Hot Potato是用于windows7,8,10,Server 2008,Server2012上的权限提升，是一种新颖的网络攻击方式。

How it works

Hot Potatok利用了windows中已有的一些问题，在windows默认配置下进行本地权限提升。实际是利用了NTLM重放(更确切的说是HTPP->SMB重放)，和NBNS欺骗。

这听起来似乎觉得有些熟悉，那是因为类似的技术Google Project Zero的伙计们也披露过link。事实上，我们的代码就借用了他们的poc中的代码，并进行了扩展。

利用这种技术，从最低权限提升到“NTAUTHORITY\SYSYTEM”-windows机器上的最高级别。

能进行权限提升这一点很重要。因为许多组织都依赖于windows账户权限来保护公司网络。通常的情况就是，一旦一个攻击者能在任何基于windows的工作站或者服务器获得高的权限，他们就可以利用它进行后续操作，攻陷同域下的其它主机。对于攻击者，我们通常都是以低权限用户或者服务帐号访问一个主机的。获取对主机的高访问权限在渗透测试中都是至关重要的一步，并且通常都是以特殊的方式来完成的，因为没有已知的公开的exp或者技术能保证一定能完成这一步。

本exp并没有使用什么新的技术来进行权限提升，但组合的使用了这些技术，这一点是关键。这些技术微软大概2000年就知道了。但是不幸的是，如果不破坏向后兼容性，就很难对其进行修复，而且攻击者利用这些技术已经超过15年了。

该exp由3部分构成，每个部分都可以通过命令行中的相关参数进行配置。每一个部分对应一个已经被使用来多年的攻击。

1. Local NBNS Spoofer

NBNS 是一个UDP广播协议，通常用于windows环境中的域名解析。当你(或者windows)进行DNS查询的时候，首先windows会检查”hosts”文件。如果其中不存在相应的条目，接着windows会进行DNS查询。如果DNS查询失败，接着会进行NBNS查询。NBNS协议会询问本地广播域中的所有主机，“谁知道主机xxx的ip地址？”。该网络中的任何主机都可以进行应答，返回什么内容由应答方决定。

在渗透测试中，我们通常会嗅探网络流量，对本地网络中的NBNS查询进行应答。我们会冒充所有的主机，用我们的ip对所有的请求进行应答，期待最终的连接会做一些我们感兴趣的事情，比如认证。

对于权限提升，我们并不能保证我们能够嗅探网络流量。为什么呢？因为这需要本地管理员访问权限。那么我们如何做到NBNS欺骗呢？

如果我们能够事先知道目标主机期望NBNS查询获得的主机名(在本次中，我们的目标是127.0.0.1)，我们可以伪造一个响应，对发送NBNS查询的那个主机快速的大量发送NBNS响应(因为它是UDP协议)。其中

麻烦的一点是在NBNS包中2字节的TXID字段，在请求和响应中必须匹配，我们并不能看到该请求。解决的办法是，我们可以通过快速的发送响应，对65536种可能的值都进行尝试。

如果我们的目标网络中有我们希望假冒的主机的DNS记录怎么办呢？我们可以利用UDP端口耗尽技术，使得系统中的所有DNS查询都失败。我们需要做的就是绑定每一个UDP端口。这会导致DNS失败，因为没有可用于UDP请求的源端口。当DNS查询失败时，就会使用NBNS。

在测试中，由于向127.0.0.1发包的速率很快，成功率是100%。

2. Fake WPAD Proxy Server

在windows中，Internet Explorer默认会自动尝试通过访问”http://wpad/wpad.dat”来检测网络的代理设置。让人惊奇的是这也适用于一些windows服务，如Windows Update,但具体情况与Windows版本有关。

当然，URL”http://wpad/wpad.dat”在网络中是不存在的，因为主机名”wpad”在DNS域名服务器中是不存在的。然而，和上面一样，我们可以利用NBNS欺骗来伪造主机名。

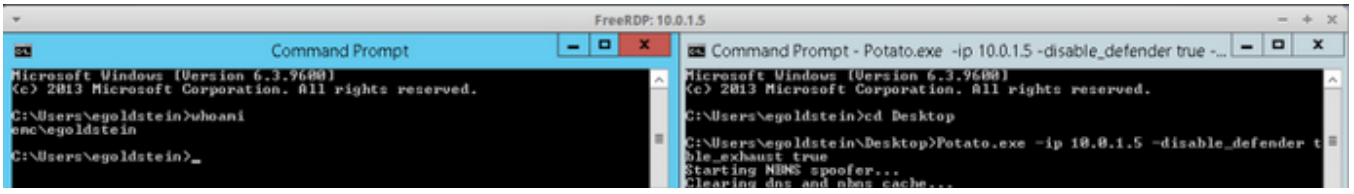
在能够进行NBNS应答欺骗后，我们将127.0.0.1作为NBNS的伪造者。我们对目标机器(我们自己的机器)用NBNS响应包进行泛洪攻击，告诉对方WPAD主机的ip地址位127.0.0.1。

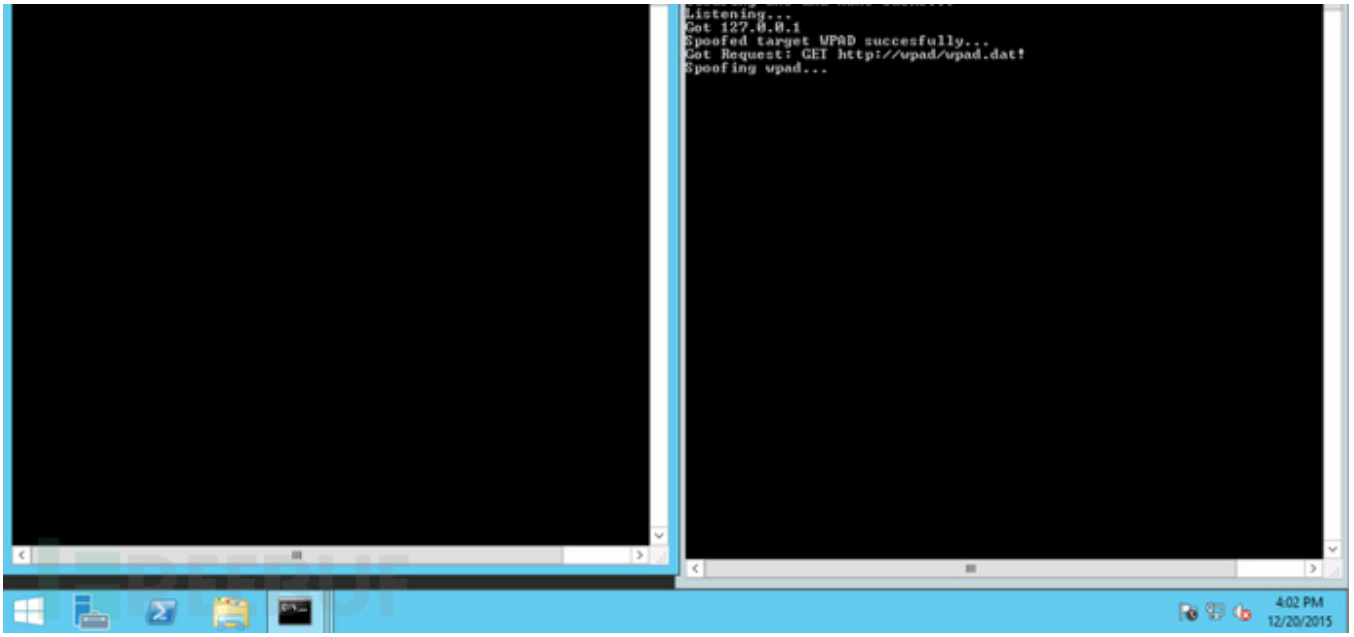
与此同时，我们在本机127.0.0.1上运行http 服务器。当它收到对”http://wpad/wpad.dat”的请求时，用类似下面的内容进行响应。

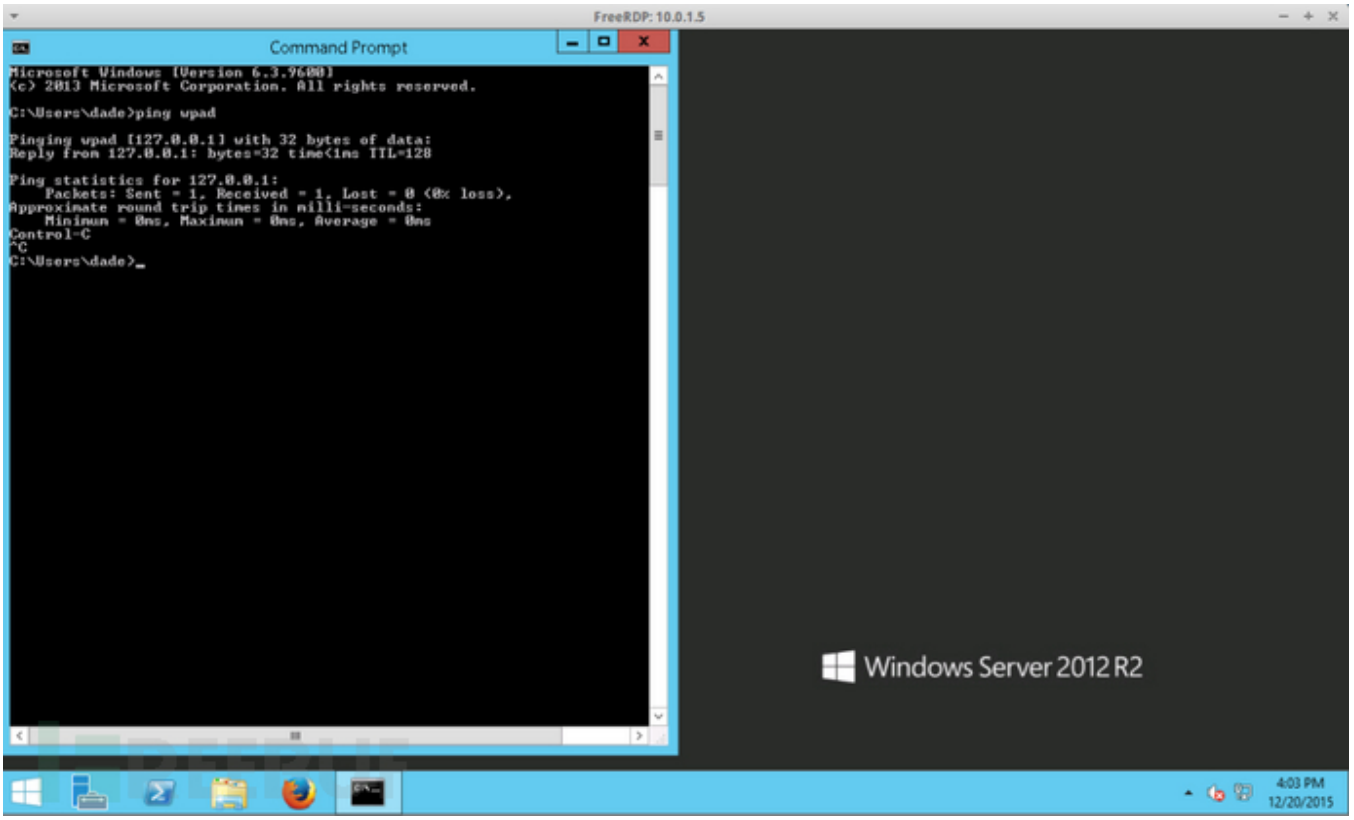
```
FindProxyForURL(url,host){
if  (dnsDomainIs(host,  "localhost"))  return  "DIRECT";
return  "PROXY  127.0.0.1:80";}
```

这会导致目标机器上的所有流量都会通过我们运行在127.0.0.1上的服务器进行重定向。

有意思的是，即使是较低权限的用户进行的这种攻击，也会影响到该机器上的所有用户。这包括管理员，以及系统账户。下面的截图显示了，两个用户同时登录到了同一台机器上，低权限用户正进行本地NBNS欺骗，高权限用户在第二个图中显示受到了影响。







3. HTTP -> SMB NTLM Relay

NTLM重放早已众所周知，但是我们通常错误理解了对Windows NTLM认证的攻击。NTLM协议容易受到中间人攻击。如果攻击者能够欺骗一个用户用NTLM向攻击者的机器进行认证，攻击者能够对其它的机器重放这个认证尝试。

这种攻击的以前的利用方式是，让受害者利用SMB协议向攻击者使用NTLM认证尝试认证。攻击者可以对受害者机器重放这些凭据，使用像” pesexec”这类技术获得远程访问。

微软对此已经进行了修复，利用挑战应答机制禁用了同协议的NTLM认证。这也就意味着SMB->SMB的NTLM重放，从一个主机再到该主机本身，是行不通的了。然而，跨协议攻击，如HTTP->SMB NTLM

重放攻击仍然有效。

假设现在所有的HTTP流量都经过我们控制的HTTP服务器，我们可以重定向到其它需要NTLM认证的地方，诸如此类。

在Potato exp中，所以HTTP请求都用302重定向到了“http://localhost/GETHASHESxxxxx”。对“http://localhost/GETHASHESxxxxx”的请求，会返回401，请求进行NTLM认证。任何NTLM凭证接着都被重放给本地的SMB listener，以创建一个新的运行用户定义的命令的系统服务。当HTTP请求来自于高权限的账户时，例如是来自windows 更新服务的请求，命令就会以”NT AUTHORITY\SYSTEM”权限运行。

Using The Exploit

exp的使用方式目前与系统版本有关。

有时候可能有点不可靠，这与windows处理代理设置和WPAD文件的奇怪方式有关。通常当exp不能奏效的时候，等它继续运行一段时间。当windows有了WPAD的缓存项，或者由于没有WPAD而允许直接访问internet时，需要30-60分钟来刷新WPAD文件。有必要让exp一直运行，这样过了那个时间段，就能触发它了。

这里列举的技术难度由低到高进行了列举。列举在后面的技术适用于以前的所有版本。每个均包含截图与视频。

Windows 7-视频link

通过利用Windows 7的Windows Defender更新机制几乎能稳定利用。

Potato.exe由代码自动触发这个。如下运行即可:

```
Potato.exe -ip -cmd [cmd to run] -disable_exhaust true
```

这会启动NBNS欺骗，伪造“WPAD”的ip为127.0.0.1,接着会检查windows Defender 更新。

如果你的网络中已经有了“WPAD”的DNS项，可以使用”-disable_exhaust false”选项。这会导致所有DNS查询失败，从而使用NBNS。windows 7上，这种利用运行相当稳定。

Winows Server 2008-视频 link

由于Windows Server并没有自带Defender,我们需要使用另一种方式。我们检查Windows更新。需要说明的是，至少在我的域上，Server 2008请求的是 WPAD.DOMAIN.TLD，而不是WPAD。下面是一个使用例子:

```
Potato.exe -ip -cmd [cmd to run] -disable_exhaust true -disable_defender true -spoof_host WPAD.EMC.LOCAL
```

运行成功后，会检查windows更新。如果没能触发它，让exp运行大约30分钟后再检查。如果还是不行，尝试真实的下载更新。

如果你的网络已经有了“WPAD”的DNS项，可以使用”-disable_exhaust false”，这也可能会造成一些其它影响。耗尽DNS端口会导致所有的DNS查询失败。Windows Update进程在请求到WPAD之前可能需要进行一些DNS查询。这种情况下，你就需要拿捏好这个时间点了(言外之意就是可能需要保证之前的那几个DNS查询成功进行)。

Windows 8/10/Server 2012 -视频link

在最新的Windows版本中，Windows Update可能不再使用”Internet Option(Internet选项)”中的代理设置，不检查WPAD。相反Windows Update的代理设置由”netsh winhttp proxy..”控制。

对于这些版本，我们利用Windows的一个较新的特性，“automatic updater of untrusted certificates(不信任证书的自动更新)”。细节参见:link,link2。

上面的TechNet文章中提到”Windows Server 2012 R2,Windows Server 2012,Windows 8.1,Windows 8有一个自动更新机制，会每天下载证书信任列表(CTLs)”。

似乎上面提到的这几个版本的Windows系统仍然使用WPAD，即使winhttp代理设置设置成了直接访问。至于为啥会这样就不得而知了！

这种情况下，以下面的方式运行Potato.

```
Potato.exe -ip -cmd [cmd to run] -disable_exhaust true -disable_defender true
```

这种情况下，你需要等待24小时，或者用其它的方式触发更新。

如果你的网络有”WPAD”的DNS项，参考Server 2008在这种情况下的处理方式。你可以尝试端口耗尽，但可能会有点难度。

TODO: SMB Signing?

尚不清楚，启用了SMB签名这个攻击是否仍能奏效。虽然Potato目前在这种情形下不能成功，但这可能是因为我们使用的CIFS库不支持SMB签名造成的。我怀疑可能支持的原因在于，所有事情都是发生在127.0.0.1上的。如果签名是基于主机的，他们可能仍然能匹配吗？

The “New Network Attack”

让我们再思考下前面的NBNS欺骗攻击。

利用与TXID穷举攻击相同的技术，从技术上讲我们可以在我们的本地网络之外进行NBNS欺骗攻击。实际上，理论上而言，只要有一个足够快的连接，我们应该就可以对任何我们能用UDP端口137进行通信的

Windows主机进行NBNS欺骗。

这实际上似乎也能工作，至少在本地网络中，我通过Internet已经成功尝试过。我们正准备发布一个修改版的“Responder.py”来完成这个攻击。下面的视频演示了如下网络情况下的攻击:

```
PFSense firewall
10.0.0.0/24 -> Corporate LAN
10.0.1.0 /24 -> Server network
```

从公司网络上，我们会攻击服务网络上的一台机器。

Demo:https://youtu.be/Mzn7ozkyG5g

Code:https://github.com/foxglovesec/Potato

***参考来源foxglovesecurity，转载请注明来自FreeBuf黑客与极客（FreeBuf.COM）**