

# Lab 03: Third-party service recon

## Table of Contents

Lab 03: Third-party service recon .....	1
Goals .....	1
Requirements.....	1
1. Collect email addresses from the target's WHOIS records.....	1
2. Identify a valid email address from the collected contacts .....	3
3. Test the target for Office 365 use .....	5
4. Additional tests to confirm Office 365 usage.....	7

## Goals

- Identify third-party services that can be targeted or leveraged in attacks.

## Requirements

- Kali Linux VM with Internet access.
- Recon-NG database created in the previous exercise.

## 1. Collect email addresses from the target's WHOIS records

1. Start Recon-NG in a new terminal window if it's not still running from the last exercise.

```
recon-ng
```

2. Within Recon-NG, execute the commands below to retrieve email addresses from WHOIS records of each domain you previously discovered.

```
modules load recon/domains-contacts/whois_pocs  
run
```

```
[recon-ng][default] > modules load recon/domains-contacts/whois_pocs  
[recon-ng][default][whois_pocs] > run
```

*Execution of the Commands Above*

- Next, use the "show contacts" command to view all the contacts that were discovered. (Press CTRL+minus to zoom out in your terminal window if needed. CTRL+0 (zero) will return to the original zoom.)

```
show contacts
```

```
[recon-ng][default][whois_pocs] > show contacts
```

rowid	first_name	middle_name	last_name	email
1	CHRIS		AADLAND	v-chrisa@microsoft.com
2	CHRISTINA		AADLAND	v-chrisa@microsoft.com
3	Christina		Aadland	v-chrisa@microsoft.com
4			Abuse	abuse@microsoft.com
5			Administrator	ips.global.admin@ipayout.onmicrosoft.com
6	Melissa		Allison	mallison@ocmcdonald.onmicrosoft.com
7	Jeffrey		Amels	jame@s@microsoft.com
8	BRAD		AUSTIN	brada@microsoft.com
9	JO		BAKER	jolynb@microsoft.com
10	Ram		Balakrishnan	rambala@microsoft.com
11	david		Balko	dbalko@sfscapital.onmicrosoft.com
12	ADAM		BECKER	adam.becker@primew.onmicrosoft.com
13	Dawn		Bedard	dabedard@microsoft.com
14	Mukeshkumar		Beher	mukeshb@microsoft.com
15	Blake		Bisset	blake.bisset@microsoft.com
16	William		Blackwood	blackwood@bigbearaggietech.onmicrosoft.com

*Contact Information Collected from WHOIS Records*

- You may notice that some of the contacts that were discovered are for domains that do not match the domain names you targeted. For example, running the module on the target domain "xbox.com" returned results for "oxbox.com" and "knoxbox.com".

To view a list of all email addresses that **do not** match one of the domains in your domains table, run the following command in Recon-NG:

```
db query select email from contacts where email not in ( SELECT t1.email from
(SELECT * FROM contacts) as t1, (SELECT * FROM domains) as t2 where t1.email like
('%@' || t2.domain) )
```

```
[recon-ng][default][whois_pocs] > db query select email from contacts where email not in (
SELECT t1.email from (SELECT * FROM contacts) as t1, (SELECT * FROM domains) as t2 where
t1.email like ('%@' || t2.domain) )
```

email
ips.global.admin@ipayout.onmicrosoft.com
mallison@ocmcdonald.onmicrosoft.com
dbalko@sfscapital.onmicrosoft.com
adam.becker@primew.onmicrosoft.com
blackwood@bigbearaggietech.onmicrosoft.com
mbradvica@bradvica.onmicrosoft.com
abb215@abbmfg215.onmicrosoft.com
james@titancomm.onmicrosoft.com
hadoss@staradio.onmicrosoft.com

*Email Addresses not Matching a Discovered Domain are Displayed*

5. Next, remove the false-positive contacts listed in the previous step by running the following command:

```
db query DELETE FROM contacts WHERE email not IN ( SELECT t1.email from (SELECT * FROM contacts) as t1, (SELECT * FROM domains) as t2 where t1.email like ('%@' || t2.domain) )
```

```
[recon-ng][default][whois_pocs] > db query DELETE FROM contacts WHERE email not IN ( SELECT t1.email from (SELECT * FROM contacts) as t1, (SELECT * FROM domains) as t2 where t1.email like ('%@' || t2.domain) )
[*] 84 rows affected.
[recon-ng][default][whois_pocs] > █
```

*Previously Listed Email Addresses Removed*

6. Now when you run "show contacts" again, you should see only contacts whose email addresses include one of the domains stored in your domains table.

```
show contacts
```

```
[recon-ng][default][whois_pocs] > show contacts
```

rowid	first_name	middle_name	last_name	email
1	CHRIS		AADLAND	v-chrisa@microsoft.com
2	CHRISTINA		AADLAND	v-chrisa@microsoft.com
3	Christina		Aadland	v-chrisa@microsoft.com
4			Abuse	abuse@microsoft.com
7	Jeffrey		Amels	jamels@microsoft.com
8	BRAD		AUSTIN	brada@microsoft.com
9	JO		BAKER	jolynb@microsoft.com
10	Ram		Balakrishnan	rambala@microsoft.com
13	Dawn		Bedard	dabedard@microsoft.com
14	Mukeshkumar		Beher	mukeshb@microsoft.com
15	Blake		Bisset	blake.bisset@microsoft.com
17	Justin		Bouska	jbouska@microsoft.com
18	Steve		Bowman	stevebow@microsoft.com
21	Lee		Butler	leebru@microsoft.com
23	Hal		Carmichael	halcar@microsoft.com
24	Todd		Carter	toddca@microsoft.com
25	Tim		Chinn	v-timchi@microsoft.com

*Remaining Contacts Shown*

## 2. Identify a valid email address from the collected contacts

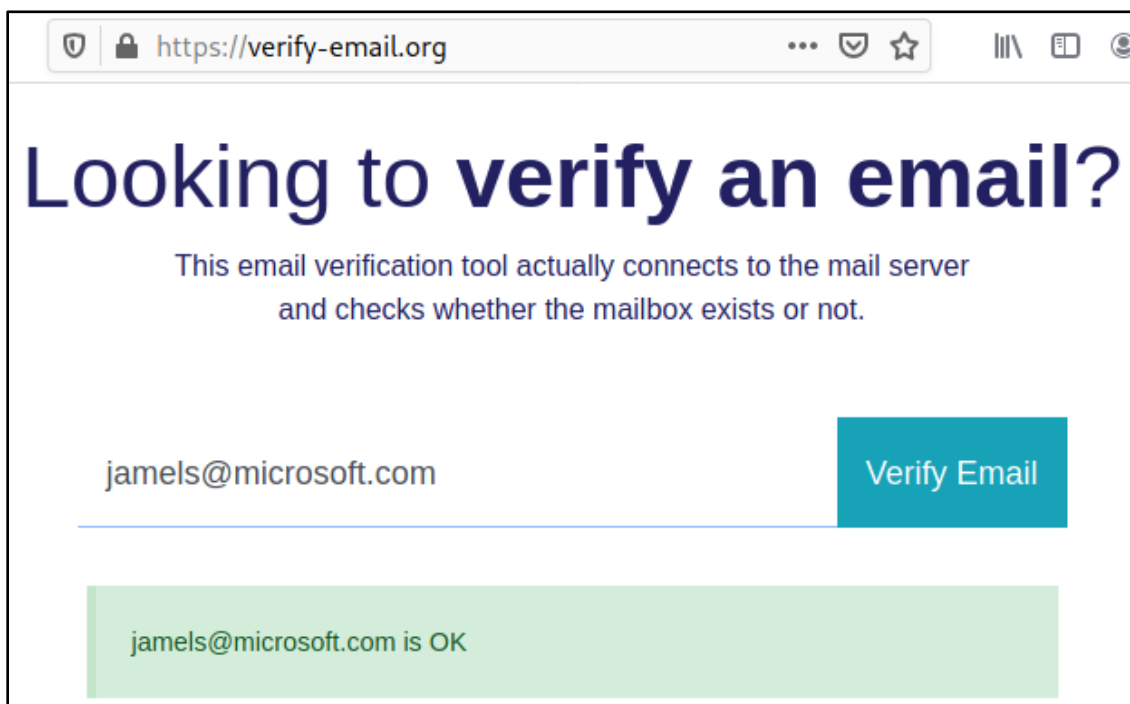
1. In your web browser, visit the email address verification service, [Verify-Email.org](https://verify-email.org). Use the search box on the page to test each email address in your contacts list until you find an email address that is valid. If your contacts list contains a large number of duplicate email addresses, you can use the command below to display the list with duplicated entries removed:

```
db query select distinct email from contacts
```

```
[recon-ng][default][whois_pocs] > db query select distinct email from contacts

+-----+
|          email          |
+-----+
| v-chrisa@microsoft.com |
| abuse@microsoft.com    |
| jamels@microsoft.com   |
| brada@microsoft.com    |
| jolynb@microsoft.com   |
| rambala@microsoft.com  |
| dabedard@microsoft.com |
| mukeshb@microsoft.com  |
| blake.bisset@microsoft.com |
| jbouska@microsoft.com  |
+-----+
```

*Unique Email Addresses Displayed*

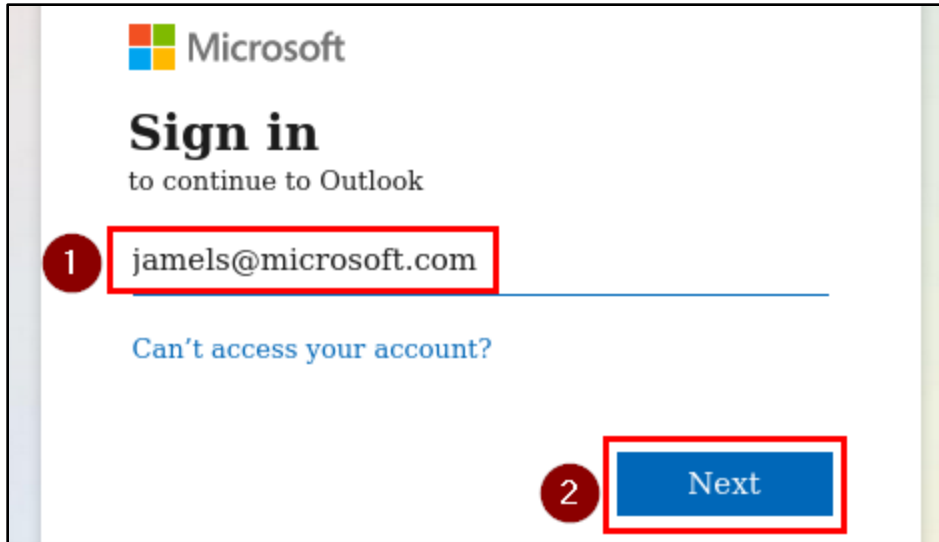


*Email Address Verified with Verify-Email.org*

### 3. Test the target for Office 365 use

1. In your web browser, visit <https://outlook.office.com>. Then enter the valid email address you identified above and click "Next".

<https://outlook.office.com>



Microsoft

## Sign in

to continue to Outlook

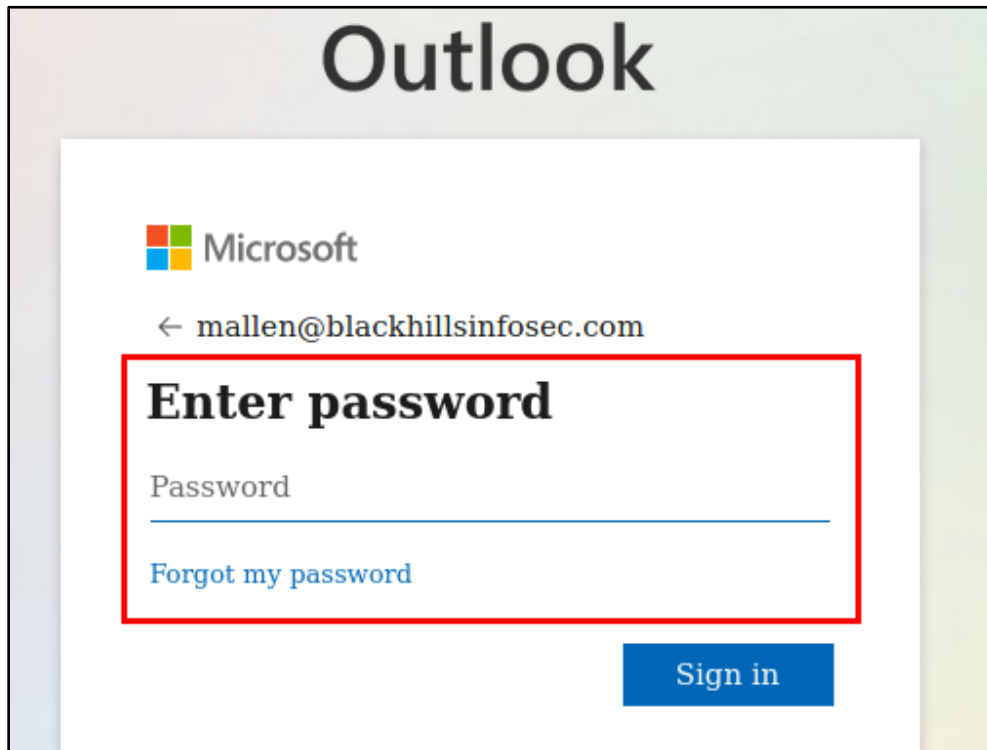
1

[Can't access your account?](#)

2

*Email Address Submitted to the Office 365 Login Form*

2. If the email address is associated with a valid Microsoft or Office 365 account, you will receive some type of authentication request - either for a password or for a multi-factor token - or you may be redirected to the organization's sign-in page. Example responses for valid Microsoft accounts are shown below.



# Outlook

Microsoft

← mallen@blackhillsinfosec.com

## Enter password

Password

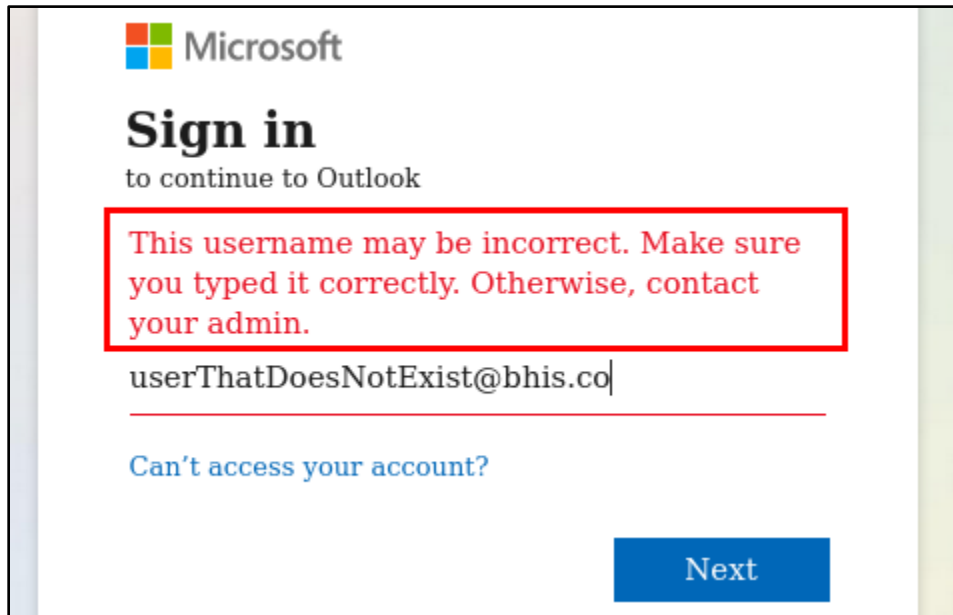
[Forgot my password](#)

*Example of a Generic Office 365 Authentication Request*



*Example of a Branded Office 365 Authentication Request*

3. If the email address is not associated with a valid Microsoft account, you will see an error message instead.



*Example Response for a Non-Existent User Account*

4. If possible, it is best to use this test on at least two different email accounts on the same domain. That will help rule out the possibility that users have setup personal Microsoft accounts with their work email addresses.

## 4. Additional tests to confirm Office 365 usage

1. The Linux "dig" command can also be used to confirm the use of Office 365. In a new terminal window, run the command below, replacing "contoso.com" with the email domain used by your target organization. A response in the answer section that includes ".mail.protection.outlook.com" is an additional indicator that the organization uses Office 365 for email.

```
dig -t mx contoso.com
```

```
(kali㉿kali)-[~]
└─$ dig -t mx contoso.com

; <<>> DiG 9.16.8-Debian <<>> -t mx contoso.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3047
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:;; udp: 512
;; QUESTION SECTION:
;contoso.com.                IN      MX
;; ANSWER SECTION:
contoso.com.                3582    IN      MX      10 contoso-com.mail.protection.outlook.com
.
```

*Indication of Office 365 Use Observed*

2. Note that the lack of ".mail.protection.outlook.com" in this DNS record does not necessarily indicate that Office 365 is *not* used. Several vendors sell email filtering services that act as the incoming email server but leverage Office 365 on the back end.

3. Similar to the process described in the last section, this can also be used to confirm the use of other email providers such as Google, as shown below.

```
(kali@kali) - [~]
$ dig -t mx mikeallen.org

; <<>> DiG 9.16.8-Debian <<>> -t mx mikeallen.org
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 35264
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;mikeallen.org.                IN      MX

;; ANSWER SECTION:
mikeallen.org.                1799    IN      MX      30      aspmx3.googlemail.com.
mikeallen.org.                1799    IN      MX      30      aspmx4.googlemail.com.
mikeallen.org.                1799    IN      MX      10      aspmx.l.google.com.
mikeallen.org.                1799    IN      MX      20      alt2.aspmx.l.google.com.
mikeallen.org.                1799    IN      MX      30      aspmx5.googlemail.com.
mikeallen.org.                1799    IN      MX      30      aspmx2.googlemail.com.
mikeallen.org.                1799    IN      MX      20      alt1.aspmx.l.google.com.
```

*Evidence of Google Email Services in use by the Target Domain*

4. TXT records can also provide indicators of which email service is used. As shown in the screenshots below, SPF records that include a subdomain of outlook.com and TXT records containing an "MS=ms#####" value are additional indicators that Office 365 is in use. You can use the following dig command to retrieve the TXT records for your target domain:

```
dig -t txt contoso.com
```



```

(kali㉿kali)-[~]
└─$ dig -t txt bhis.co

; <<>> DiG 9.16.8-Debian <<>> -t txt bhis.co
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56552
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;bhis.co.                IN      TXT

;; ANSWER SECTION:
bhis.co.                 3499   IN      TXT     "google-site-verification=OG_kczftsBwnJPuN
fd0sC3MCg0oXvv0W2lJ3vaLN9iM"
bhis.co.                 3499   IN      TXT     "v=spf1 include:spf.protection.outlook.com
~all"
bhis.co.                 3499   IN      TXT     "MS=ms10535992"

```

*Indicators of Office 365 use Present in TXT Records*

- Again, this same test can also indicate the possible use of Google or other services as well.

```

(kali㉿kali)-[~]
└─$ dig -t txt mikeallen.org

; <<>> DiG 9.16.8-Debian <<>> -t txt mikeallen.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43102
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;mikeallen.org.         IN      TXT

;; AUTHORITY SECTION:
mikeallen.org.         299    IN      TXT     "v=spf1 mx include:_spf.google.com ~all"

```

*Indicators of Google Services Found in TXT Records*