The Active Directory Botnet

Ty Miller

Managing Director

Threat Intelligence Pty Ltd

ty.miller@threatintelligence.com

www.threatintelligence.com

Paul Kalinin

Senior Security Consultant

Threat Intelligence Pty Ltd

paul.kalinin@threatintelligence.com

www.threatintelligence.com

Introduction

Who are we?

- **Ty Miller**
  - Managing Director, Threat Intelligence Pty Ltd (www.threatintelligence.com)
  - Specialist Security Company based in Australia
  - CREST Australia New Zealand (Board of Directors, Technical Team Lead, Assessor)

- **Security Researcher, Penetration Tester, Presenter and Trainer**
  - Black Hat Training — The Shellcode Lab
  - Black Hat Training — Practical Threat Intelligence
  - Black Hat Presentation — Reverse DNS Tunneling Shellcode
  - Black Hat Webcast — The Best Way to Catch a Thief
  - Black Hat: — Black Hat Asia Review Board
  - Core Impact: — DNS Channel Payload
  - Development and Presentation — BeEF Bind Shellcode
  - Co-Author — Hacking Exposed Linux 3rd Edition
  - Presentation — Machine Learning and Modern Malware Mitigations
  - Presentation — Modern Threat Detection and Prevention
  - Presentation — Securing Your Startup to Secure Big Brands
  - Presentation — Can your application be breached?

Who are we?

- Paul Kalinin
  - Senior Security Consultant, Threat Intelligence Pty Ltd (www.threatintelligence.com)
  - Certs: CREST, CEH, CISSP, PCI QSA

- Penetration Tester, Security Specialist and Security Researcher
  - Black Hat Training — Practical Threat Intelligence
  - Specialty — Internal Infrastructure and Wireless Penetration Testing
  - Specialty — Red Team Penetration Testing
  - Specialty — Web and Mobile Penetration Testing
  - Specialty — Cyber Threat Intelligence Analyst
  - Specialty — Attack Design, Development and Weaponization
  - Specialty — Security Architecture and Governance
  - Black Hat Presentation — The Active Directory Botnet

What are we talking about?

- We are going to demonstrate how to exploit a fundamental flaw in the way that nearly every organization implements their Active Directory solution
- This attack technique introduces a gaping hole within your internal and hybrid-cloud security architectures that impacts your ability to contain security breaches
- This is achieved by turning your Active Directory solution into an internal Botnet Command & Control cluster
- Demonstrate the capability to bypass your internal firewalls and network segmentation to communicate with all internal hosts
- Remotely controlling the AD Botnet

Current State of Play

Threat Actors are Winning

	Incidents				Breaches			
	Total	Small	Large	Unk	Total	Small	Large	Unk
Total	**42,068**	**606**	**22,273**	**19,189**	**1,935**	**433**	**278**	**1,224**
Accommodation (72)	215	131	17	67	201	128	12	61
Administrative (56)	42	6	5	31	27	3	3	21
Agriculture (11)	11	1	1	9	1	0	1	0
Construction (23)	6	3	1	2	2	1	0	1
Education (61)	455	37	41	377	73	15	15	43
Entertainment (71)	5,534	7	3	5,524	11	5	3	3
Finance (52)	998	58	97	843	471	39	30	402
Healthcare (62)	458	92	108	258	296	57	68	171
Information (51)	717	57	44	616	113	42	21	50
Management (55)	8	2	3	3	3	2	1	0
Manufacturing (31-33)	620	6	24	590	124	3	11	110
Mining (21)	6	1	1	4	3	0	1	2
Other Services (81)	69	22	5	42	50	14	5	31
Professional (54)	3,016	51	21	2,944	109	37	8	64
Public (92)	21,239	46	20,751	442	239	30	59	150
Real Estate (53)	13	2	0	11	11	2	0	9
Retail (44-45)	326	70	36	220	93	46	14	33
Trade (42)	20	4	10	6	10	3	6	1
Transportation (48-49)	63	5	11	47	14	3	4	7
Utilities (22)	32	2	5	25	16	1	1	14
Unknown	8,220	3	1,089	7,128	68	2	15	51
Total	**42,068**	**606**	**22,273**	**19,189**	**1,935**	**433**	**278**	**1,224**

Table 1: Number of security incidents by victim industry and organization size, 2016 dataset.

** Verizon 2017 Data Breach Investigations Report - 10th Edition*

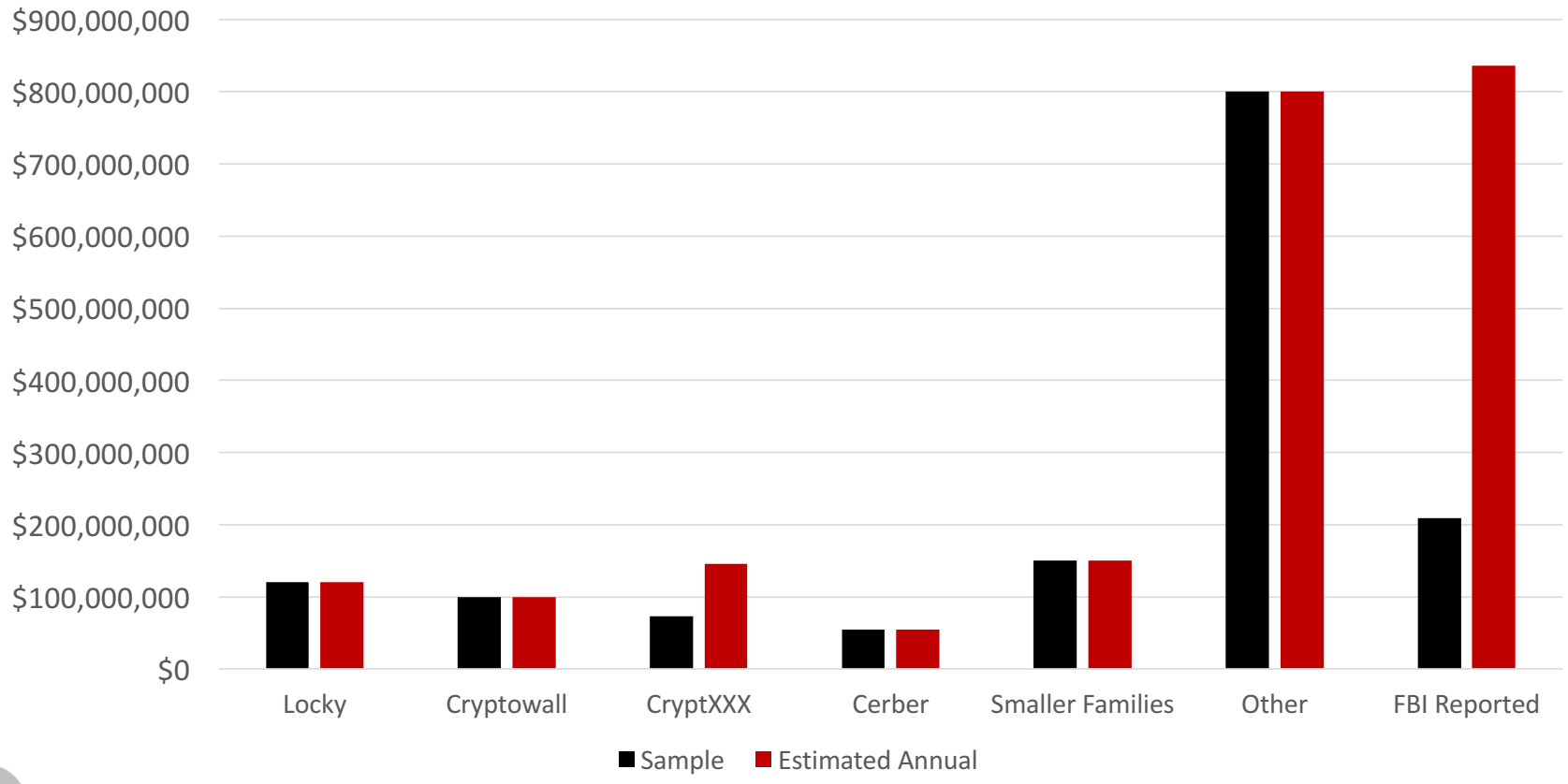
Financially Motivated Attackers

		Variety		
		ESP	FIG	FIN
Use of stolen creds	(hacking)	27	6	598
Use of backdoor/C2	(hacking)	121		557
Theft	(physical)			39
Tampering	(physical)			27
Surveillance	(physical)			21
SQLi	(hacking)			14
Spyware/Keylogger	(malware)	38		557
Skimmer	(physical)			60
Ransomware	(malware)			14
Ram scraper	(malware)			191
Privilege abuse	(misuse)	17	37	74
Pretexting	(social)			39
Possession abuse	(misuse)	6	9	29
Phishing	(social)	163		490

		Vector		
		ESP	FIG	FIN
Website	(social)	19		
Web drive-by	(malware)	26		
Web application	(hacking)	5	23	507
Victim work area	(physical)			16
Victim public area	(physical)			39
Victim grounds	(physical)			31
Remote access	(misuse)		7	7
Public facility	(physical)			6
Physical access	(misuse)	8	11	34
Phone	(social)			5
Personal vehicle	(physical)			7
Partner facility	(physical)			5
Partner	(hacking)			108
LAN access	(misuse)	19	31	68

** Verizon 2017 Data Breach Investigations Report - 10th Edition*

Ransomware Revenue



** csoonline.com article "Ransomware took in $1 billion in 2016"*

Primary Attack Techniques

- Command & Control (C2) systems are key to modern security breaches
- *"phishing remaining a favorite technique of attackers … payloads are commonly delivered via email (73%) and drive-by downloads (13%)"*
- *"If the attachment is opened, it will drop command and control malware to establish and maintain control of the device"*

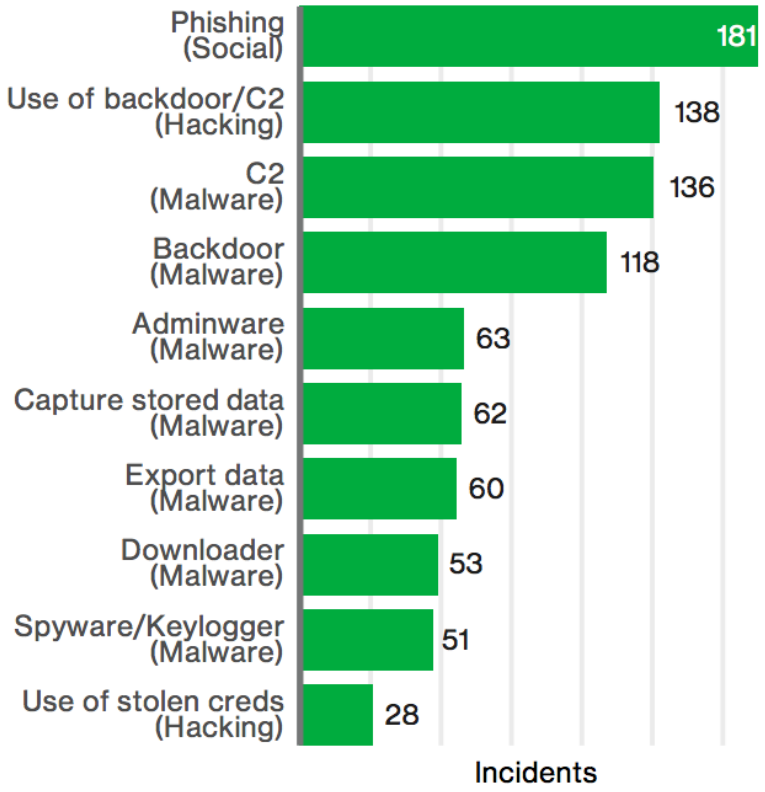


Figure 39: Top threat action varieties within Cyber-Espionage, (n=271)

** Verizon 2017 Data Breach Investigations Report - 10th Edition*

Attack Capabilities

- Currently attackers are still trying to be stealthy … but for how long?
- Open Source technologies have enabled attacks to have the capability to become highly sophisticated
- Cloud platforms are extending internal networks out to the internet, and often introducing significant security weaknesses and removing visibility of threats
- Huge potential for attacks to turn noisy for Fast Escalation and Large Impact attacks
- Harder to recover from … Lead to increased revenue stream for attackers

Current State of Play

Highly motivated Threat Actors
utilizing endpoint exploitation techniques
that connect to Command & Control (C2) servers
to launch fast and effective internal attacks

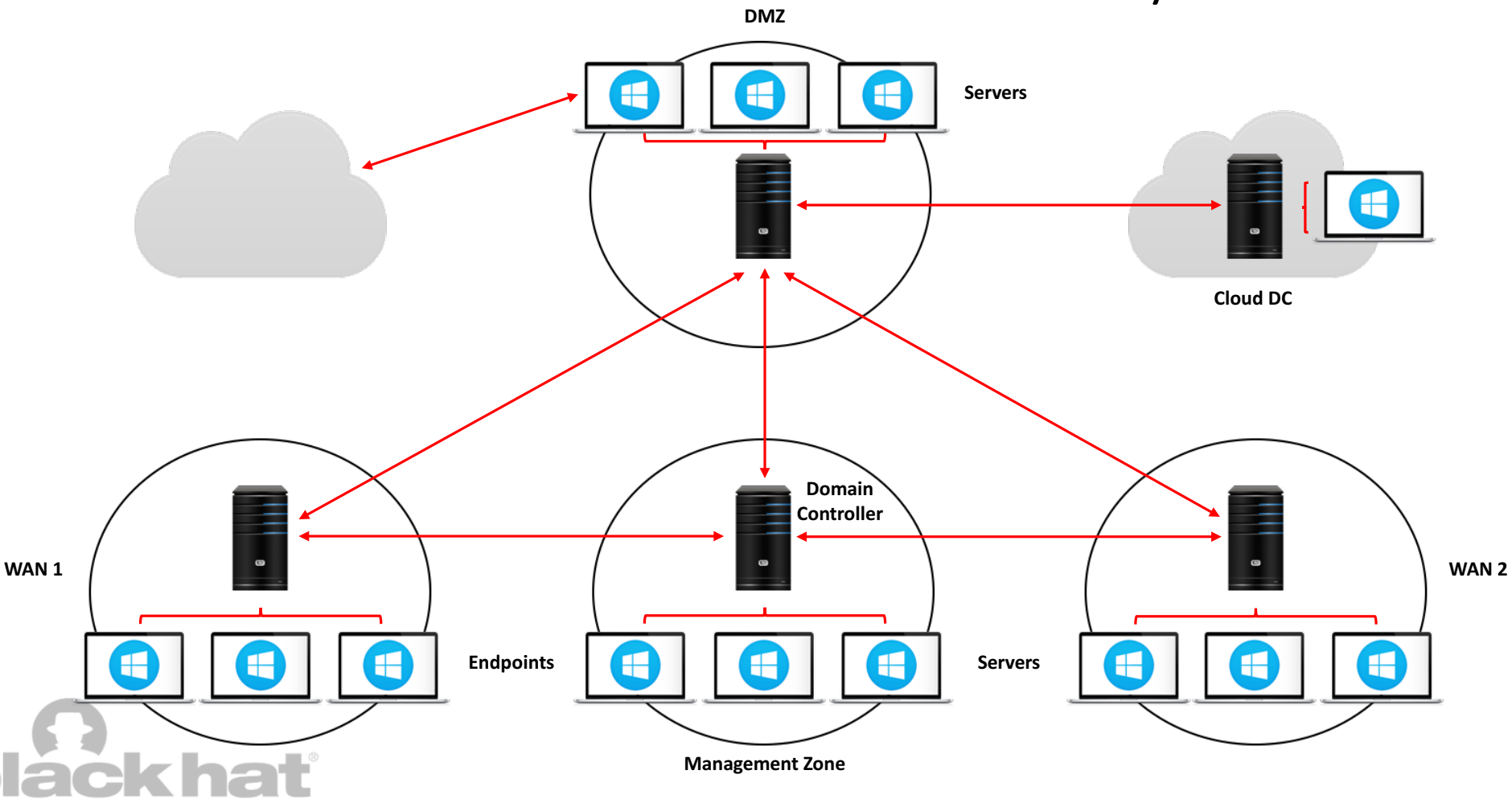
The main challenge is to reliably connect out to the internet-based C2 servers

But what if … ?

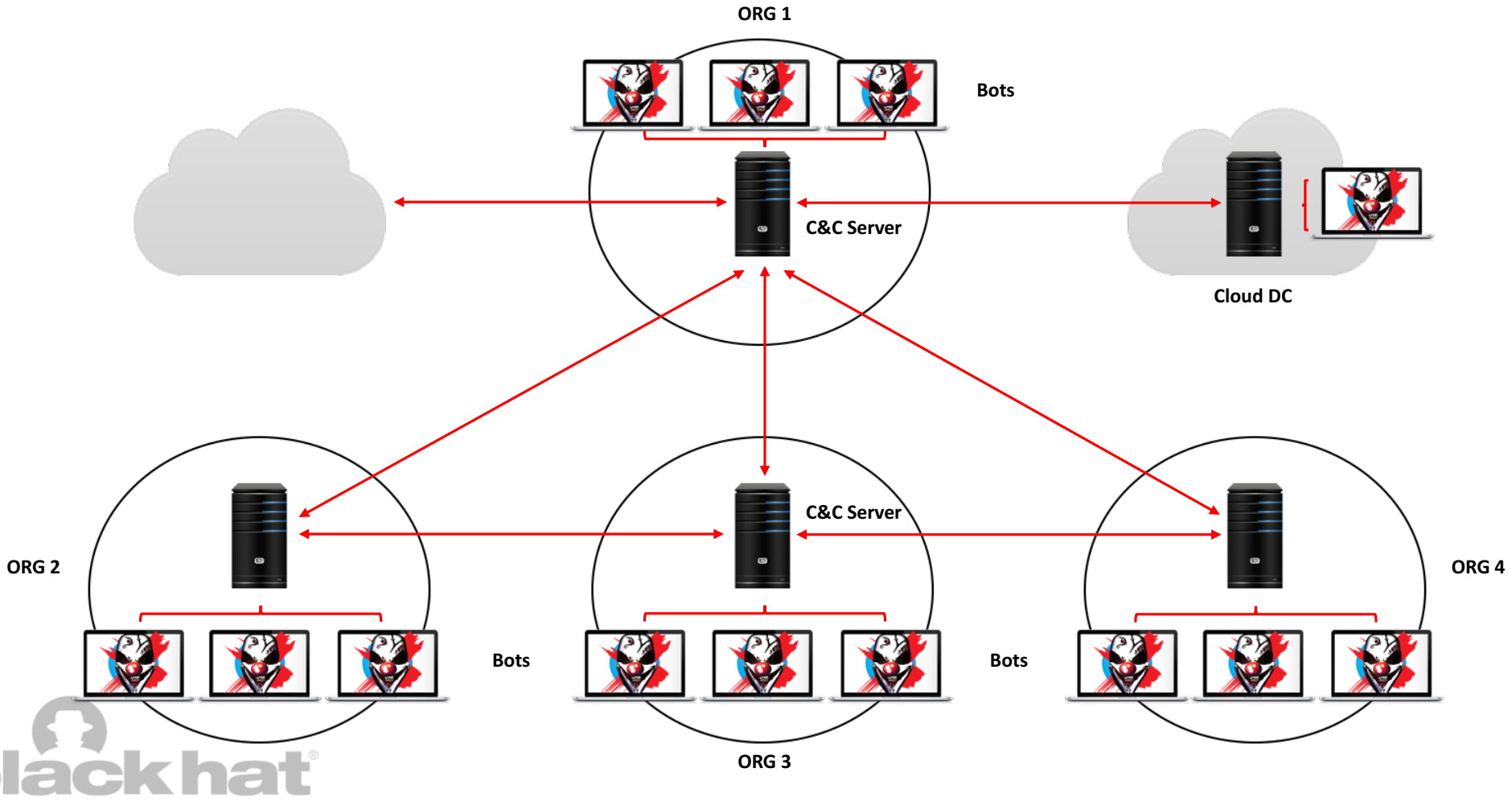
- What if the C2 servers exist inside your internal network?
- What if the C2 servers exist as a part of your critical infrastructure?
- What if the C2 servers use your production services for communication?
- What if the C2 servers can bypass your internal firewalls and network segmentation to communicate with all hosts?
- What if the C2 servers can communicate with remote attackers using your production cloud?

The Active Directory Botnet

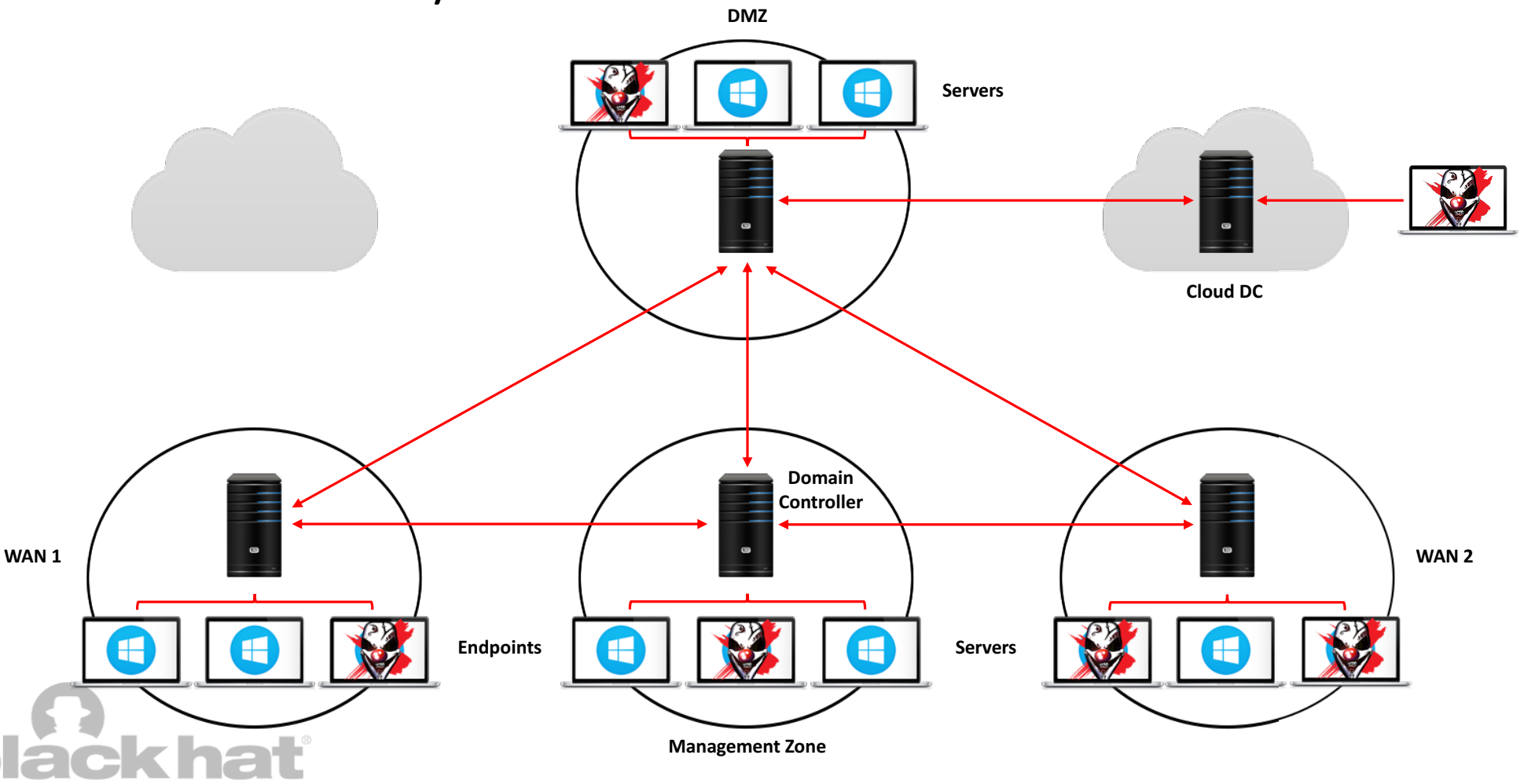
Common Architecture: Active Directory



Common Architecture: Botnet



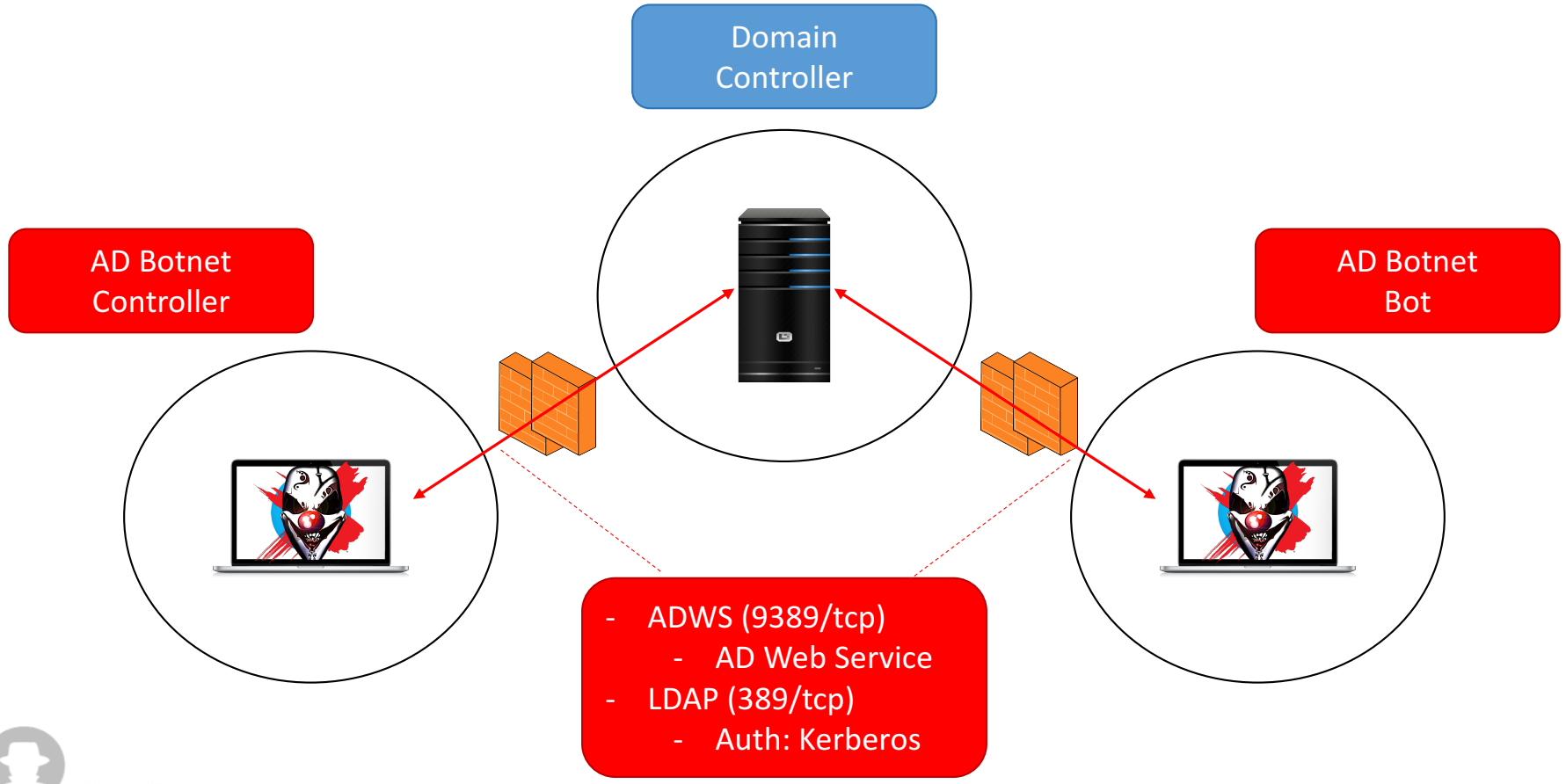
Active Directory Botnet Architecture



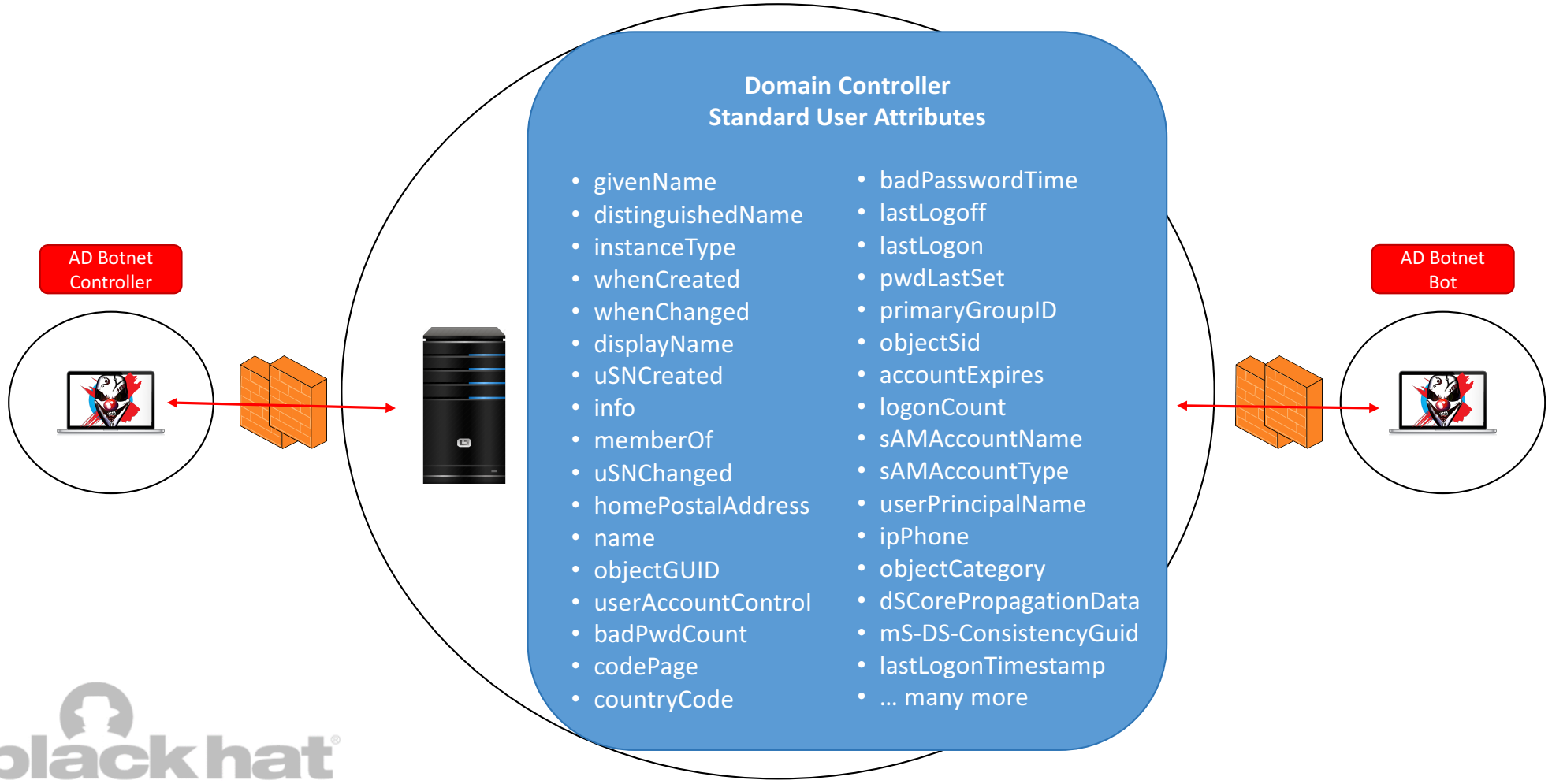
Active Directory as a suitable C2 channel

- AD is a central authentication and access control point for organizations
- All end user devices need connectivity to AD for authentication
- All servers (or most) need connectivity to AD for authentication
- This means that AD is a central connectivity point for all systems
- This introduces the capability to bypass all network-layer security using AD
- All users can (by default) write data into their own account attributes
- When AD integrates with Azure AD, then direct remote controls is possible

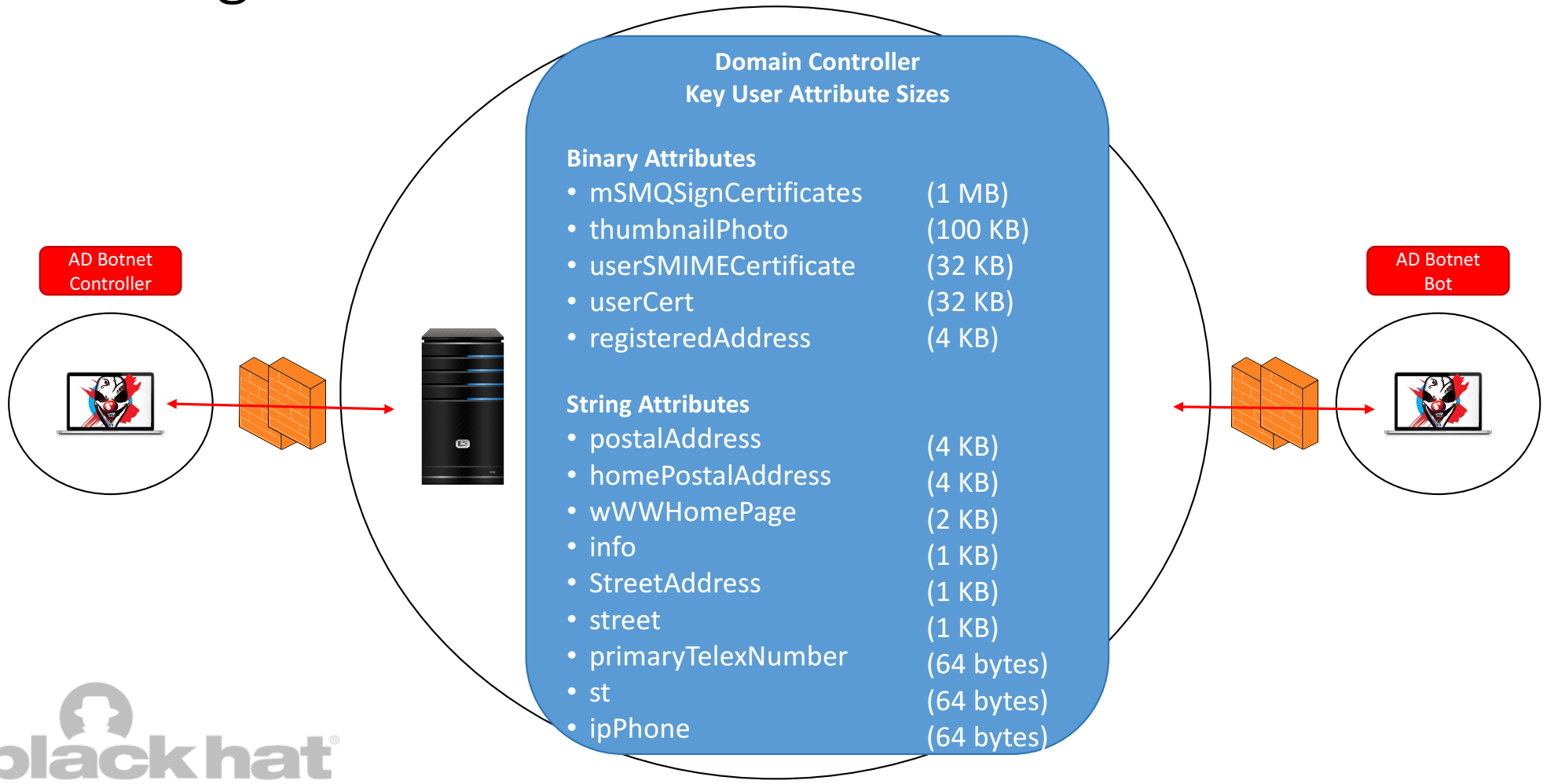
How does the AD Botnet Work?



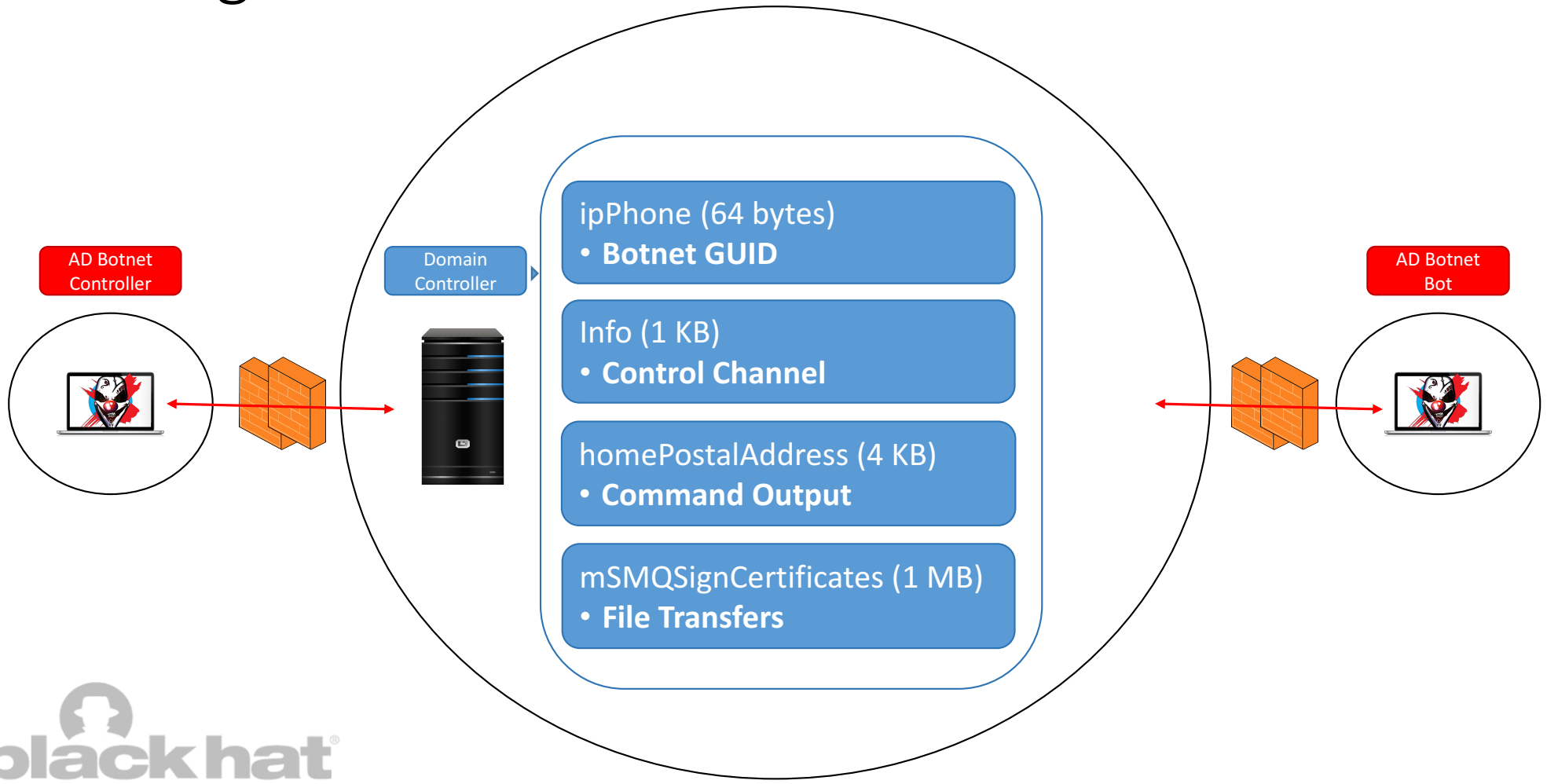
AD Standard User Attributes



Abusing AD Standard User Attributes



Abusing AD Standard User Attributes



Bot Registration Process

- ipPhone (Botnet GUID)
  - AD Botnet GUID stored in ipPhone attribute
  - Enables bots to search AD to identify other members of the AD Botnet
- homePostalAddress (Command Output)
  - This attribute is used to return the Command Output to the calling bot
  - This attribute is simply initialized to a known value
  - The attribute used for the “Command Output” is configurable

Bot Registration Process

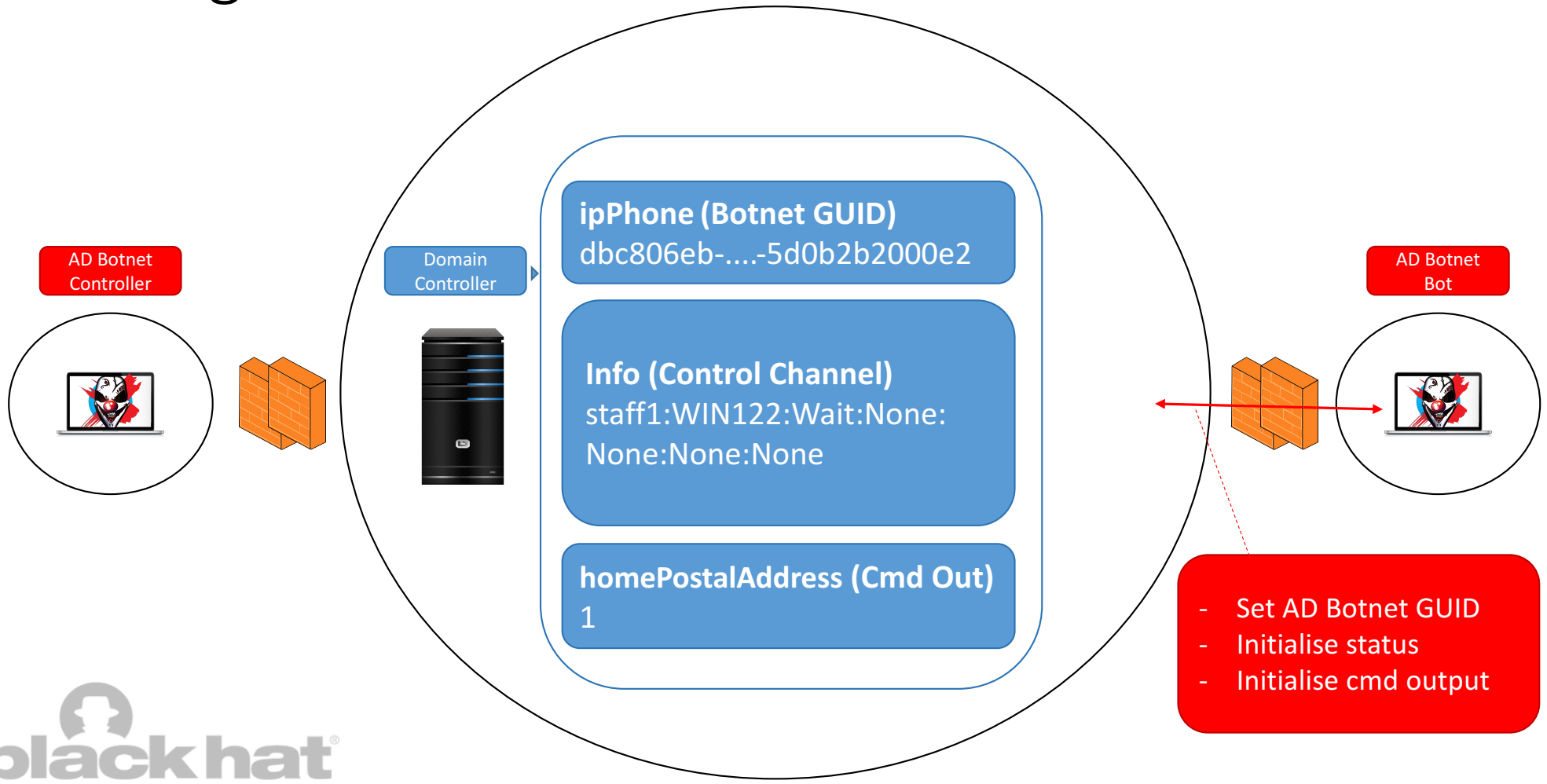
- Info Attribute (Control Channel)

<Username>:<Hostname>:<BotState>:<DstUser>:<DstHost>:<CommandID>:<Cmd>

- Username: User the bot is running as (eg, “staff1”)
- Hostname: Host the bot is running on (eg, “WIN-15PSMN6GMS4”)
- BotState: Current State or Function: Wait, Ack, RunCmd, GetFile, SendShell
- DstUser: Destination user to run the command as
- DstHost: Destination host to run the command on
- CommandID Unique command ID to allow tracking of different commands
- Cmd AD Botnet command to execute

- Example: staff1 : WIN122 : RunCmd : staff2 : WIN184 : 1603000 : base64(ipconfig)

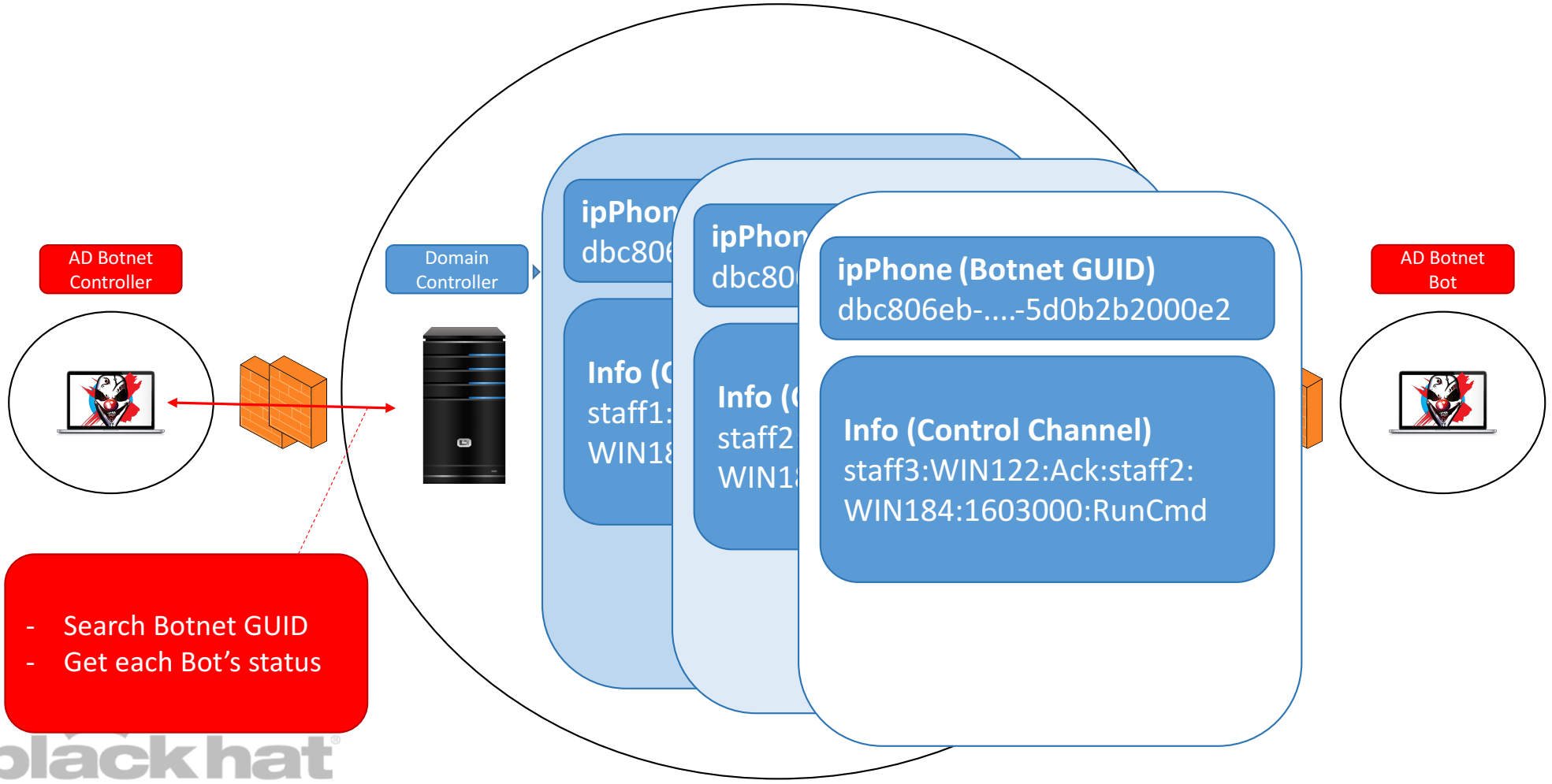
Bot Registration Process



List All Bot Details

- AD Botnet Controller
  - AD Botnet GUID stored in “ipPhone” attribute
  - Search AD accounts where the “ipPhone” attribute contains the AD Botnet GUID
  - Retrieve “info” attribute of each Bot
  - Display User, Host and Status details of each Bot

List All Bot Details



Send Command to Bot

- AD Bot Controller
  - Bot Master selects the Bot to run the command on
  - Reads a command to be executed on the Bot
  - Generates a unique Command ID for tracking
  - Updates the info Attribute status to “RunCmd” and Cmd field to contain base64 command
    - Username: staff2
    - Hostname: WIN-Q84272PAIQD (Controller Hostname)
    - BotState: RunCmd
    - DstUser: staff3
    - DstHost: WIN-R3RCIAMC2AF (Target Bot Hostname)
    - CommandID 4723000
    - Cmd aQBwAGMAbwBuAGYAaQBnAA== … base64(ipconfig)
  - Polls the target Bot’s “info” attribute for an “Ack” status to acknowledge receipt of the cmd

Send Command to Bot

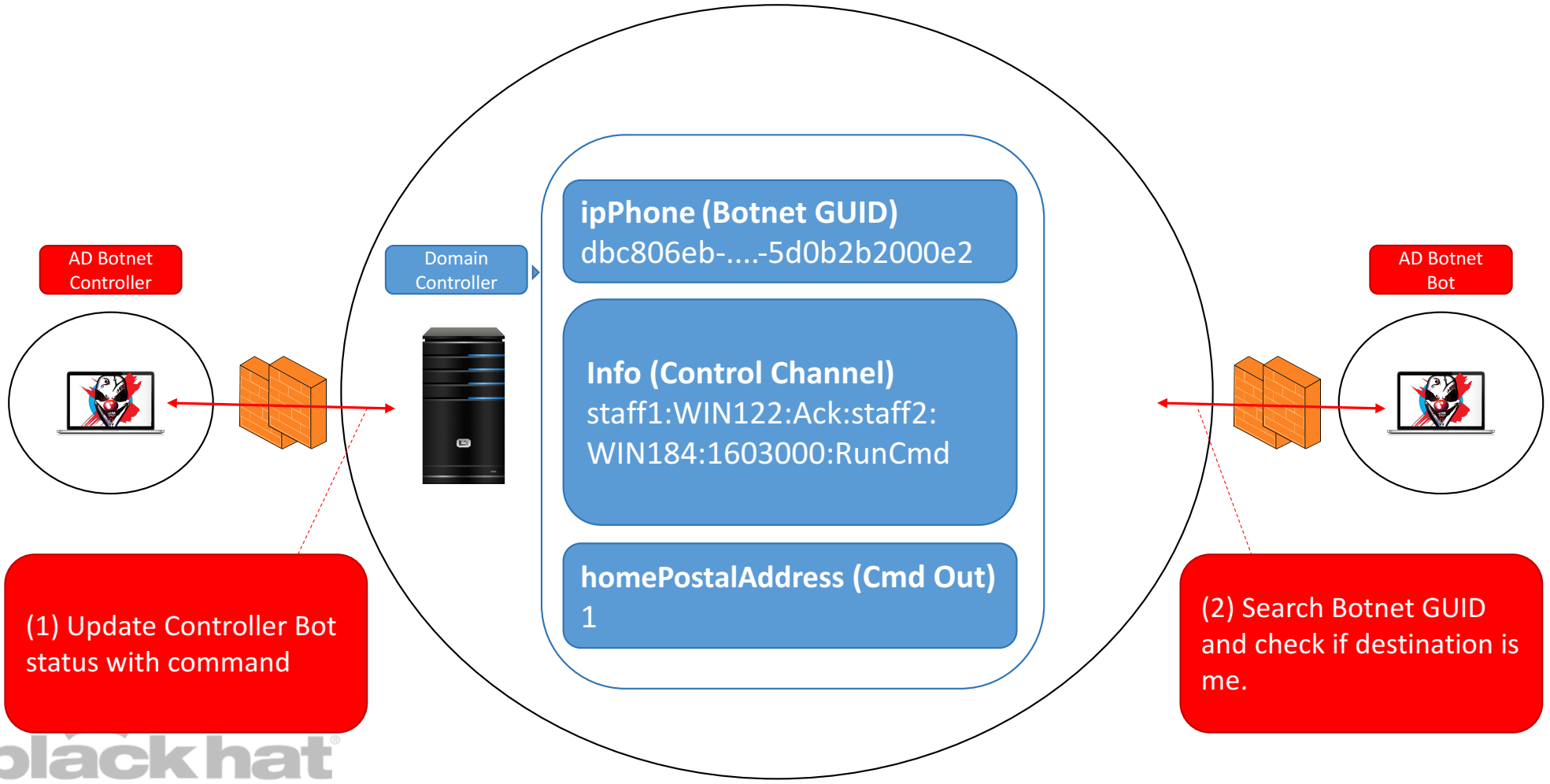
- AD Bot
  - Search AD accounts and retrieves “info” attribute of each Bot
  - Checks if the “DstUser” and “DstHost” fields match itself
  - Checks the “BotState” field for the feature to carry out (RunCmd)
  - Base64 decodes the “Cmd” field and runs the command
  - Updates its own “info” attribute status to “Ack” to acknowledge receipt of the command
- AD Bot Controller
  - Updates its own “info” attribute status to “SendOut” so the Bot knows it is ready

Send Command to Bot

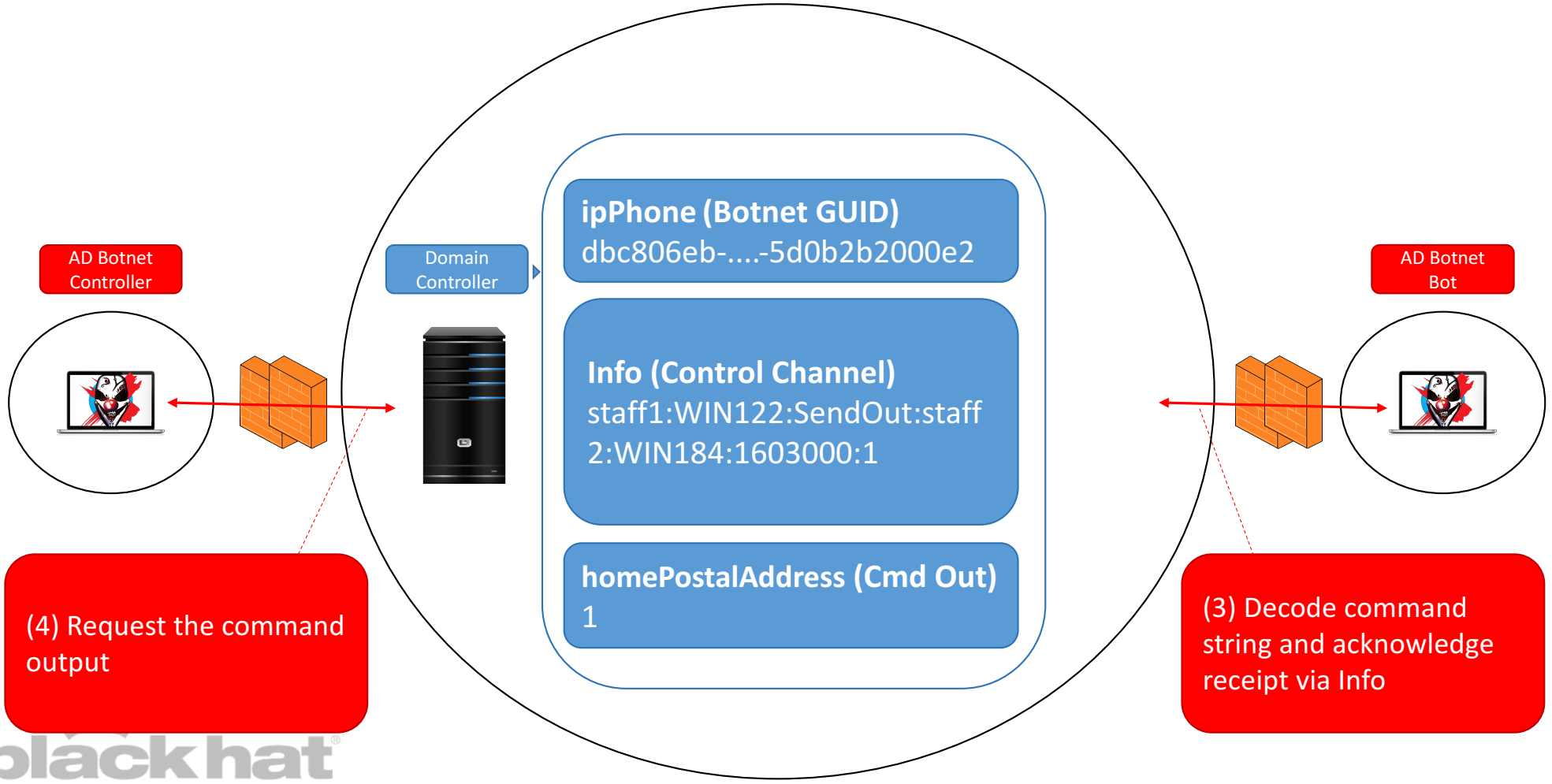
- AD Bot
  - Base64 encodes command output
  - Saves encoded command output into its “homePostalAddress” attribute
  - Once received, updates its status to be “Complete”
- AD Bot Controller
  - Updates its own “info” attribute to acknowledge receipt of the command output
  - Decodes the base64 and displays the command output

The “Interactive Shell” basically loops around this process to simulate a shell.

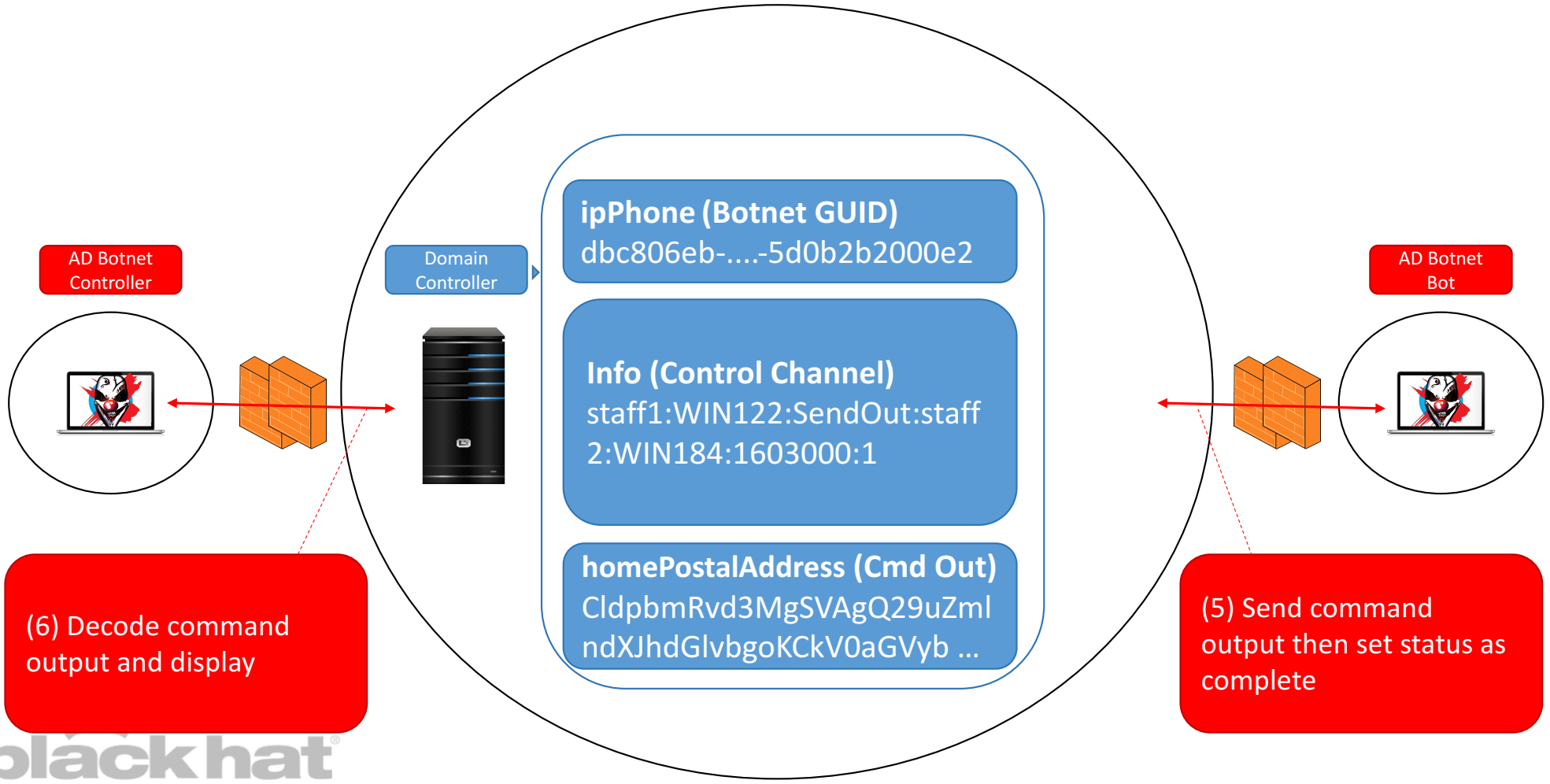
Send Command to Bot



Send Command to Bot



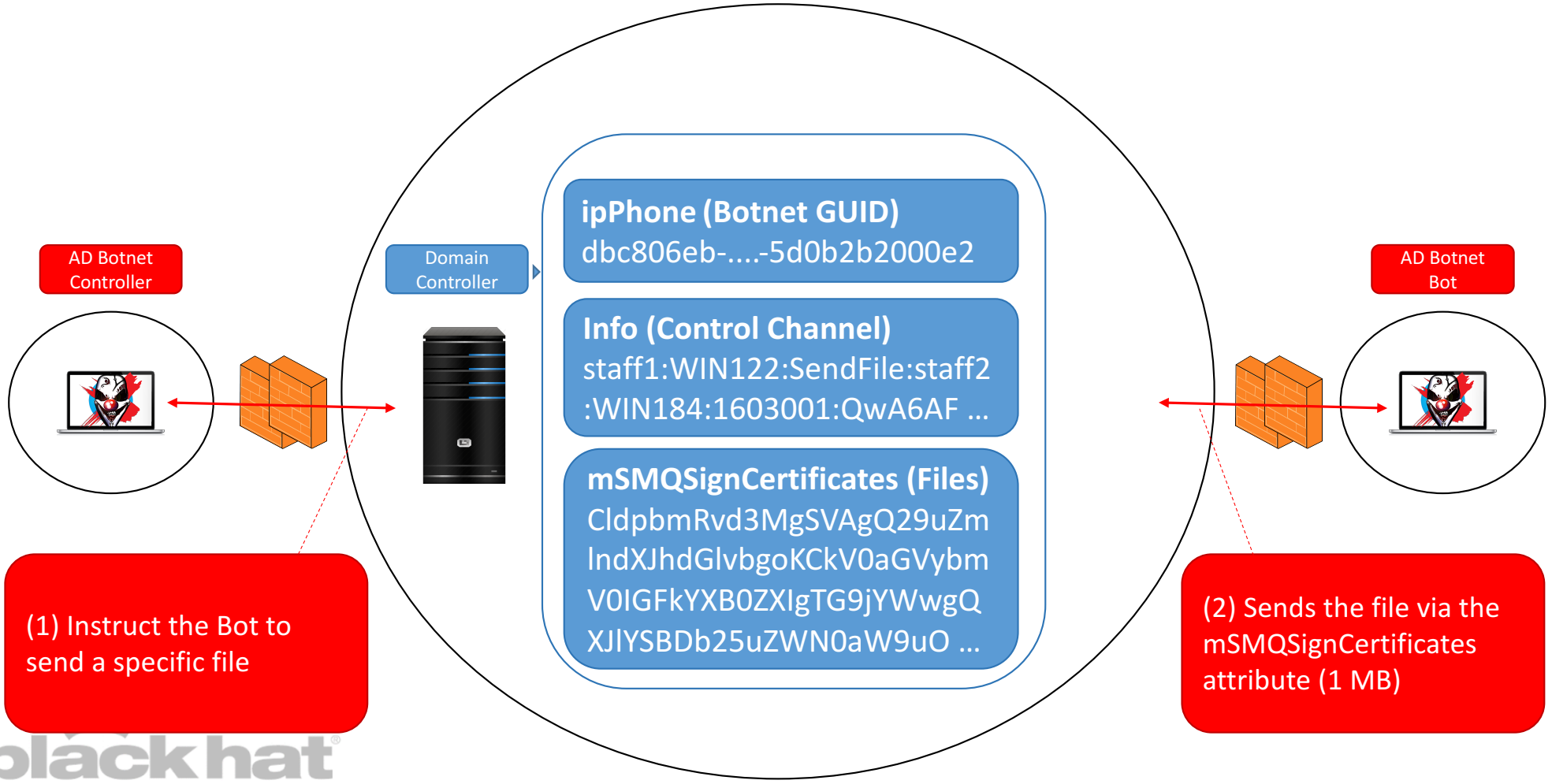
Send Command to Bot



Download File from Bot

- AD Bot Controller
  - Specify the file path to download from and where to save the file
  - Status updated for the destination host and instruction “SendFile” and Base64 encoded path of the file to download in the “Cmd” field
- AD Bot
  - Repeats a similar process to the command execution
  - Difference is that the file is returned in the “mSMQSignCertificates” attribute (1 MB)

Download File from Bot



Live Demo

- Register Bots
- List all active Bots
- Remotely execute a command on a Bot
- Gain an interactive shell on a Bot
- Download a file from a Bot

Remote Command & Control Options

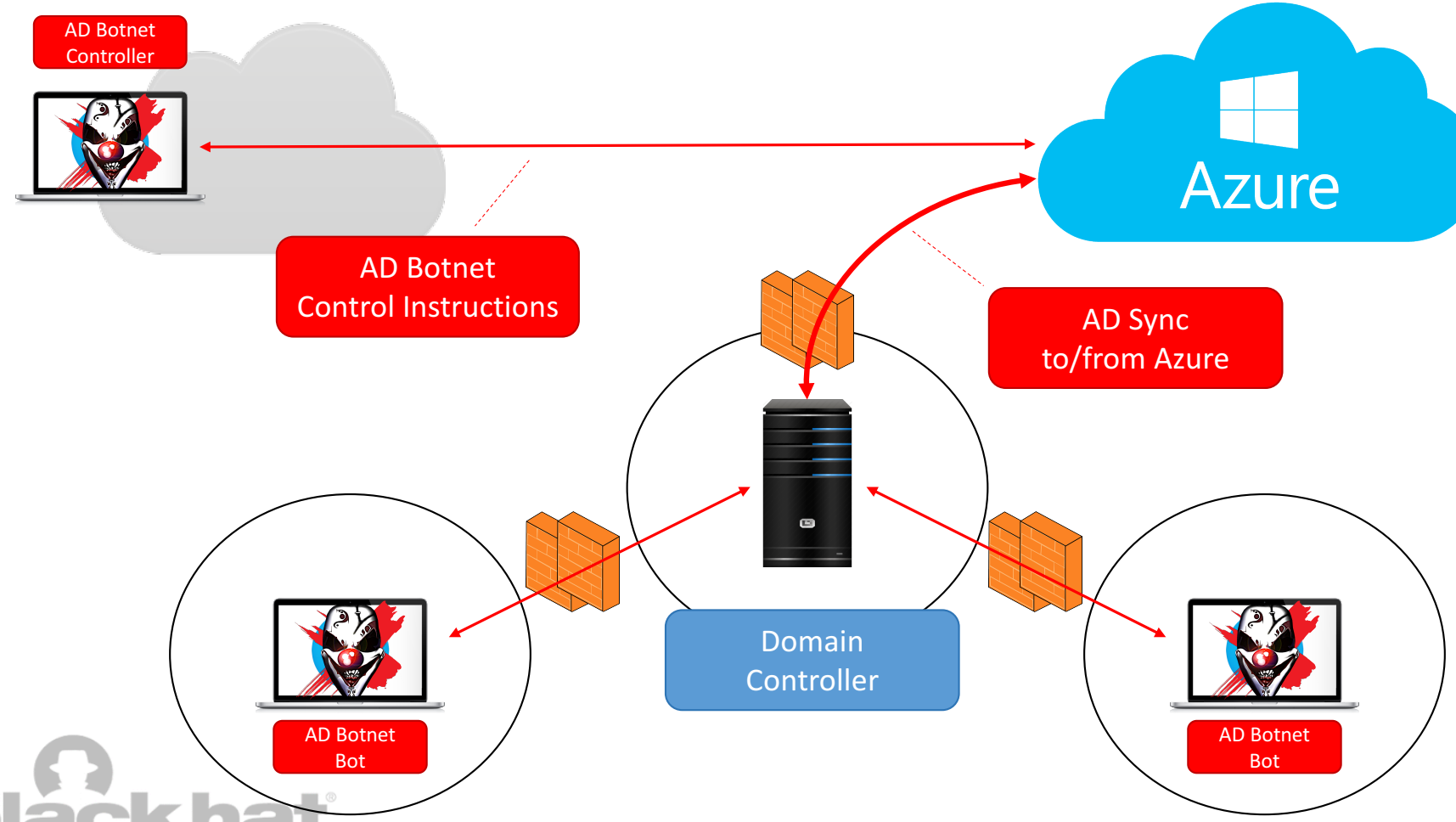
- Microsoft provide an Azure cloud-hosted API into your Active Directory known as “GraphAPI”
- This lets you interact with your AD from the Internet with any standard user account!
- Unfortunately (for us) Azure AD doesn’t sync attributes back to your on-premise AD (for now)
- However, your on-premise AD does sync attributes out to Azure AD … Data Exfiltration!
- AD Botnet has a feature “Xfiltrate Data”

Xfiltrate Data

- AD Botnet pushes data into an attribute
- AD Sync's to Azure AD
- Authenticate as standard user to GraphAPI
- List the user attributes
- Extract the data

```
========== Select Option ==========
1: Press '1' to List bots
2: Press '2' to send command to bot
3: Press '3' start shell on bot
4: Press '4' to download file from bot
5: Press '5' Xfiltrate Data
6: Press '6' Upload File to bot
Q: Enter 'Q' to quit
Enter Selection:
```

Xfiltrate Data



Remote Command & Control Options

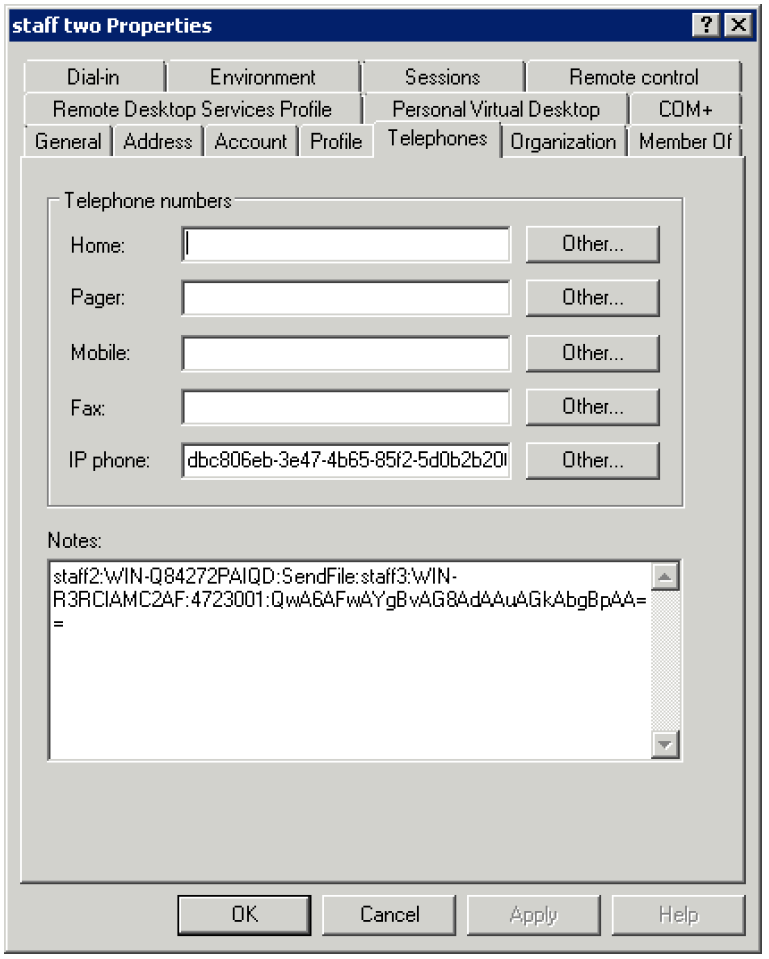
- What other options do we have?
  - AD Botnet Reverse TCP Handler
    - Connect out to a system on the internet
    - Tunnel shell through AD to an internal bot
  - AD Botnet Bind TCP Handler
    - Setup a local bind handler on the bot (in DMZ)
    - Connect from a system on the internet
    - Tunnel shell through AD to an internal bot

Live Demo

- AD Botnet Reverse TCP Handler
  - Connect out to a system on the internet
  - Tunnel shell through AD to an internal bot
- AD Botnet Bind TCP Handler
  - Setup a local bind handler on the bot (in DMZ)
  - Connect from a system on the internet
  - Tunnel shell through AD to an internal bot

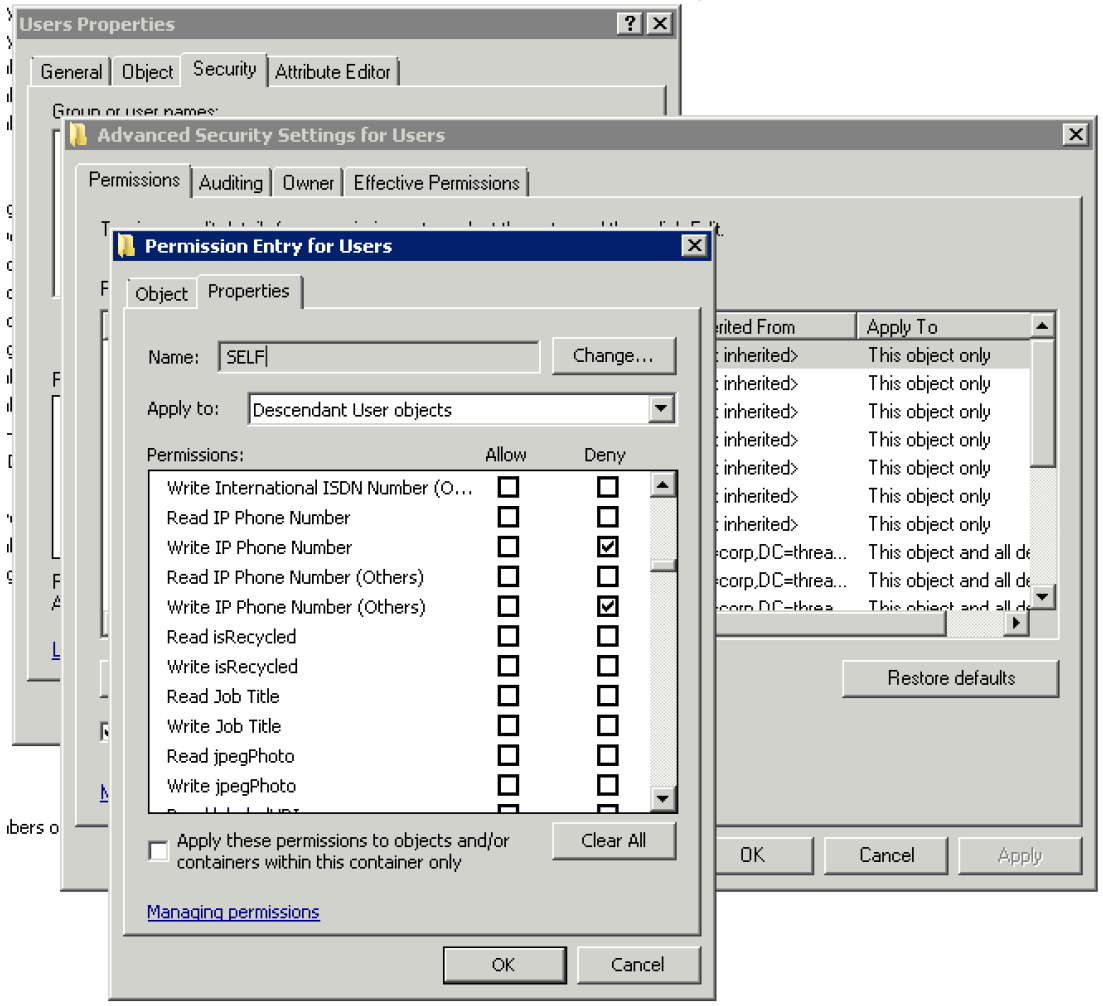
Mitigating the AD Botnet Attack

- Separating your domain into different domains based on security roles. This prevents users in one domain bypassing network filtering to escalate their privileges.
- Noticing odd values in fields …
- Monitoring regular changes of “Personal Information” attributes



Mitigating the AD Botnet Attack

- Locking down permissions for standard users to update their own “Personal Information” attributes



Credits / References

- @harmj0y
  - http://www.harmj0y.net/blog/powershell/command-and-control-using-active-directory/

- Threat Intelligence
  - The AD Botnet concept was thought up internally at Threat Intelligence by Ty Miller in 2014 and investigated at the time to identify if the attack existed, which resulted in no other references.
  - Development of the AD Botnet began in late 2016 by Paul Kalinin of Threat Intelligence
  - We came across the above blog post in Q2 of 2017
    - We found that different approaches were used for the communications, both using similar AD attributes, but Harmj0y injects PowerShell into a single attribute, whereas the AD Botnet uses multiple attributes for botnet registration, command channel, data channel, file transfers, and socket communication data transfer.
  - We had planned to contact Harmj0y and mention his contribution the presentation, which was missed in the understandably busy preparations leading up to the presentation
    - Apologies @harmj0y and thanks for sharing your work!

Credits / References

- The AD Botnet includes a range of additional features including:
  - Multi-Botnet Support
  - Bot Registration
  - Multi-Attribute Control Channel
  - Multi-Attribute Data Channels
  - Individual Bot Single Command Execution
  - Simulated Interactive Bot Command Shell
  - AD Botnet Bind Handler (Socket-based Bot Communication and Remote Port Forwarding)
  - AD Botnet Reverse TCP Handler (Reverse Socket-based Bot Communication and Remote Port Forwarding)
  - Azure AD and Graph API Integration:
  - Azure AD and Graph API One-Way Remote Data Egress
  - Azure AD and Graph API Two-Way Remote AD Botnet Command and Control including “Remote Sockets” (when Azure AD Connect is used)

… and more features to come!

The Active Directory Botnet

Thank you for attending

Ty Miller

Managing Director

Threat Intelligence Pty Ltd

ty.miller@threatintelligence.com

www.threatintelligence.com

Paul Kalinin

Senior Security Consultant

Threat Intelligence Pty Ltd

paul.kalinin@threatintelligence.com

www.threatintelligence.com