意与SDP客户端建立连接。除此之外SDP网关都直接将该数据包丢弃，让其他终端无法确认对方是否存在。

SPA本质上是下一代PortKnocking（PK），但解决了PK表现出的许多局限性，同时保留了其核心优势。PK限制包括防止重放攻击的一般困难，不对称密码和HMAC方案通常不可能可靠地支持，并且仅通过将额外的数据包欺骗为PK序列就很容易对PK服务器发起DoS攻击。它遍历网络，从而使得PK服务器客户端不知道正确的顺序。SPA都完美的解决了所有这些缺点，同时SPA将服务隐藏在默认丢弃防火墙策略之后，被动地（通常通过libpcap或其他方式）获取SPA数据，并为SPA数据包身份验证和加密/解密实现标准的加密操作。

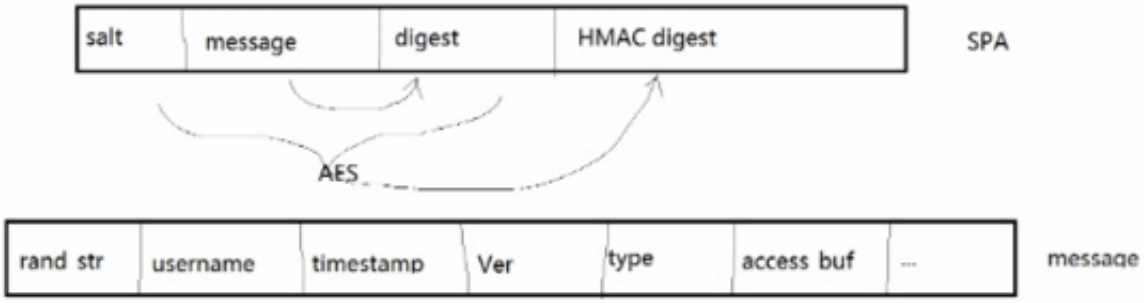
讲到这里，可能有的同学已经大致清楚了SPA究竟是个什么东西，没关系我们用最通俗的话来再为大家讲一遍：SDP架构又称“黑云”，意思是在SDP网络中，你无法“看到”除SDP控制器之外的任何其他资产（终端/端口等等），而实现这一功能的就是SPA。SPA存在于SDP客户端以及SDP网关上，但所做的工作却不相同。以黑客在内网进行端口扫描攻击举例，黑客控制了某台肉鸡，然后使用Nmap批量扫描网段，这个扫描过程其实就是在该网段下向每一个IP地址发送ICMP数据包，也就是Ping命令。在SDP网关上，SPA对接收到的数据包进行拆解，发现是ICMP数据包而不是由SDP客户端上的SPA发送的特殊单个数据包，所以将ICMP数据包丢弃，反观黑客，发现一个针对发出去的ICMP数据包的应答都没有，这也导致了黑客

无法确定到底有多少主机存活而无法进行下一步攻击步骤。这也就达到了SDP的“看不见”的效果。

02

SPA的工作原理

第一个公共可用的SPA实施项目是在2005年5月份作为一个叫fwknop的软件的一部分被发售。fwknop全称是“Fire Wall KNock Operator”，实现了单包授权的授权方案，这种授权方案基于默认丢弃数据包过滤器和libpcap。（fwknop在Linux、OpenBSD、FreeBSD和MacOS X上支持四种不同的防火墙：iptables、firewall、PE、ipfw）



最后一步SPA客户端便要将该数据包发出：

- 确定发送数据spa_data，若是Rijndael加密形式则忽略前10个字节，若是GPG加密，则忽略前2个字节（去掉salt）。
- 随机生成源端口，随机生成目的端口（可选）。
- 发送spa_data。

以上便是SPA客户端所作的详细工作说明，下面是SPA服务端的工作流程。

SPA服务端首先要监听指定的协议和端口，其具体信息可以由管理员设置。监听的作用便是能第一时间获取到针对所监听端口的数据包。

SPA服务端使用libpacp获取到数据包后，需要对该数据包进行拆解和解析：

1

获取报文的来源和目的地址。

2

匹配数据段前10（或2）字节是否和预留salt一样，相同则丢弃此数据包。

3

检测数据段是否为base64编码格式，否则丢弃。

4

从第一个stanza开始检查，若来源IP在允许访问的IP范围内，则进行后续操作。

5

还原加盐密文，头部添加salt。

6

取出数据段中的HMAC摘要，重新计算message密文的HMAC摘要，验证是否一致。用这种方式来匹配客户端对应的stanza。

7

检测message加密类型，并解密。

8

检测是否收到重放攻击报文。

9

查看SPA中是否存在timeout字段，没有则用access.conf中设置的认证超时字段。

10

查看SPA中的时间戳，与本机比对，差值不得超过设定值（default:120s）。

11

若在access.conf中的REQUIRE_USERNAME字段被设置，需确保username在SPA中可被匹配。

12

生成并配置iptables规则。

03

SPA的特征

1

在Linux上的 iptables 和firewalld防火墙，Free BSD、Open BSD和Mac OS X上的ipfw防火墙以及OpenBSD上的PF上实现单个数据包授权。

2

fwknop 客户端运行在Linux，Mac OS X，FreeBSD、Open BSD和Windows上。此处还有一个单独的Windows UI，其中包含源代码，iPhone和Android手机都有一个客户端端口，可以生成SPA数据包。

3

支持Rijndael和GnuPG方法用于SPA数据包的加密/解密。

4

支持Rijndael和GnuPG的HMAC认证加密。操作顺序是加密 - 然后验证以避免各种密码分析问题。

5

通过有效传入SPA数据包的SHA-256摘要比较来检测和阻止重播攻击。除此之外还支持SHA-1和MD5等其他摘要算法，但SHA-256是默认算法。

6

通过libpcap从导线上被动地嗅探SPA包。fwknopd服务器还可以从由单独的以太网嗅探器（例如 tcpdump-w<file>），iptables ULOG pcap writer或直接通过UDP套接字--udp-server模式写入的文件中获取数据包数据。

7

对于iptables防火墙，fwknop添加的ACCEPT规则在自定义iptables链中添加和删除（在可配置的超时之后），以便fwknop不会干扰可能已经加载到系统上的任何现有iptables策略。

8

支持经过身份验证的SPA通信的入站NAT连接（仅限 iptables 防火墙）。这意味着可以将 fwknop 配置为创建DNAT规则，以便您可以从开放Internet访问RFC 1918 IP地址上的内部系统上运行的服务（如SSH）。还支持SNAT规则，它实质上将fwknopd转换为SPA验证网关，以从内部网络访问Internet。

9

fwknop服务器支持多个用户，并且可以通过/etc/fwknop/access.conf文件为每个用户分配自己的对称或非对称加密密钥。

10

通过自动解析外部IP地址（当从NAT设备后面运行fwknop客户端时，这非常有用）。由于外部IP地址在此模式下在每个SPA数据包内进行加密，因此中间人（MITM）会攻击内联设备拦截SPA数据包并仅从其他IP转发以获取访问权限受到阻碍。

11

SPA数据包的目标端口以及通过iptables NAT功能建立后续连接的端口支持。后者适用于转发到内部服务的连接以及授予运行fwknopd的系统上的本地套接字的访问权限。

12

与Tor集成（如本演示文稿中所述）。请注意，由于Tor使用TCP进行传输，因此通过Tor网络发送SPA数据包要求每个SPA数据包都通过已建立的TCP连接发送，因此

从技术上讲，这打破了“单数据包授权”的“单一”方面。但是，Tor提供的匿名优势在某些部署中可能超过这一考虑因素。

13

实现SPA通信的版本化协议，因此很容易扩展协议以提供新的SPA消息类型，同时保持与旧的fwknop客户端的向后兼容性。支持代表有效SPA数据包执行shell命令。

14

fwknop服务器可以配置为对入站SPA数据包施加多个限制，超出加密密钥强制执行的限制和重放攻击检测。即，包年龄，源IP地址，远程用户，对请求端口的访问等。

15

捆绑了fwknop是一个全面的测试套件，它发布了一系列测试，旨在验证fwknop的客户端和服务器部分是否正常工作。这些测试涉及通过本地环回接口嗅探SPA数据包，构建临时防火墙规则，根据测试配置检查相应的访问权限，并解析来自fwknop客户端和fwknopd服务器的输出，以获得每个测试的预期标记。测试套件输出可以很容易地匿名化，以便与第三方进行通信以进行分析。

16

fwknop是第一个将端口敲击与被动操作系统指纹识别相结合的程序。但是，单包授权提供了除端口 knocking之外的许多安全优势，因此通常不推荐端口 knocking操作模式。