Privileged Account Insecurity: The Target of All Exploitive Attacks

Presented by: Vishal Patel,CISSP

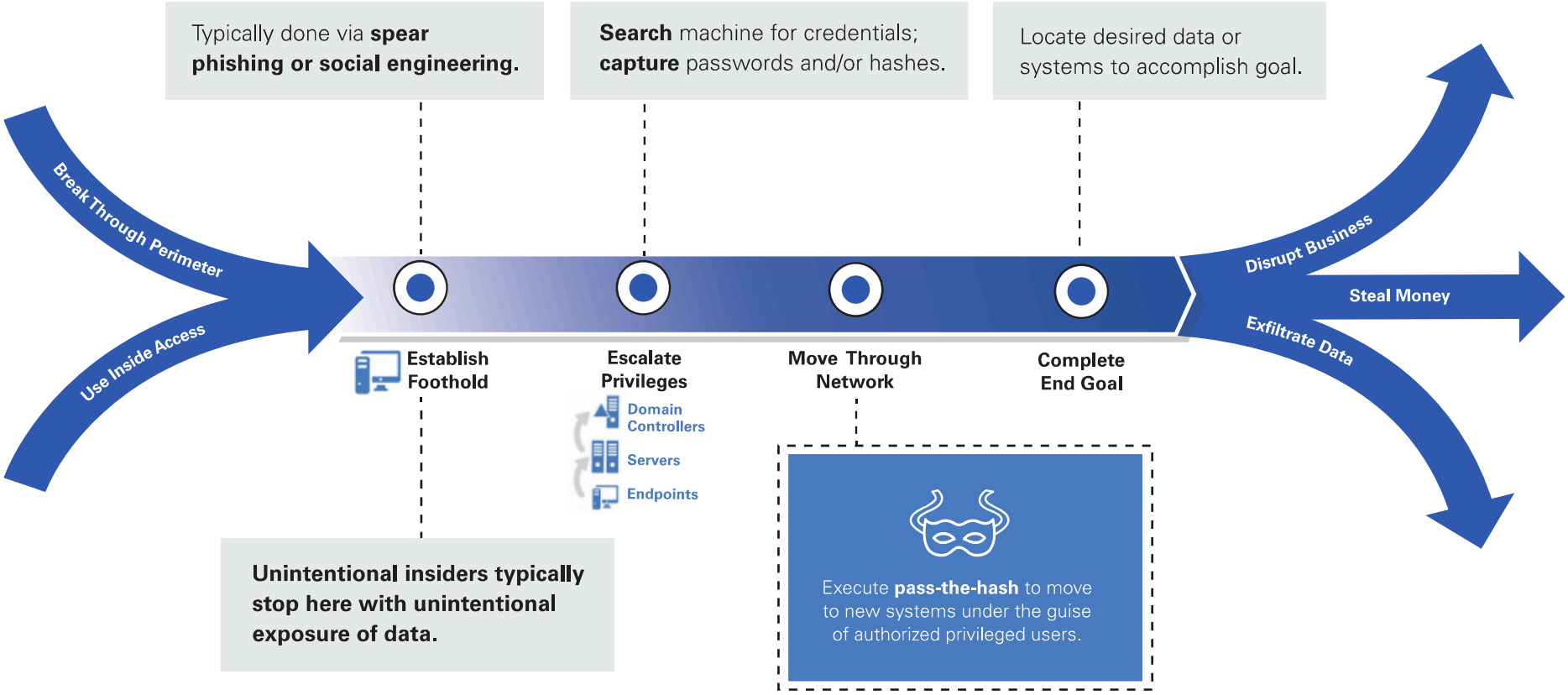
An Attacker Must Obtain Insider Credentials

"…100% of breaches involved stolen credentials."

"APT intruders…prefer to leverage privileged accounts where possible, such as Domain Administrators, service accounts with Domain privileges, local Administrator accounts, and privileged user accounts."

Mandiant, M-Trends and APT1 Report

The privileged pathway of insider attacks



Typical processes that attackers expose…

Local admin accounts set to the same password

Unmanaged SSH Keys used for interactive sessions and applications

Separate, named domain accounts created for each admin

Workstation users granted local admin rights

Non-expiring passwords for critical accounts

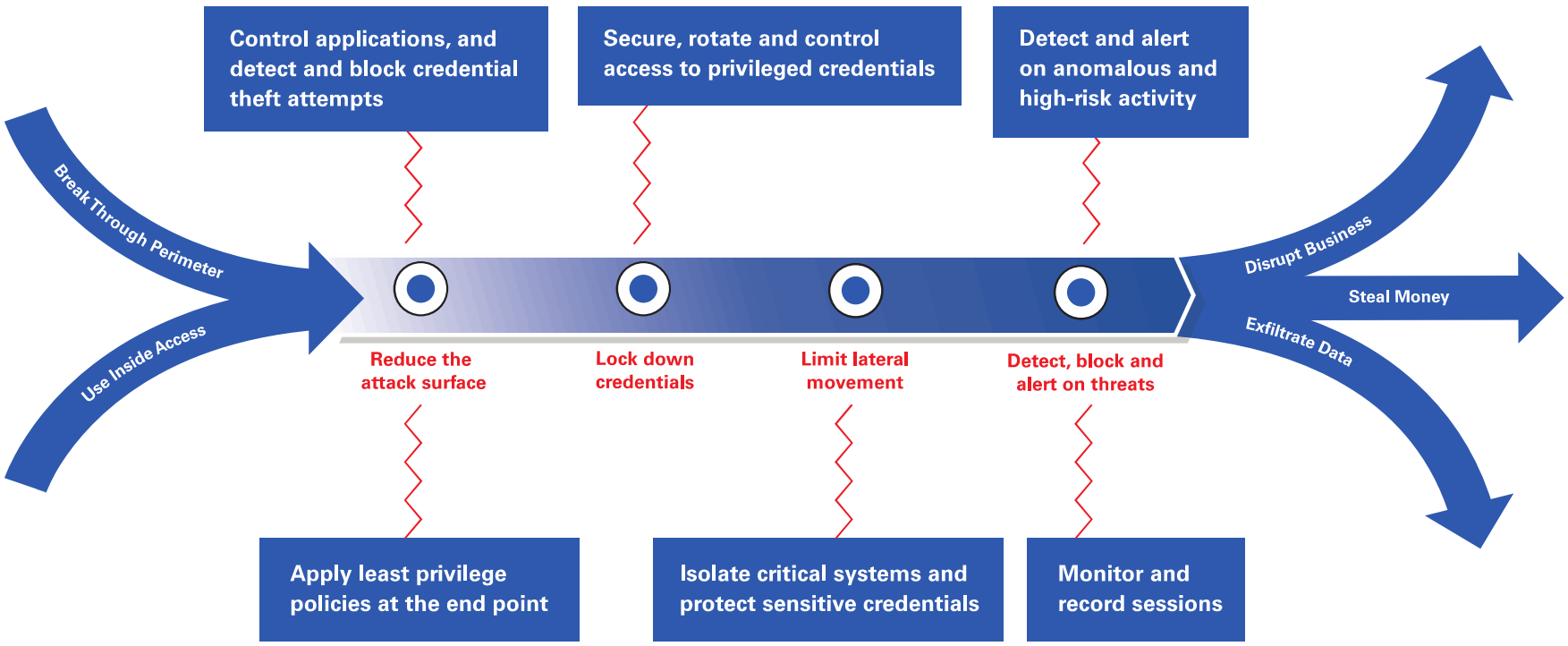
Standing Access – network, access and authentication

Hard-coded credentials for applications in code, scripts and appliances

Excessive Permissions for specific roles, like; DBA, Developers, etc.

Lack of visibility around who, why and is it legitimate access

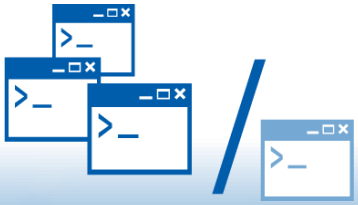
Make the attacker's job as difficult as possible



Next steps

Lock Down Credentials

Secure, rotate and control access to privileged passwords and SSH keys



Isolate & Control Sessions

Prevent direct access to critical systems and enforce least privileges

Continuously Monitor

Continuously monitor privileged user and account activity

AD User Audit – Low Hanging Fruit

Administrator: Windows PowerShell ISE

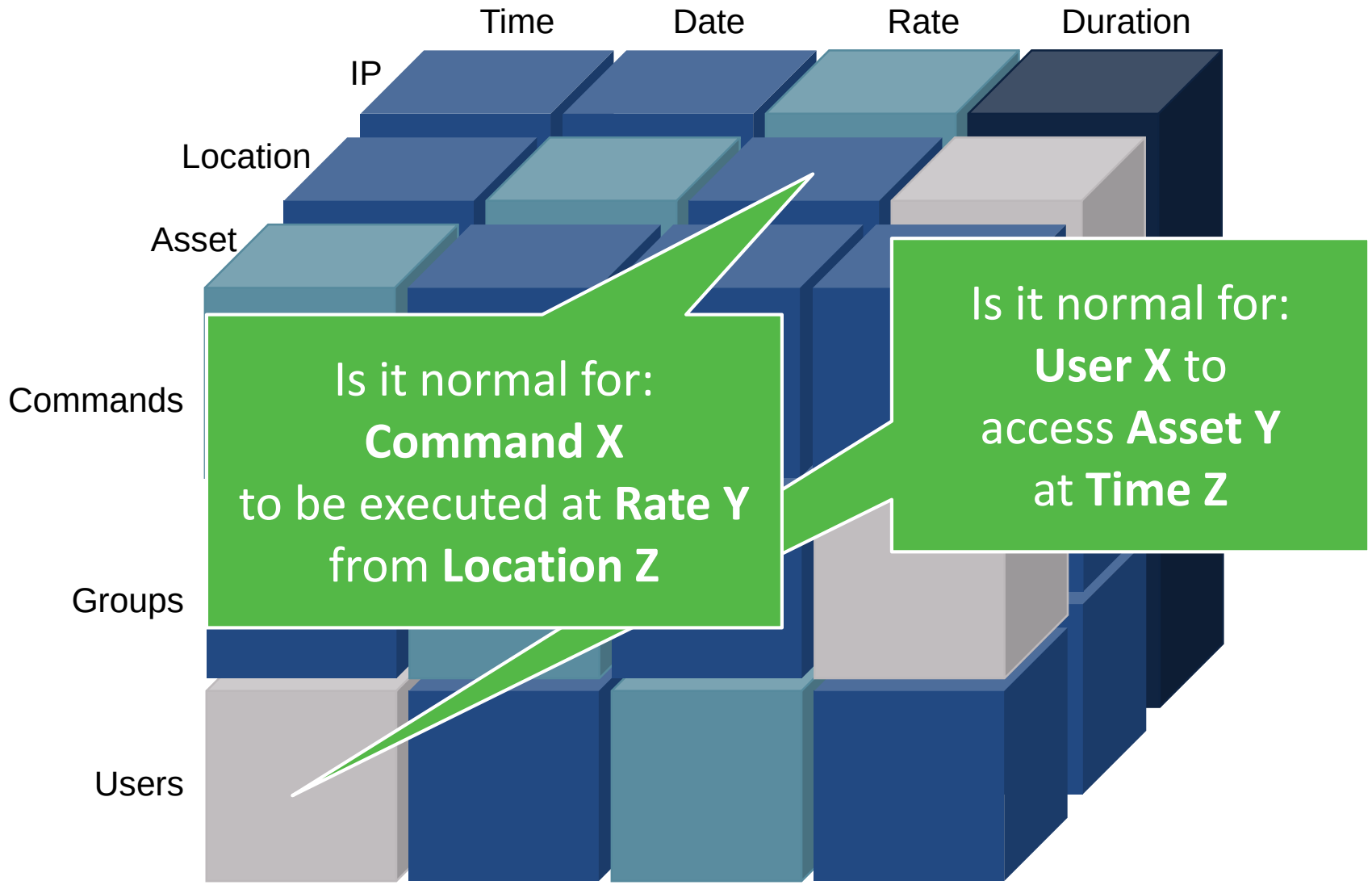
File Edit View Tools Debug Add-ons Help

UserAccountAudit.ps1

```
$Domain = [System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()
$ADSearch = New-Object System.DirectoryServices.DirectorySearcher
$ADSearch.SearchRoot = "LDAP://$Domain"
$ADSearch.SearchScope = "subtree"
$ADSearch.PageSize = 100

$ADSearch.Filter = "(objectClass=user)"

$ADSearch.PropertiesToLoad.Add("distinguishedName")
$ADSearch.PropertiesToLoad.Add("sAMAccountName")
$ADSearch.PropertiesToLoad.Add("lastLogonTimeStamp")
$ADSearch.PropertiesToLoad.Add("pwdLastSet")
$ADSearch.PropertiesToLoad.Add("userAccountControl")

$userObjects = $ADSearch.FindAll()
foreach ($user in $userObjects)
{
    $dn = $user.Properties.Item("distinguishedName")
    $sam = $user.Properties.Item("sAMAccountName")
    $logon = $user.Properties.Item("lastLogonTimeStamp")
    if($logon.Count -eq 0)
    {
        $lastlogon = "Never"
    }
    else
    {
        $lastlogon = [DateTime]$logon[0]
        $lastlogon = $lastlogon.AddYears(1600)
    }
```

```
"CN=Charlie Pace,OU=iSeries Admins,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",Charlie,Never,04/27/2015 16:19:03,Enabled-password never expires
"CN=Clarie Littleton,OU=iSeries Admins,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",Clarie,Never,04/27/2015 16:20:16,Enabled-password never expires
"CN=Cindy Schumaker,OU=IT Auditors,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",Cindy,06/29/2016 19:07:01,04/27/2015 16:24:15,Enabled-password never exp
ires
"CN=Hugo Reyes,OU=IT Auditors,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",Hugo,Never,04/27/2015 16:25:51,Enabled-password never expires
"CN=Judy Fleming,OU=IT Auditors,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",Judy,Never,04/27/2015 16:26:48,Enabled-password never expires
"CN=Carlos Rodriguez,OU=Network Admins,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",Carlos,09/09/2016 22:06:19,04/27/2015 16:52:02,Enabled-password neve
r expires
"CN=Chris Bracket,OU=Security Admins,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",Chris,Never,04/27/2015 16:54:43,Enabled-password never expires
"CN=James Ford,OU=Security Admins,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",James,Never,04/27/2015 17:01:06,Enabled-password never expires
"CN=Jin Kwon,OU=Security Admins,OU=Users,OU=CyberArk,DC=CyberArk,DC=Local",Jin,Never,04/27/2015 17:12:16,Enabled-password never expires
```

Framework for Privileged Access Behavior – Alerting and Remediation



Thank You!

- Q&A