Six Degrees of Domain Admin
About Us

I am Andy Robbins

**Job:** Pentester at Veris Group's ATD
**Speaker:** BSidesLV/Seattle, ISC2 World Congress, ISSA International
**Trainer:** Black Hat USA 2016
**Other:** Ask me about ACH

**Twitter:** @_wald0
I am Rohan Vazarkar

Job: Pentester at Veris Group’s ATD
**Tool creator/dev:** EyeWitness, Python Empyre, etc.
**Presenter:** BSidesDC/LV/DE, Black Hat Arsenal
**Trainer:** Black Hat USA 2016

**Twitter:** @CptJesus
I am Will Schroeder

**Job:** Researcher at Veris Group’s ATD

**Tool creator/dev:** Veil-Framework, PowerView, PowerUp, Empire/Empyre

**Speaker:** Ask me

**Trainer:** Black Hat USA 2014-2016

**Other:** Microsoft PowerShell/CDM MVP

**Twitter:** @harmj0y
The Current State of Active Directory Domain Privilege Escalation
“Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win.”

John Lambert
GM, Microsoft Threat Intelligence Center
AD Domain Priv Esc

- Active Directory is ubiquitous
- Ubiquity = Attention = Research time and $$$
- Sometimes we get easy buttons!
Derivative Local Admin

“The chaining or linking of administrator rights through compromising other privileged accounts”

Justin Warner @sixdub
Challenges

- Extremely time consuming and tedious
- Not comprehensive
- Limited situational awareness
- Did you even need DA?
Graph Theory
And attack graph design
Basic Elements of a Graph

**Vertices** represent individual elements of a system

**Edges** generically represent relationships between vertices

**Paths** are sets of vertices and edges that connect non-adjacent vertices
Vertex 1 → Edge → Vertex 2
BloodHound Attack
Graph Design

**Vertices** represent users, groups, computers, and domains

**Edges** identify group memberships, admin rights, user sessions, and domain trusts

**Paths** always lead toward escalating rights. Always.
Put Simply…

- Who is logged on where?
- Who has admin rights where?
- What users and groups belong to what groups?
Stealthy Data Collection with PowerView
“The best tool these days for understanding Windows networks is PowerView…”

Phineas Phisher
http://pastebin.com/raw/0SNSvyjJ
A pure PowerShell v2.0+ domain/network situational awareness tool

Collects the data that BloodHound is built on and doesn’t need elevated privileges for most collection methods!
Who’s Logged in Where?
aka “user hunting”

- **Invoke-UserHunter:**
  - Get-NetSession – sessions w/ a remote machine

- **Stealth:**
  - Enumerate commonly trafficked servers and query remote sessions for each
We can enumerate members of a local group on a remote machine, without admin privileges!

- The WinNT service provider or NetLocalGroupMembers()

PowerView:
- Get-NetLocalGroup -ComputerName IP [-API]
GPOs can set local administrators
GPOs are applied to OUs/Sites
- correlation == local admin information through communication with only a DC!

PowerView:
- Find-GPOLocation
Who’s in What Groups?

- Enumerate all groups and pull the members of each

- PowerView:
  - `Get-NetGroup` | `Get-NetGroupMember`

- That’s it!
Bringing it All Together
The BloodHound Ingestor

Get-BloodHoundData automates gathering PowerView data for a domain.

Export-BloodHoundData exports collected data to a neo4j batch REST API for ingestion.

Export-BloodHoundCSV exports collected data to a series of CSVs for offline ingestion.
BloodHound
Live demo!
BloodHound

- Built with Linkurious.js
- Compiled with Electron
- Uses a neo4j graph database
- Fed by the custom PowerShell ingestor
bit.ly/GetBloodHound
Thanks!

@_wald0
@CptJesus
@harmj0y