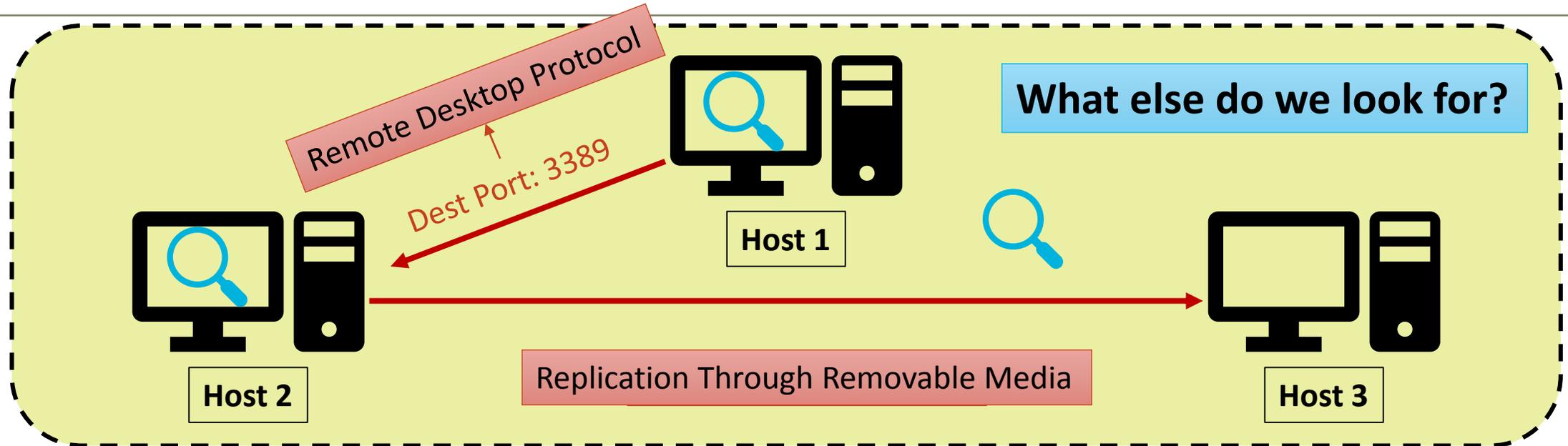


Finding Dependencies Between Adversary Techniques

Andy Applebaum
@andyplayse4

FIRST Conference
June 19th, 2019

An Example Scenario



- Credential Dumping on Host 1 (Credential Access)
- Valid Accounts on Host 1 or Host 2 (Persistence, Privilege Escalation)
- ~~Replication Through Removable Media from Host 2 to Host 3 (Lateral Movement)~~
- Windows Admin Shares from Host 2 to Host 3 (Lateral Movement)

Understanding Intuition

- **Adversaries rarely execute techniques as one-offs**

Account Discovery

Exfiltration over C2 Channel

Credential Dumping

Remote Desktop Protocol

Understanding Intuition

- **Adversaries rarely execute techniques as one-offs**
 - Instead, adversaries typically leverage *chains of techniques* to achieve their desired effect



If we can understand how adversaries construct these chains, then we can better optimize our defenses

Behind the Chains: Technique Relationships

Dependence

One technique helps enable executing another in the future

Credential Dumping



Remote Desktop Protocol

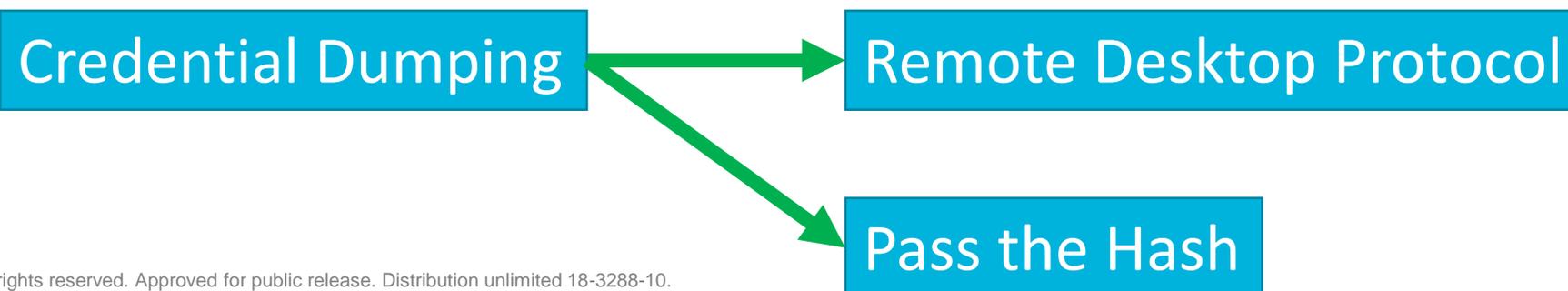
Behind the Chains: Technique Relationships

Dependence

One technique helps enable executing another in the future

Alternative

A technique achieves a similar goal and shares dependencies with another, but can be executed in a different context



Behind the Chains: Technique Relationships

Dependence	One technique helps enable executing another in the future
Alternative	A technique achieves a similar goal and shares dependencies with another, but can be executed in a different context
Implementation Overlap	Implementations of one technique also implement another

net localgroup administrators

Account Discovery

Permissions Group Discovery



Behind the Chains: Technique Relationships

Dependence

One technique helps enable executing another in the future

Alternative

A technique achieves a similar goal and shares dependencies with another, but can be executed in a different context

Implementation
Overlap

Implementations of one technique also implement another

Same Target

Techniques apply to the same system(s), but have no other notable relationship

LLMNR/NBT-NS Poisoning and Relay

Control Panel Items

Behind the Chains: Technique Relationships

Dependence

One technique helps enable executing another in the future

Alternative

A technique achieves a similar goal and shares dependencies with another, but can be executed in a different context

Implementation
Overlap

Implementations of one technique also implement another

Same Target

Techniques apply to the same system(s), but have no other notable relationship

This talk: Primarily *Dependence*, with some *Alternative* + *Implementation*

Why Technique Relationships are Important

How could we use this knowledge?

Hunting

Dependent: hunt for techniques that enable your hypothesis

Alternative: if the hypothesis fails, hunt for a reasonable alternative

Detection

Dependent: develop high-fidelity rules by correlating dependent and independent techniques

Alternative: correlate technique execution failures with follow-up alternatives

Security Engineering

Dependent: configure endpoints to prevent techniques that enable others

Alternative: collect appropriate logs to cover related sets of alternatives

Why related *techniques*?

Examples of non-technique detection

Hunting for File Hashes

- If I've seen a hash that's associated with other hashes, I can hunt for the others to confirm my hypothesis

Engineering Against Bad IP Addresses

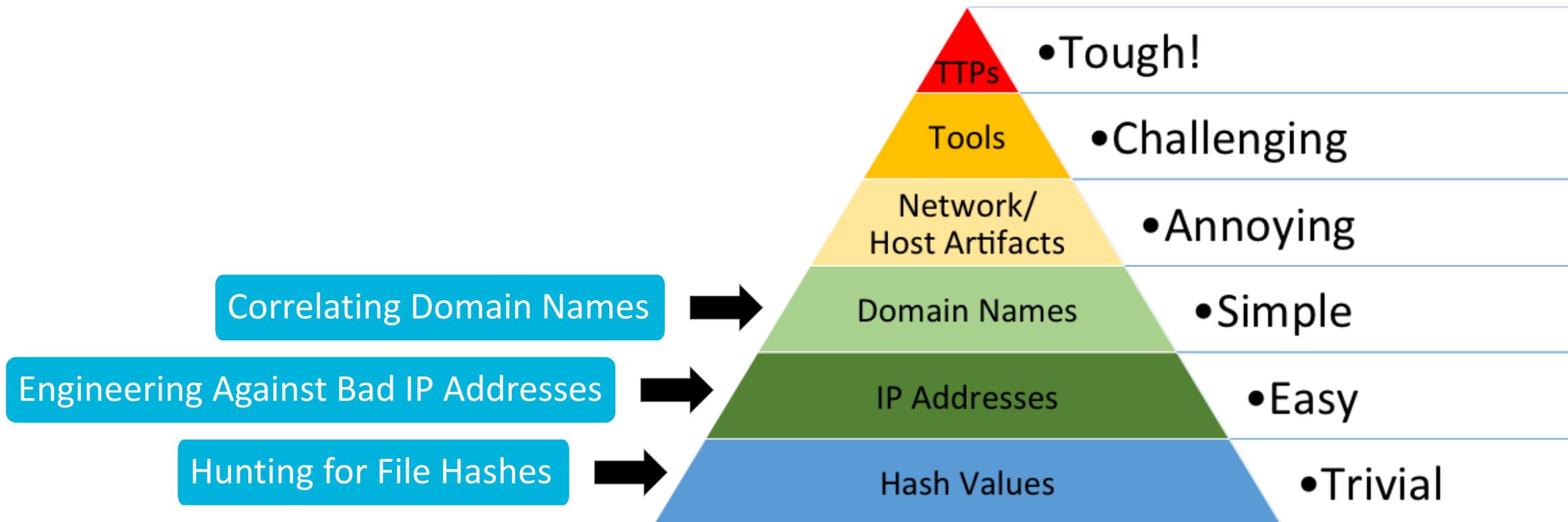
- Block IP address space corresponding to bad ASNs

Correlating Domain Names

- Create rules that correlate across WHOIS information to detection malicious domains

Why related techniques?

Answer: The Pyramid of Pain



Source: David Bianco

<https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html>

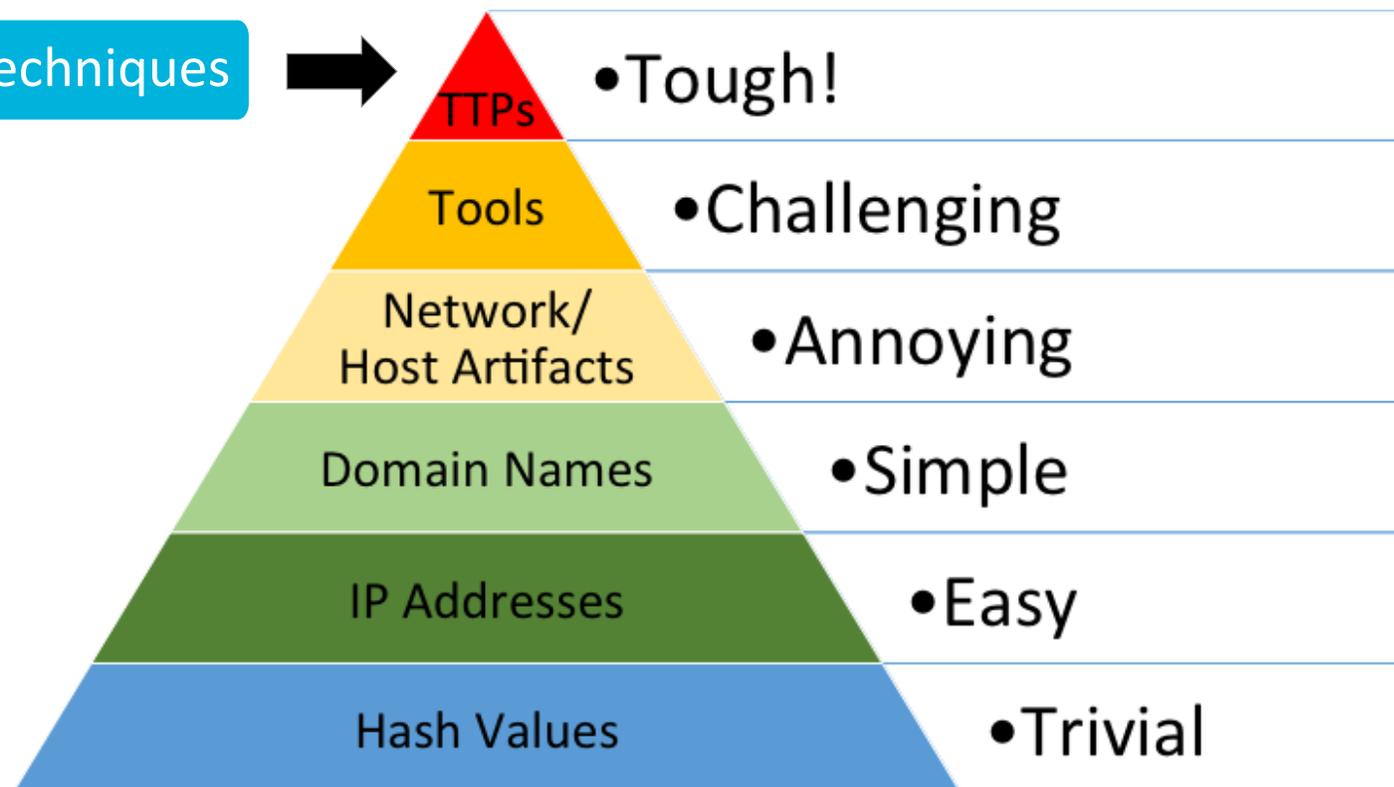
TTPs = Tactics, Techniques, and Procedures

Why related techniques?

Answer: The Pyramid of Pain

By finding related adversary techniques, we can key in on the things that are hardest for adversaries to change

Finding Related Techniques



Source: David Bianco

<https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html>

TTPs = Tactics, Techniques, and Procedures

Talk Outline

- **Assumed premise: Understanding the relationships between techniques can help us enhance our defenses**
- **This talk: how do we actually find the relationships between techniques?**
 - Three studies showing how we can find technique relationships
 - Data driven, using threat reporting
 - Semantic, using logical modeling
 - Experimental, using actual data
- **Take-aways:**
 - Importance of technique relationships
 - Ways you can identify technique relationships (and what the tradeoffs are)
- **Bonus: data and software used here is publicly available**
 - Experiments and analysis can be replicated and modified

Finding Related Techniques

Data Analysis Using Threat Reporting

An Example Report

"url": "https://cdn2.hubspot.net/hubfs/3354902/Cybereason%20Labs%20Analysis%20Operation%20Cobalt%20Kitty.pdf",



4.2. Information gathering commands

The attackers used several tools built into the Windows OS to gather information on the environment's network and its users. Those tools included netsh, ipconfig, netstat, arp, net user/group/localgroup, nslookup and Windows Management Instrumentation (WMI).

The following are a few examples of command line arguments that were used to gather information on the infected hosts and the network:

Command	Purpose
net localgroup administrators	Enumerating admin users

©2017 Cybereason Inc. All rights reserved.

Account Discovery

5.1. Obtaining credentials

Before the attackers could spread to new machines, they had to obtain the necessary credentials, such as passwords, NTLM hashes and Kerberos tickets. To obtain these credentials, the attackers used various, known tools to dump locally stored credentials.

Credential Dumping

5.2. Pass-the-hash and pass-the-ticket

Cybereason detected multiple lateral movement techniques that were used during the attack. The attackers successfully carried out [pass-the-hash](#) and [pass-the-ticket](#) attacks using stolen NTLM hashes and Kerberos tickets from compromised machines.

Pass-the-Hash

Pass-the-Ticket

5.3. Propagation via Windows Admin Shares

Another lateral movement technique that was used extensively in the attack involved using the [Windows Admin Shares](#) via the built-in Windows "net.exe" tool. This technique uses Windows' hidden network shares, which administrators can only access and use to copy their tools to remote machines and execute them.

Windows Admin Shares

Looking at Relationships

To laterally move via **Pass-the-Hash** , **Pass-the-Ticket** , or **Windows Admin Shares** :

- gain access to credentials with **Credential Dumping**
- discover admins on the target with **Account Discovery**

Measuring Relationship Frequency: The Idea

Two techniques are likely related if they are frequently mentioned alongside each other in threat reports

...but how do we identify techniques in reports?

ATT&CK: A Technique Corpus

Publicly Available
attack.mitre.org

Tactics – Adversary’s technical goal

Techniques – How goal is achieved

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	CredentialAccess	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise		Scheduled Task		Binary Padding		Network Sniffing		AppleScript		Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	Launchctl		Access Token Manipulation		Account Manipulation	Account Discovery	Application Deployment	Audio Capture	Commonly Used Port	Data Encrypted	Data Encrypted for Impact
External Remote Services	Local Job Scheduling		Bypass User Account Control		Bash History	Application Window Discovery	Software	Automated Collection	Communication Through Removable Media	Data Compressed	Defacement
Hardware Additions	LSASS Driver		Extra Window Memory Injection		Brute Force	Discovery	Distributed Component Object Model	Clipboard Data	Connection Proxy	Data Transfer Size Limits	Disk Content Wipe
Replication Through Removable Media	Trap		Process Injection		Credential Dumping	Browser Bookmark Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Exfiltration Over Other Network Medium	Disk Structure Wipe
Spearpishing Attachment	AppleScript		DLL Search Order Hijacking		Credentials in Files	Domain Trust Discovery	Logon Scripts	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Spearpishing Link	CMSTP		Image File Execution Options Injection		Credentials in Registry	File and Directory Discovery	Remote Services	Data from Network Shared Drive	Exploitation of Alternative Protocol	Firmware Corruption	Inhibit System Recovery
Supply Chain Compromise	Command-Line Interface		Plist Modification		Exploitation for Credential Access	Network Service Scanning	Pass the Hash	Data from Removable Media	Data Encoding	Exfiltration Over Alternative Protocol	Network Denial of Service
Trusted Relationship	Compiled HTML File		Valid Accounts		Forced Authentication	Network Share Discovery	Pass the Ticket	Data Staged	Data Obfuscation	Exfiltration Over Physical Medium	Resource Hijacking
Valid Accounts	Control Panel Items		Accessibility Features		Hooking	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Fronting	Scheduled Transfer	Runtime Data Manipulation
	Dynamic Data Exchange		AppCert DLLs		Input Capture	Peripheral Device Discovery	Remote File Copy	Man in the Browser	Domain Generation Algorithms		Service Stop
	Execution through API		AppInit DLLs		Input Prompt	Permission Groups Discovery	Replication Through Removable Media	Screen Capture	Fallback Channels		Stored Data Manipulation
	Module Load		Dylib Hijacking		Kerberoasting	Process Discovery	Video Capture	Multiband Communication	Multi-hop Proxy		Transmitted Data Manipulation
	Exploitation for Client Execution		File System Permissions Weakness		Keychain	Query Registry	Shared Webroot		Multi-layer Encryption		
	Graphical User Interface		Hooking		Component Object Model Hijacking	LLMNR/NBT-NS Poisoning and Relay	SSH Hijacking		Multi-Stage Channels		
	InstallUtil		Launch Daemon		Control Panel Items	Password Filter DLL	Taint Shared Content		Port Knocking		
	Mshsta		New Service		DCShadow	Private Keys	Third-party Software		Remote Access Tools		
	PowerShell		Path Interception		Deobfuscate/Decode Files or Information	Securityd Memory	Windows Admin Shares		Remote File Copy		
	Regsvcs/Regasm		Port Monitors		Service Registry Permissions Weakness	Two-Factor Authentication Interception	Windows Remote Management		Standard Application Layer Protocol		
	Regsvr32		Service Registry Permissions Weakness		Disabling Security Tools				Standard Cryptographic Protocol		
	Rundll32		Setuid and Setgid		DLL Side-Loading				Standard Non-Application Layer Protocol		
	Scripting		Startup Items		Execution Guardrails				Uncommonly Used Port		
	Service Execution		Web Shell						Web Service		
	Signed Binary Proxy Execution	.bash_profile and .bashrc	Exploitation for Privilege Escalation		Exploitation for Defense Evasion						
	Signed Script Proxy Execution	Authentication Package	SID-History Injection		File Deletion						
	Source	BITS Jobs	Sudo		File Permissions Modification						
	Space after Filename	Bootkit	Sudo Caching		File System Logical Offsets						
	Third-party Software	Browser Extensions			Gatekeeper Bypass						
	Trusted Developer Utilities	Change Default File Association			Group Policy Modification						
	User Execution	Component Firmware			Hidden Files and Directories						
	Windows Management Instrumentation	Component Object Model Hijacking			Hidden Users						
	Windows Remote Management	Create Account			Hidden Window						
	XSL Script Processing	External Remote Services			HISTCONTROL						
		Hidden Files and Directories			Indicator Blocking						
		Hypervisor			Indicator Removal from Tools						
		Kernel Modules and Extensions			Indicator Removal on Host						
		Launch Agent			Indirect Command Execution						
		LC_LOAD_DYLIB Addition			Install Root Certificate						
		Login Item			InstallUtil						
		Logon Scripts			Launchctl						
		Modify Existing Service			LC_MAIN Hijacking						
		Netsh Helper DLL			Masquerading						
		Office Application Startup			Modify Registry						
		Port Knocking			Mshsta						
		Rc.common			Network Share Connection Removal						
		Redundant Access			NTFS File Attributes						

ATT&CK: A Technique Corpus

Publicly Available
attack.mitre.org

Tactics – Adversary’s technical goal

Techniques – How goal is achieved

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise		Scheduled Task		Binary Delivery		Network Sniffing	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	Launchctl		Access Token Manipulation		Account Manipulation	Account Discovery	Application Deployment	Automated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
External Remote Services	Local Job Scheduling		Bypass User Account Control		Bash History	Application Window	Software	Clipboard Data		Data Encrypted	Defacement
Hardware Additions	Trap										
Replication Through Removable Media	AppleScript										
Spearpishing Attachment	CMSTP										
Spearpishing Link	Command-Line Interface										
Spearpishing via Service	Compiled HTML File										
Supply Chain Compromise	Control Panel Items	Accessibility									
Trusted Relationship	Dynamic Data Exchange	AppCert									
Valid Accounts	Execution through API	AppInit									
	Execution through Module Load	Applications									
	Exploitation for Client Execution	Dylib Hijacking									
	Graphical User Interface	File System Permissions									
	InstallUtil	Hook									
	Mshina	Launch D									
	PowerShell	New Se									
	Regsvcs/Regasm	Path Inter									
	Regsvr32	Part Mo									
	Rundll32	Service Registry P									
	Scripting	Setuid an									
	Service Execution	Startup									
	Signed Binary	Web S									
	Proxy Execution	.bash_profile and .bashrc									
	Signed Script	Account Manipulation									
	Proxy Execution	Authentication Package									
	Source	BITS Jobs									
	Space after Filename	Bootkit									
	Third-party Software	Browser Extensions									
	Trusted Developer Utilities	Change Default File Association									
	User Execution	Component Firmware									
	Windows Management Instrumentation	Component Object Model Hijacking									
	Windows Remote Management	Create Account									
	XSL Script Processing	External Remote Services									
		Hidden Files and Directories									
		Hypervisor									
		Kernel Modules and Extensions									
		Launch Agent									
		LC_LOAD_DYLIB Addition									
		Login Item									
		Logon Scripts									
		Modify Existing Service									
		Netsh Helper DLL									
		Office Application Startup									
		Port Knocking									
		RC.common									
		Redundant Access									

Grounded in real data from cyber incidents

Focuses on describing adversary TTPs, not IoCs

Decouples the problem from the solution

(also has information on groups and software)

Finding Related Techniques with ATT&CK: Methodology

- **Straightforward methodology using ATT&CK:**
 - Initialize an array storing the number of references each technique has been reported with each other technique
 - Iterate through each reference in ATT&CK, updating the array

- **Easy to implement: ATT&CK is in STIX**
 - Each technique has references that describe that technique
 - Relationship objects link software or groups to techniques
 - Bonus: freely available in JSON form!

Caveat: Bias

- **Frequency analysis from the ATT&CK corpus suffers from two bias types:**
 - Bias added by the ATT&CK team (report → ATT&CK data)
 - Bias added by the source (i.e., report author)
- **Examples:**
 - ATT&CK bias: we only recall so many techniques during report analysis
 - ATT&CK bias: we're more likely to hone in on new novelties in reports
 - Source bias: sources are more likely to report on novelties
 - Source bias: sources only have vision into what they can detect
- ***It's important to acknowledge these biases before doing analysis!***
 - Results are still useful, but note: they're not ground truth

Caveat: Bias – for more info:

Turning Intelligence into Action with MITRE ATT&CK™

Katie Nickels @likethecoins
Adam Pennington @_whatshisface
MITRE ATT&CK @MITREattack

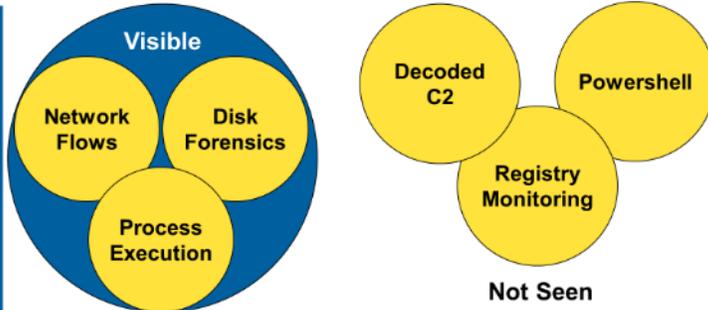


Biases in ATT&CK's Mapped Data

- Important to understand and state our biases in CTI
- Two kinds of bias in technique examples in ATT&CK
 - Bias introduced by us
 - Bias inherent in the sources we use
- Understanding these is the first step in properly leveraging this data



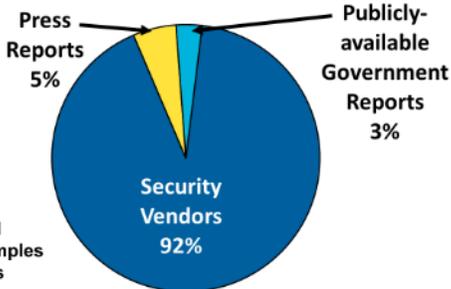
Source Biases: Visibility bias




Source Biases: Novelty Bias




Our Biases: Sources We Select



From reports used for technique examples in ATT&CK Groups



Hedging Our Biases

- Work together
 - Diversity of thought makes for stronger teams
- Adjust and calibrate your data sources
- Add different data sources
- Remember we're prioritizing the *known* over the *unknown*
 - As opposed to absolute comparison



- For more on bias: <https://www.slideshare.net/KatieNickels/first-cti-symposium-turning-intelligence-into-action-with-mitre-attck>

Counting Co-occurrences: Shared References

Credential Dumping	108.0	20.0	10.0	20.0	14.0	34.0
Valid Accounts	20.0	43.0	9.0	13.0	5.0	16.0
Windows Admin Shares	10.0	9.0	31.0	4.0	8.0	7.0
Remote Desktop Protocol	20.0	13.0	4.0	37.0	4.0	10.0
Service Execution	14.0	5.0	8.0	4.0	27.0	10.0
Standard Application Layer Protocol	34.0	16.0	7.0	10.0	10.0	158.0

Counting Co-occurrences: Shared References

There were 20 reports mentioning both Valid Accounts and Credential Dumping

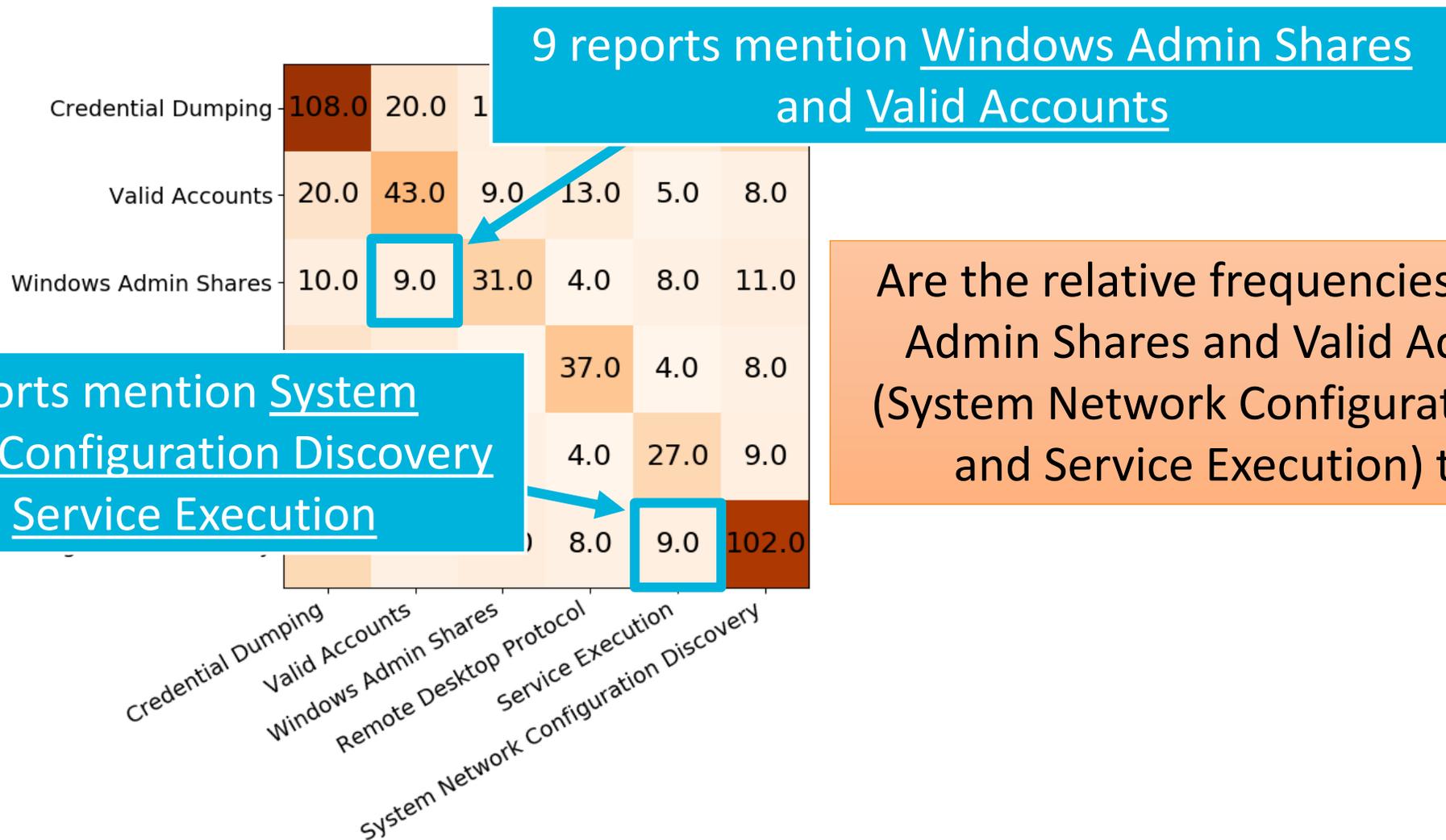
There were 5 reports mentioning both Valid Accounts and Service Execution

There were 14 reports mentioning both Service Execution and Credential Dumping

There were 158 reports mentioning Standard Application Layer Protocol

Credential Dumping	108.0	20.0	10.0	20.0	14.0	34.0
Valid Accounts	20.0	43.0	9.0	13.0	5.0	16.0
Windows Admin Shares	10.0	9.0	31.0	4.0	8.0	7.0
Remote Desktop Protocol	20.0	13.0	4.0	37.0	4.0	10.0
Service Execution	14.0	5.0	8.0	4.0	27.0	10.0
Standard Application Layer Protocol	34.0	16.0	7.0	10.0	10.0	158.0
	Credential Dumping	Valid Accounts	Windows Admin Shares	Remote Desktop Protocol	Service Execution	Standard Application Layer Protocol

Normalization 1: Percentages



Are the relative frequencies of (Windows Admin Shares and Valid Accounts) and (System Network Configuration Discovery and Service Execution) the same?

Normalization 1: Percentages

Credential Dumping	108.0	20.0	10.0	20.0	14.0	26.0
Valid Accounts	20.0	43.0	9.0	13.0	5.0	8.0
Windows Admin Shares	10.0	9.0	31.0	4.0	8.0	11.0
Remote Desktop Protocol	20.0	13.0	4.0	37.0	4.0	8.0
Service Execution	14.0	5.0	8.0	4.0	27.0	9.0
System Network Configuration Discovery	26.0	8.0	11.0	8.0	9.0	102.0

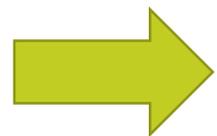


0	100.0	18.5	9.3	18.5	13.0	24.1
1	46.5	100.0	20.9	30.2	11.6	18.6
2	32.3	29.0	100.0	12.9	25.8	35.5
3	54.1	35.1	10.8	100.0	10.8	21.6
4	51.9	18.5	29.6	14.8	100.0	33.3
5	25.5	7.8	10.8	7.8	8.8	100.0

Normalization 1: Percentages

29% of reports mentioning Windows Admin Shares also mention Valid Accounts (9/31)

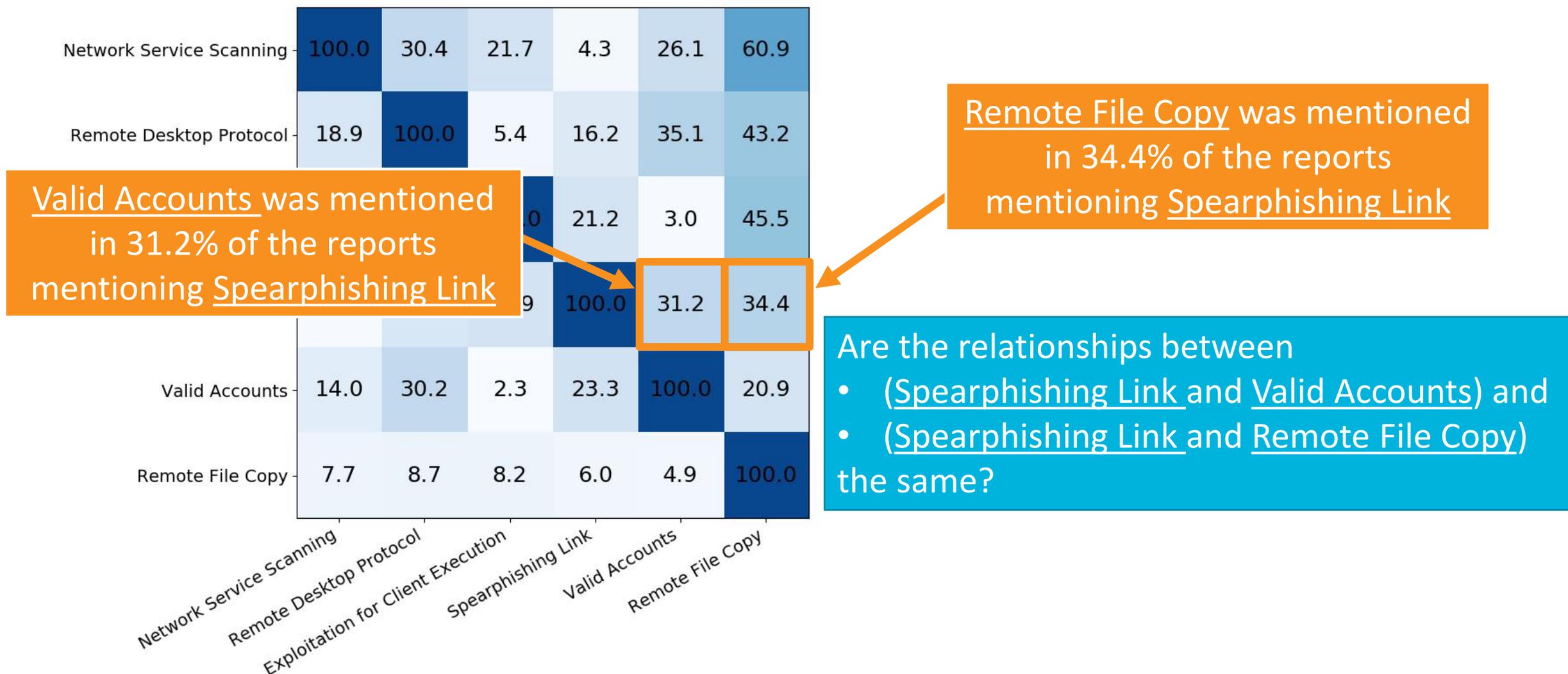
Credential Dumping	108.0	20.0	10.0	20.0	14.0	26.0
Valid Accounts	20.0	43.0	9.0	13.0	5.0	8.0
Windows Admin Shares	10.0	<u>9.0</u>	<u>31.0</u>	4.0	8.0	11.0
Remote Desktop Protocol	20.0	13.0	4.0	37.0	4.0	8.0
Service Execution	14.0	5.0	8.0	4.0	27.0	9.0
System Network Configuration Discovery	26.0	8.0	11.0	8.0	<u>9.0</u>	<u>102.0</u>



1	46.5	100.0	20.9	30.2	11.6	18.6
2	32.3	<u>29.0</u>	100.0	12.9	25.8	35.5
3	54.1	35.1	10.8	100.0	10.8	21.6
4	51.9	18.5	29.6	14.8	100.0	33.3
5	25.5	7.8	10.8	7.8	<u>8.8</u>	100.0

8.8% of reports mentioning System Network Configuration Discovery also mention Service Execution (9/102)

Normalization 2: Deviations from Mean Popularity



Normalization 2: Deviations from Mean Popularity

The average percentage that Valid Accounts has been reported with any technique is 7%.

The average percentage that Remote File Copy has been reported with any technique is 28%.

				4.3	26.1	60.9
			15.2		35.1	43.2
Exploitation for Client Execution	15.2	6.1	100.0	21.2	3.0	45.5
Spearphishing Link	3.1	18.8	21.9	100.0	31.2	34.4
Valid Accounts	14.0	30.2	2.3	23.3	100.0	20.9
Remote File Copy	7.7	8.7	8.2	6.0	4.9	100.0
	Network Service Scanning	Remote Desktop Protocol	Exploitation for Client Execution	Spearphishing Link	Valid Accounts	Remote File Copy

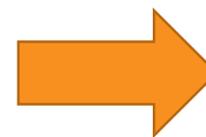
The percentage of reports with Spearphishing Link that also had Remote File Copy was 0.3 standard deviations greater than the average percentage for Remote File Copy

For Spearphishing Link and Valid Accounts, this number was **2.2!**

Hypothesis: the co-occurrence between Spearphishing Link and Valid Accounts is due to something inherent about the techniques; not popularity of one on its own

Normalization 2: Deviations from Mean Popularity

Network Service Scanning	100.0	30.4	21.7	4.3	26.1	60.9
Remote Desktop Protocol	18.9	100.0	5.4	16.2	35.1	43.2
Exploitation for Client Execution	15.2	6.1	100.0	21.2	3.0	45.5
Spearphishing Link	3.1	18.8	21.9	100.0	31.2	34.4
Valid Accounts	14.0	30.2	2.3	23.3	100.0	20.9
Remote File Copy	7.7	8.7	8.2	6.0	4.9	100.0
	Network Service Scanning	Remote Desktop Protocol	Exploitation for Client Execution	Spearphishing Link	Valid Accounts	Remote File Copy



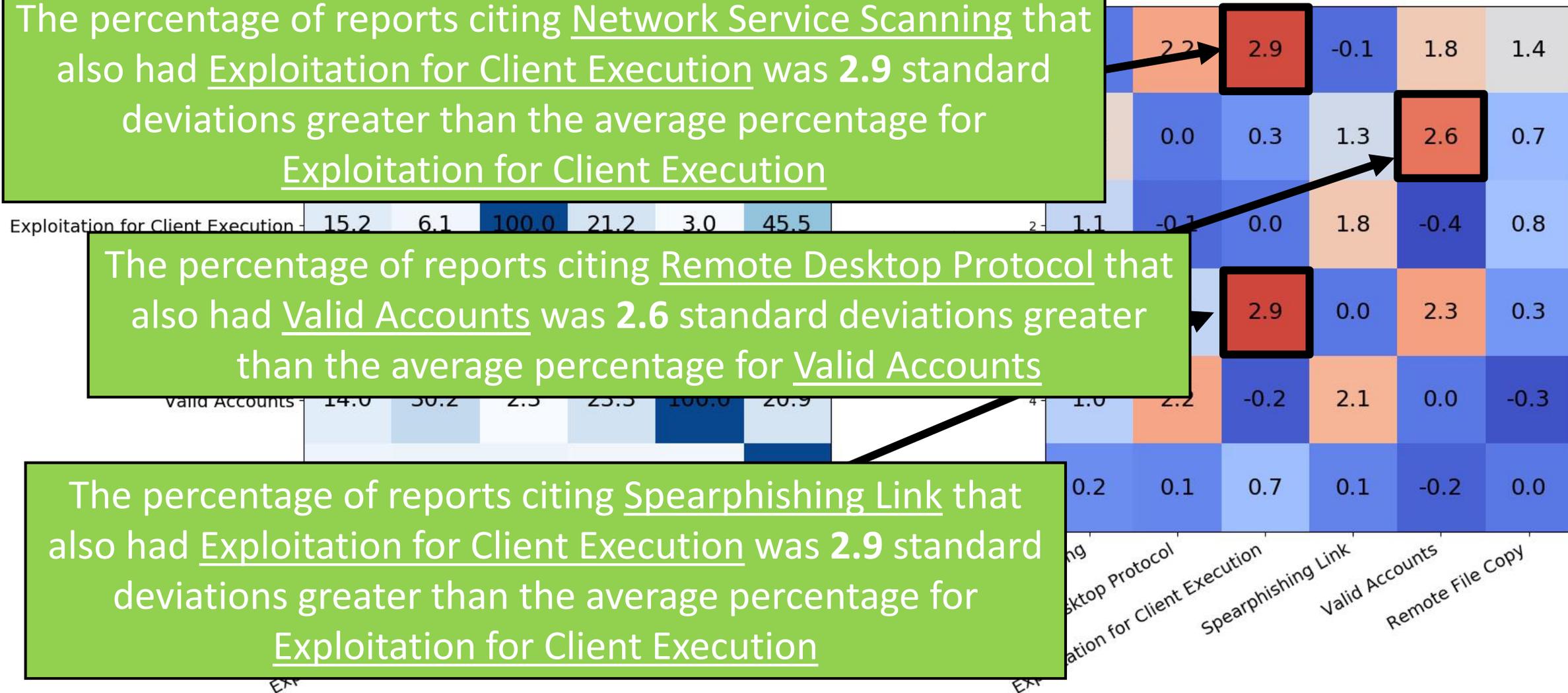
0	0.0	2.2	2.9	-0.1	1.8	1.4
1	1.6	0.0	0.3	1.3	2.6	0.7
2	1.1	-0.1	0.0	1.8	-0.4	0.8
3	-0.4	1.1	2.9	0.0	2.3	0.3
4	1.0	2.2	-0.2	2.1	0.0	-0.3
5	0.2	0.1	0.7	0.1	-0.2	0.0
	Network Service Scanning	Remote Desktop Protocol	Exploitation for Client Execution	Spearphishing Link	Valid Accounts	Remote File Copy

Normalization 2: Deviations from Mean Popularity

The percentage of reports citing Network Service Scanning that also had Exploitation for Client Execution was 2.9 standard deviations greater than the average percentage for Exploitation for Client Execution

The percentage of reports citing Remote Desktop Protocol that also had Valid Accounts was 2.6 standard deviations greater than the average percentage for Valid Accounts

The percentage of reports citing Spearphishing Link that also had Exploitation for Client Execution was 2.9 standard deviations greater than the average percentage for Exploitation for Client Execution



Select Associated Pairs (Shared References >10)

Technique 1	Technique 2	Score	Type*
Video Capture	Audio Capture	6.78	Implementation
Standard Non-Application Layer Protocol	Custom Command and Control Protocol	5.52	Implementation
User Execution	Spearphishing Attachment	5.06	Dependence
Permission Groups Discovery	Account Discovery	4.92	Implementation
Exploitation for Client Execution	Spearphishing Attachment	3.94	Dependence
Remote System Discovery	Windows Admin Shares	3.27	Dependence
Data from Removable Media	File and Directory Discovery	3.11	Dependence
Shortcut Modification	Registry Run Keys/Start Folder	3.08	Alternative
Query Registry	Modify Registry	2.99	Implementation
Exfiltration Over Command and Control Channel	Process Discovery	2.45	?
Peripheral Device Discovery	Input Capture	2.36	?

*: inferred relationship type

Summary: Why This Matters

- **Correlating techniques can be used across use cases for prioritization**
 - Using ATT&CK: low overhead; we've done the parsing work for you
 - Using your own threat model: can customize to your own intel
- **For the future: grow methodology to include more rigorous analysis**

- **Still – several shortcomings:**
 - Still have to consider bias from reporting + classification
 - Lots of discovery techniques have high co-incidence scores!
 - Relationship type between techniques needs to be inferred
 - No notion of sequencing...

Finding Related Techniques

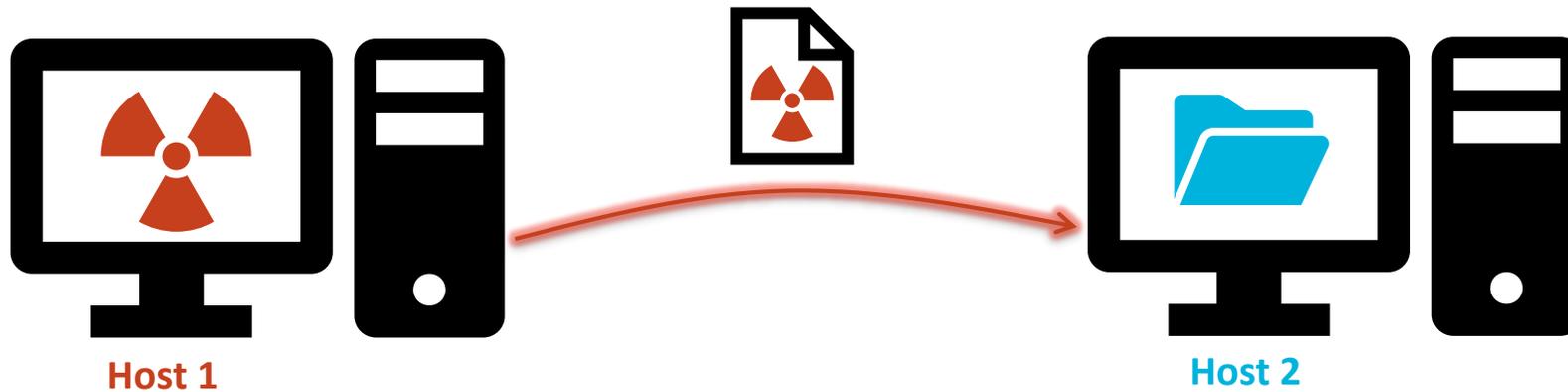
Semantic Analysis

Semantic Analysis: Motivation

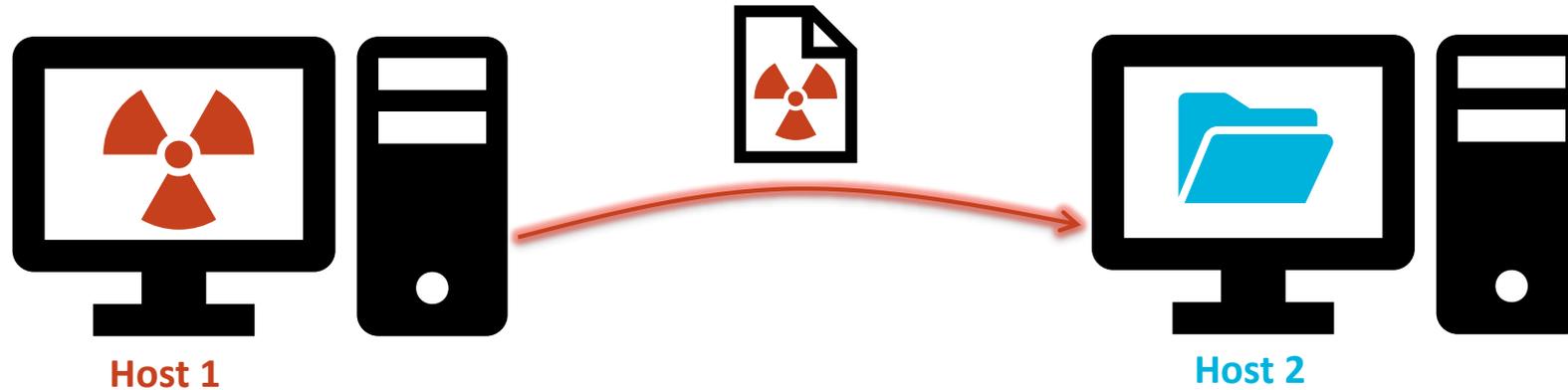
- **Analyzing threat reports gives us information about technique relationships**
- **However, the methodology:**
 - Needs to be built off of a large corpus of already analyzed threat reports;
 - Is subject to reporting + ingestion bias;
 - Does not provide information about relationship type; and
 - More often than not lacks intuitively-explainable results.
- **Is there a better way?**

Example Scenario: Remote File Copy

Suppose I'm an adversary... How would I execute Remote File Copy?



Example Scenario: Remote File Copy



■ What needs to be true for me to copy a file from Host 1 to Host 2?

- Code execution and file containing a RAT on Host 1
- Mounted file share from Host 2 on Host 1
- Write access to file share

} Requirements, or *preconditions*

■ What will be true after copying the file?

- There will be a new file on Host 2
- That file will contain the RAT

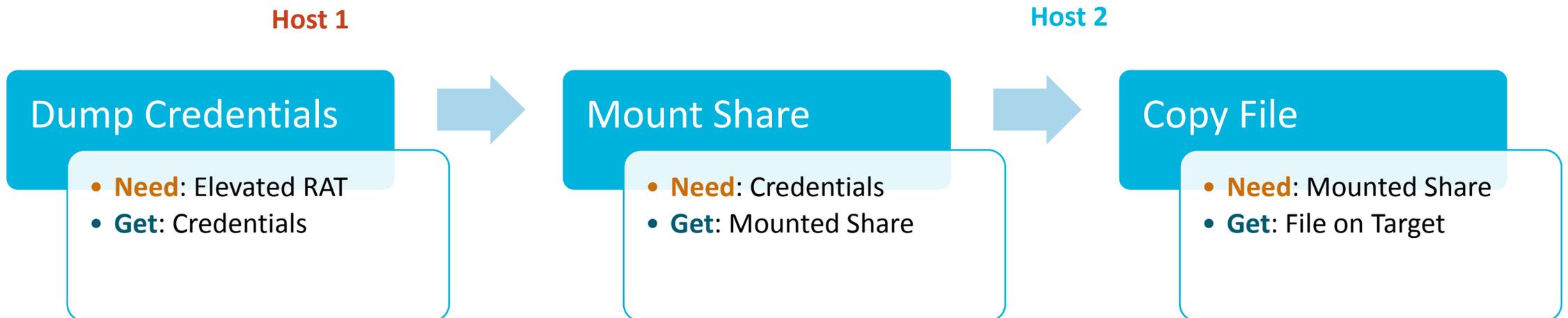
} Consequences, or *postconditions*

Creating a Technique Chain with Remote File Copy

Leveraging pre and postconditions allows us to construct technique chains!

Using these chains, we can identify technique relationships:

- Remote File Copy depends on Windows Admin Shares
- Windows Admin Shares depends on Credential Dumping



Semantic Analysis: The Idea

- **By logically modeling techniques with:**
 - The requirements to execute each technique and
 - The consequences of executing each technique

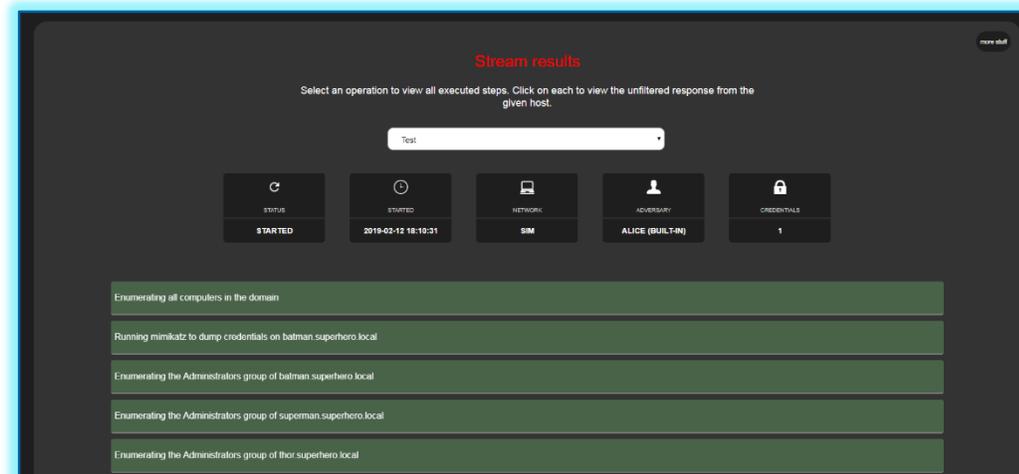
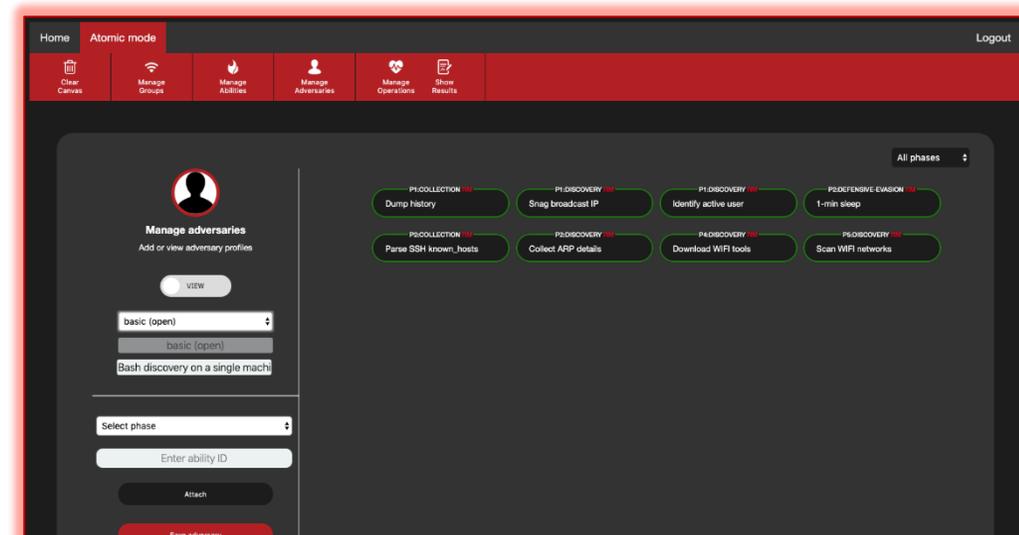
...we can easily identify how techniques chain together

- **Using this information, we can:**
 - Identify technique dependencies, architecting our defenses to block “critical” techniques that enable many others
 - Identify technique alternatives, creating detection rules that work with high fidelity

Where can we get a semantic model?

Enter: Automated Adversary Emulation with CALDERA

- **Software built to act like a realistic adversary**
 - Built around ATT&CK as the threat model
 - Internal model with adversary actions that uses AI to make decisions during operations
 - Highly configurable, easy to mix-and-match new adversary capabilities/change behavior
- **Features:**
 - Low install overhead – can run on a laptop
 - Modular plugin architecture
- **Two main modes: fully automated and scripted**
- **In fully automated mode, CALDERA needs to make *intelligent* decisions to advance its operation**
 - Behind-the-scenes: pre and postconditions!



Leveraging Actions in CALDERA's Adversary Mode

- **33 implemented actions, each with**
 - A name + ATT&CK mapping
 - A set of object requirements
 - A set of object consequences
- **Idea: connect actions to objects**
 - Link objects to actions they enable
 - Link actions to objects they change
- **Object-oriented logic**
 - Statements: object + property
- **Disclaimers**
 - Bugs/omissions in logic create loss
 - CALDERA logic is unintuitive

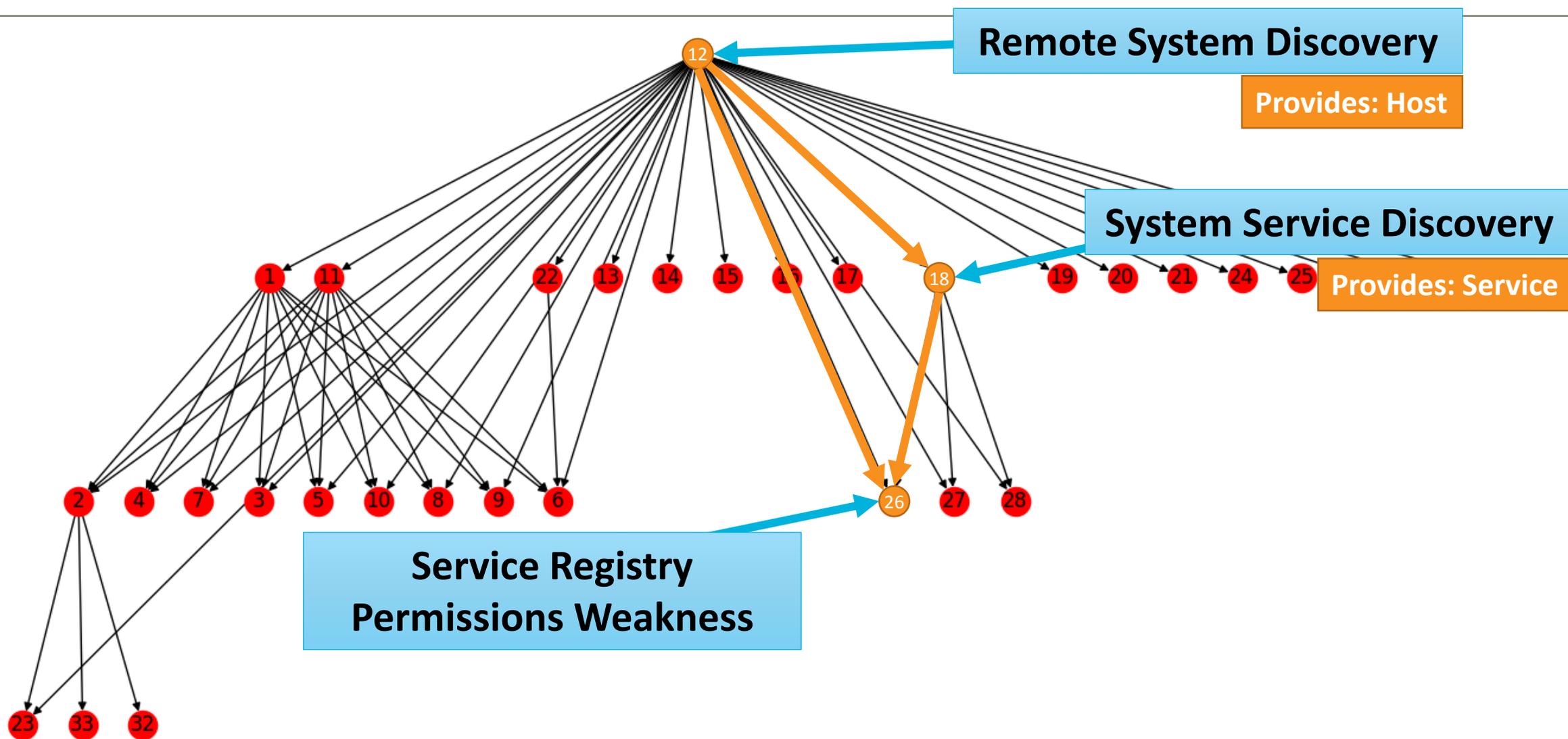
```
display_name = "copy_file"
summary = "Copy a file from a computer to another using a mounted network share"
preconditions = [("rat", OPRat),
                 ("share", OPShare({"src_host": OPVar("rat.host")}) )]
postconditions = [("file_g", OPFile({'host': OPVar("share.dest_host")}) )]
preproperties = ['rat.executable', 'share.share_path']
postproperties = ['file_g.path']
```

A First Look: Mandatory Dependencies

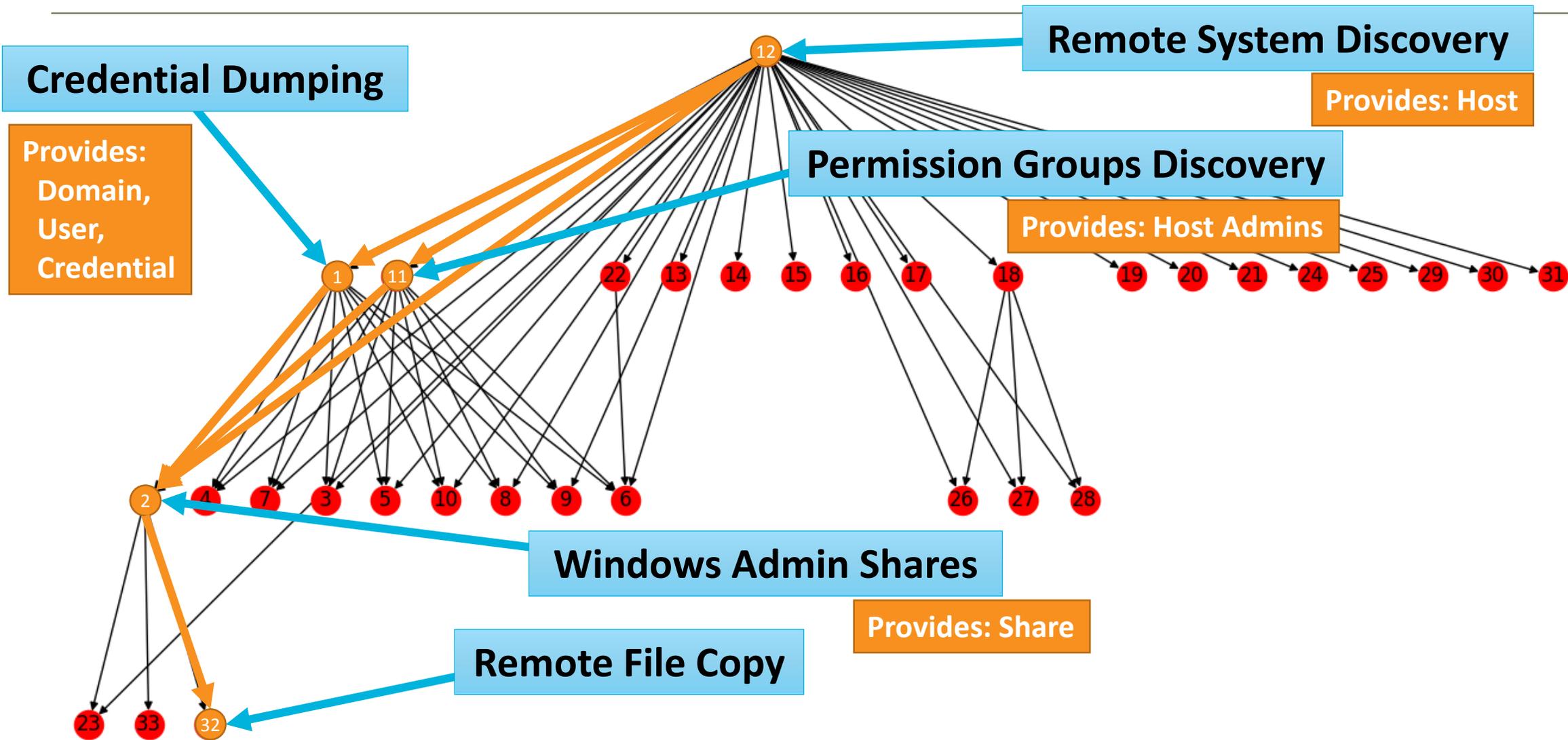
- **Observations:**
 1. All techniques require some objects for execution
 2. Many techniques discover/create new objects
 3. Some objects can be discovered/created by only 1 technique

- **Idea: identify those techniques which are *mandatory* – i.e., no alternatives exist – for specific objects**

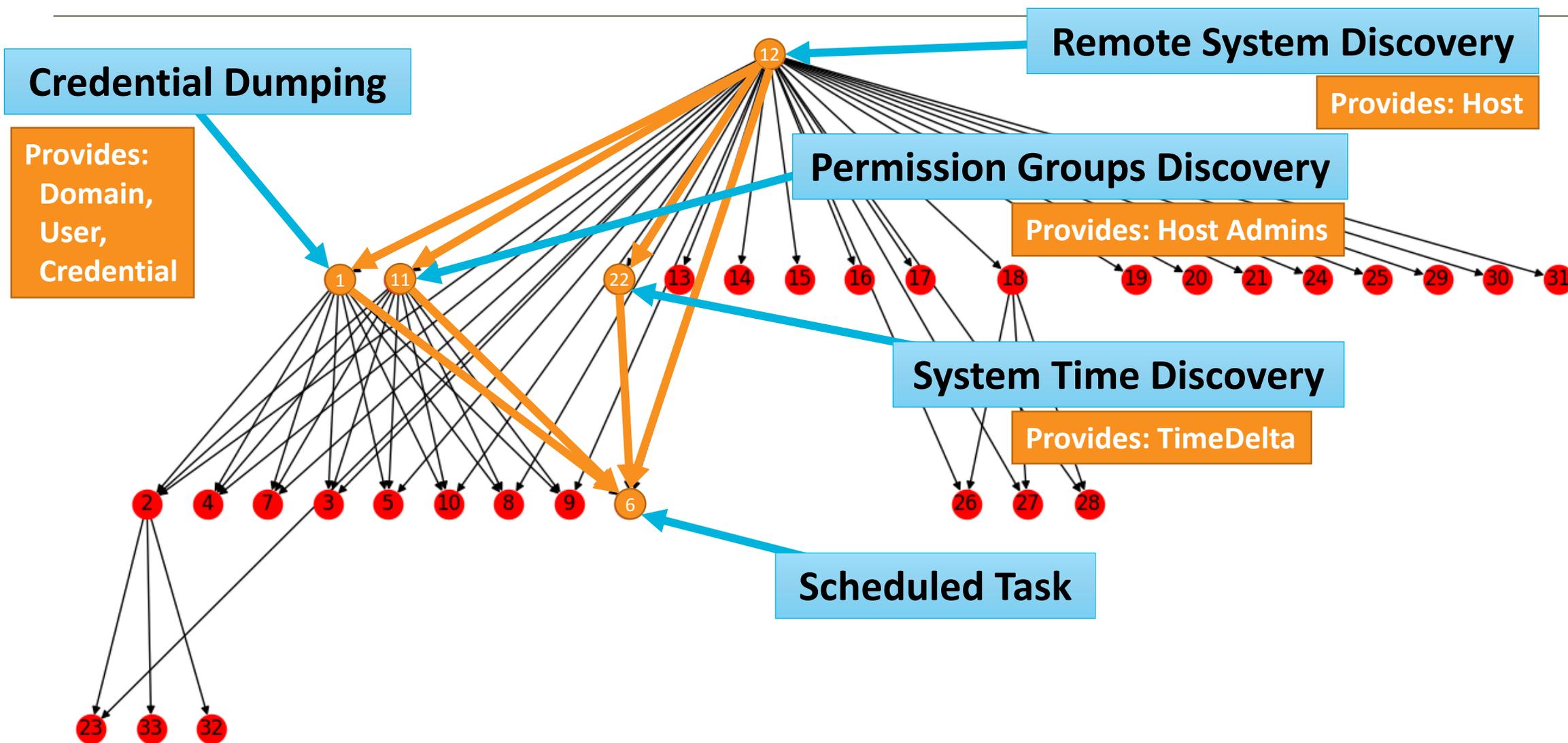
Mandatory Dependencies in CALDERA's Logic



Mandatory Dependencies in CALDERA's Logic



Mandatory Dependencies in CALDERA's Logic



Mandatory Dependencies: By the Numbers

Action Name	ATT&CK Technique	Critical Object	# Dependent Actions
get_creds	Credential Dumping	Credential	9
get_admin	Permission Groups Discovery	Host.admins	9
get_computers	Remote System Discovery	Host	30
priv_esc(service)	System Service Discovery	Service	3
net_time	System Time Discovery	TimeDelta	1
net_use	Windows Admin Shares	Share	3

- **Most actions have dependencies that can be met by multiple techniques**
- **Focusing on techniques that are the *only* one to satisfy dependencies can help us optimize our defenses**
- **(note: CALDERA nuances result in Remote System Discovery being mandatory)**

Technique Set Enhancement

- **Given a set of techniques, can we determine:**
 - If that set is self-contained
 - And if not, what techniques could we add to it to make it so?
- **Useful for filling gaps during hunting**
- **Example:**

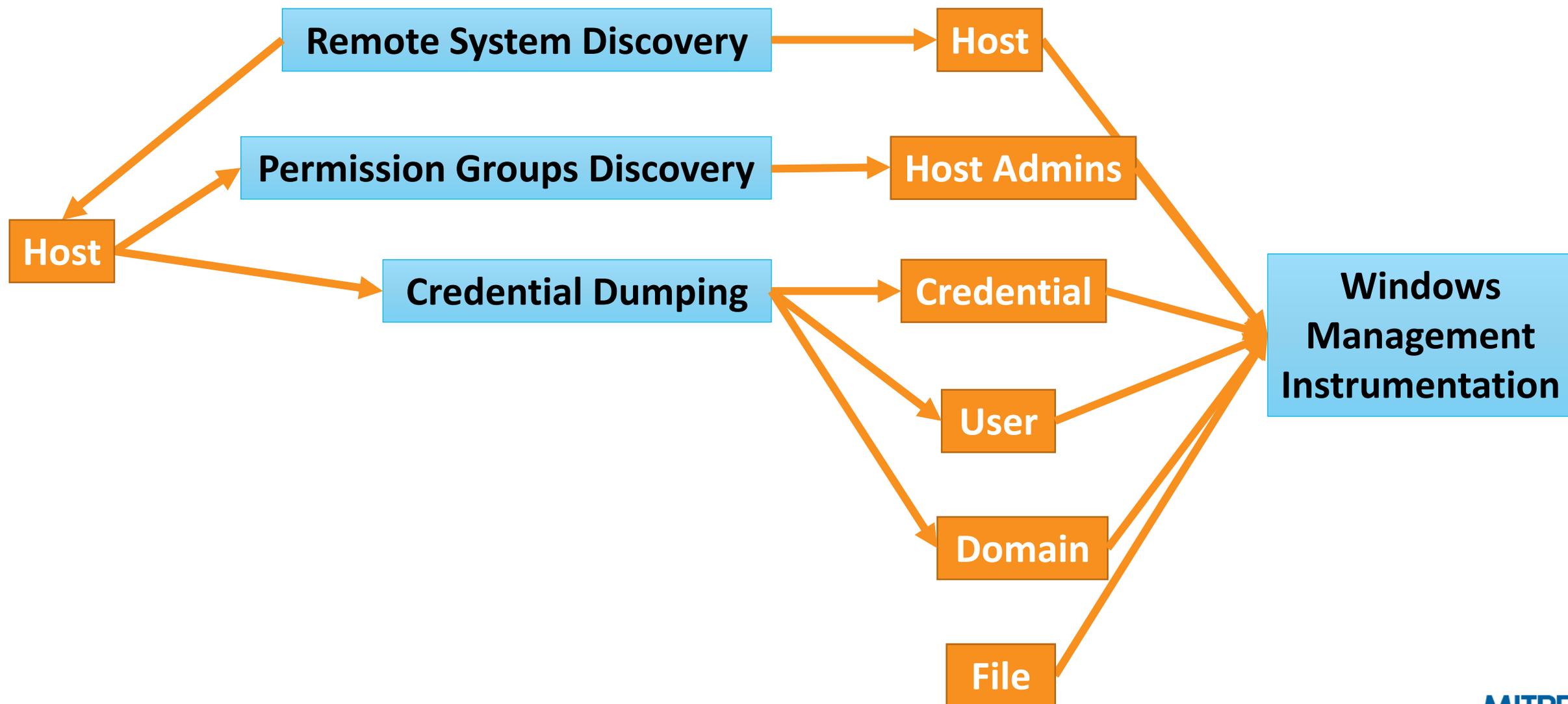
Permission Groups Discovery

Remote System Discovery

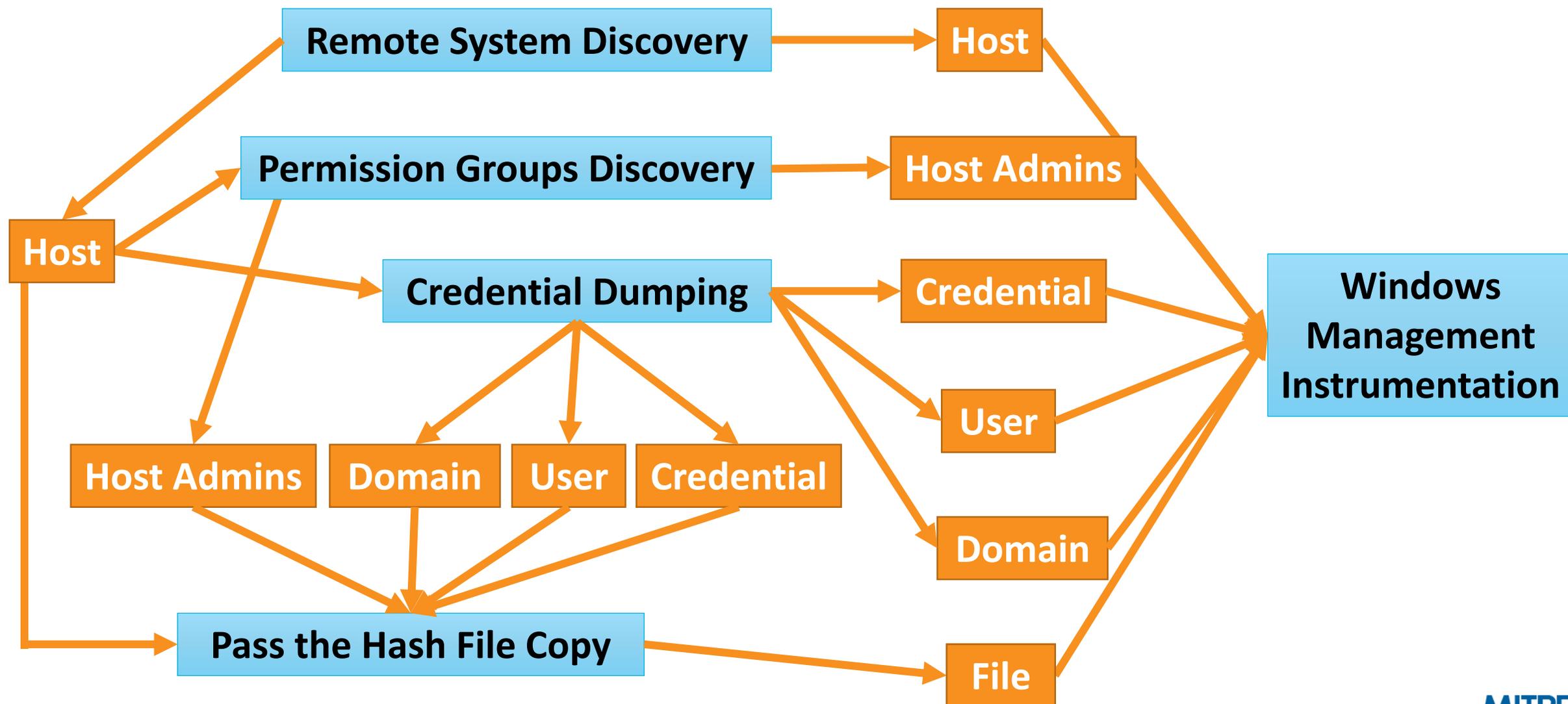
**Windows
Management
Instrumentation**

Credential Dumping

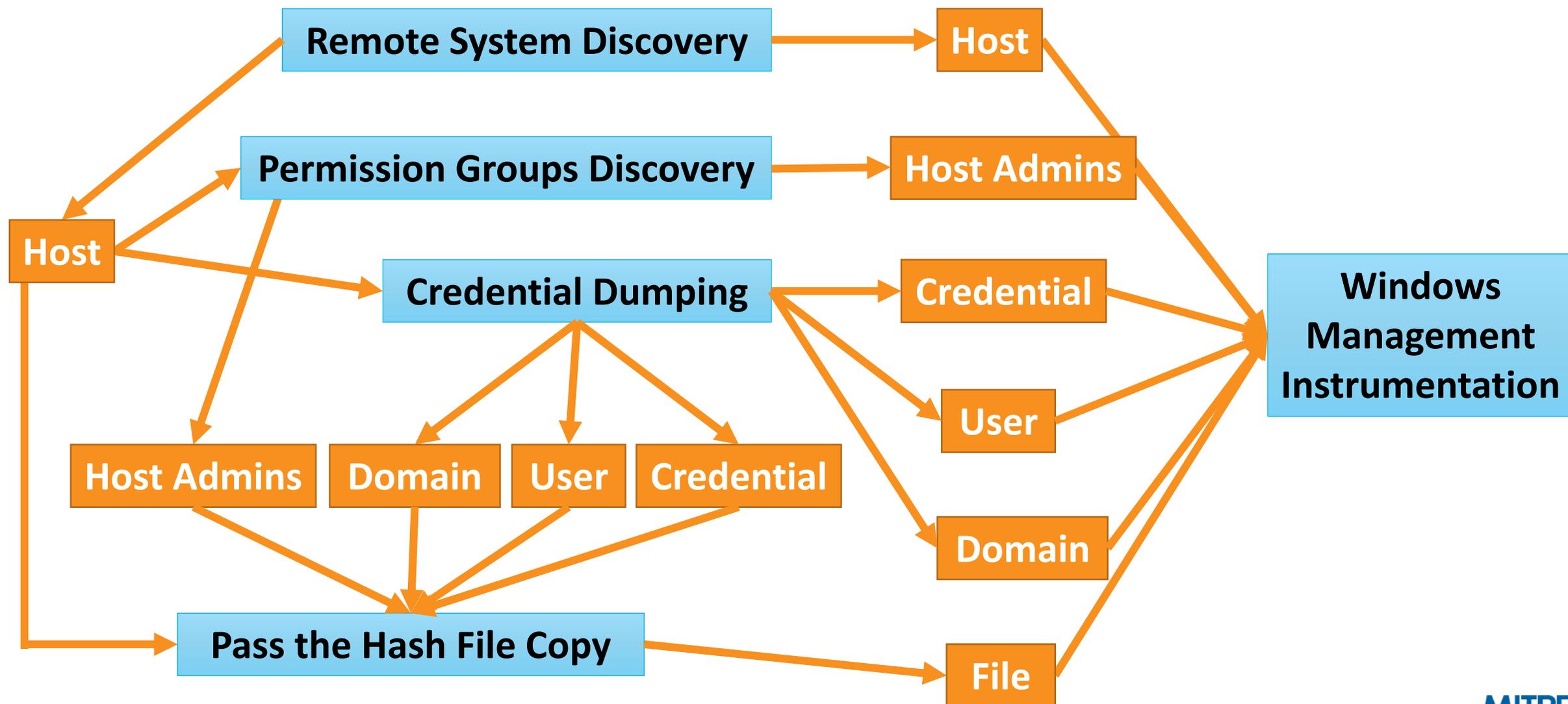
Thinking Backwards: Backsolving the Graph



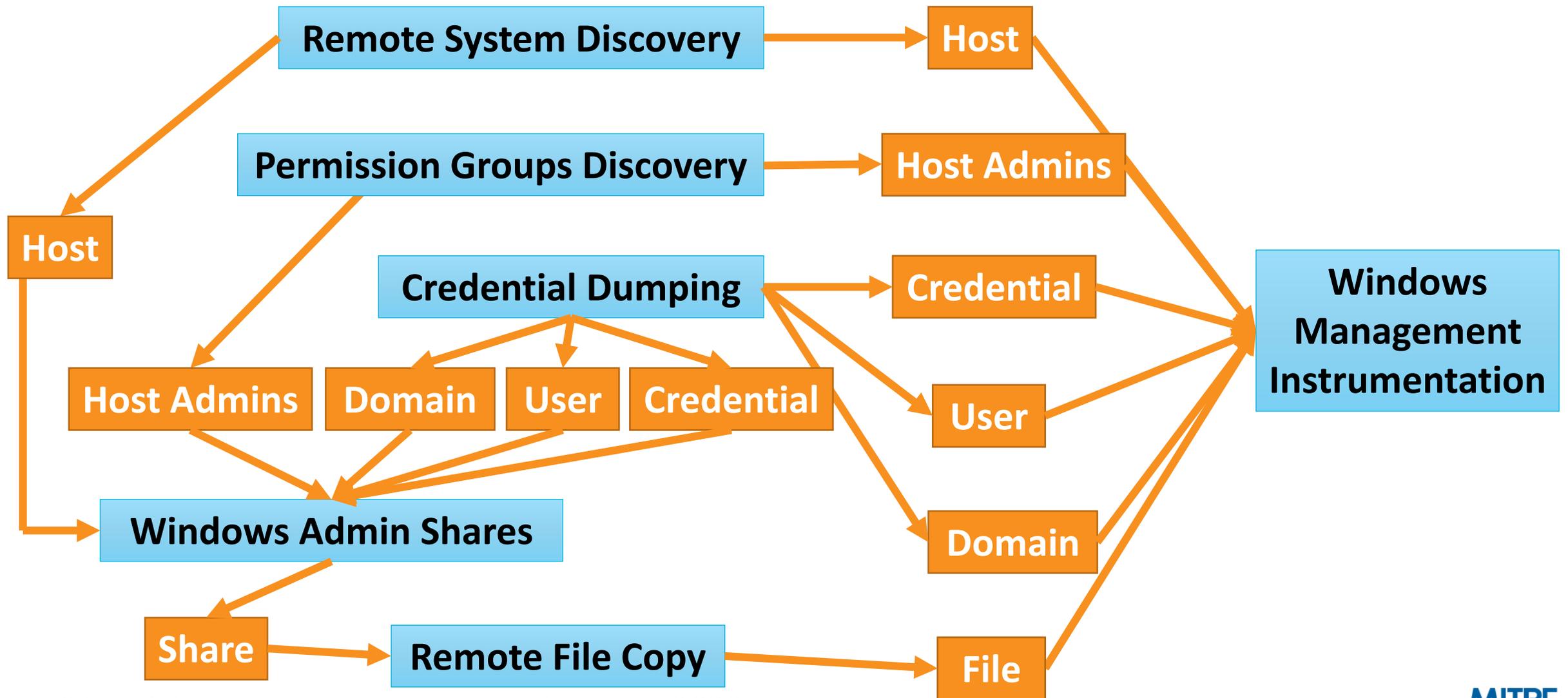
Thinking Backwards: Backsolving the Graph



Thinking Backwards: Backsolving the Graph



Thinking Backwards: Backsolving the Graph



Using Set Enhancement

- **Start with one technique: build out all sets that self-contain that technique**
 - Use beforehand for security engineering or detection
- **Start with a set of techniques: build out**
 - Use live for threat hunting
- **Start with one technique:**
 - Build out all sets for that technique
 - Remove the technique from all sets
 - Rebuild-out and see what's new
 - Great for alternatives

Action	# Plans	Longest	Shortest
Exfiltrate	13	6	3
WinRM	10	6	5
Remove Share	10	7	6
Scheduled Task Lateral Move	10	7	6
Remote Process (WMI)	10	6	5
Pass the Hash SC	10	6	5
Timestomp	8	6	4
SC Persist	8	6	4
Xcopy File	2	5	5

Technique Sequence Enumeration

- **Given an adversary profile, can we figure out the ways in which the adversary's actions might be actuated?**

Permission Groups Discovery

**Windows Management
Instrumentation**

Windows Admin Shares

Credential Dumping

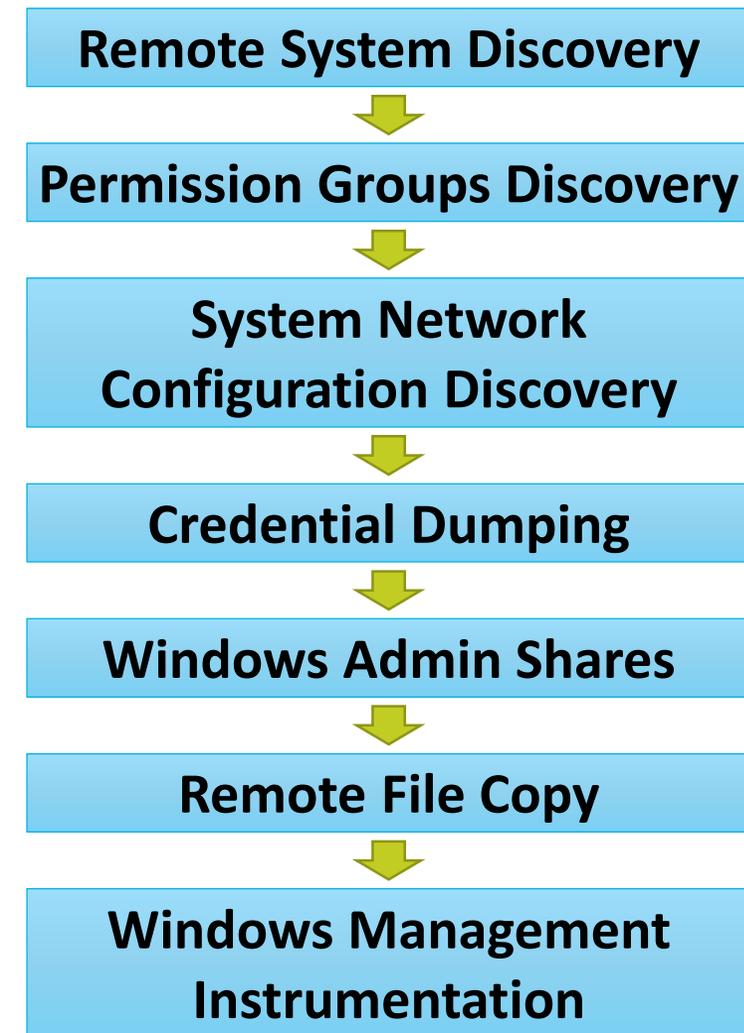
Remote System Discovery

Remote File Copy

**System Network
Configuration Discovery**

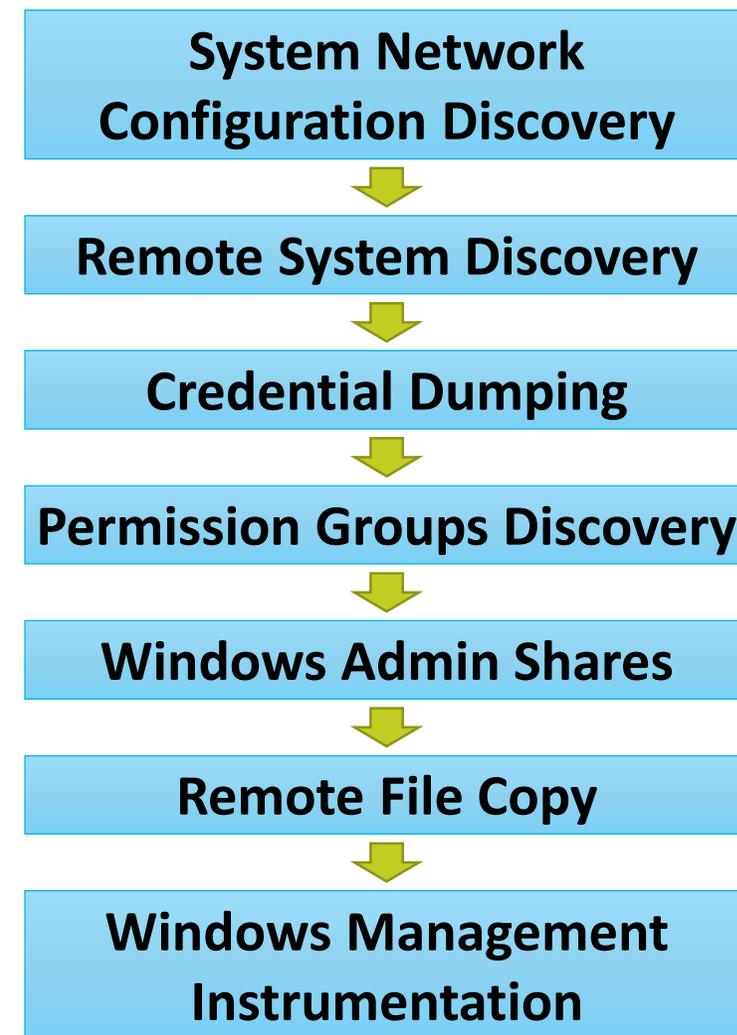
Technique Sequence Enumeration

- **Given an adversary profile, can we figure out the ways in which the adversary's actions might be actuated?**
- **Yes! Leverage pre and postconditions to construct technique sequences**
- **Sequence 1:**
 - Remote System Discovery (provides “Host”)
 - Permissions Groups (provides “Host Admins”)
 - Network Configuration (provides “Domain”)
 - Credential Dumping (provides “Credential”)
 - Windows Admin Shares, Remote File Copy, and Windows Management Instrumentation last



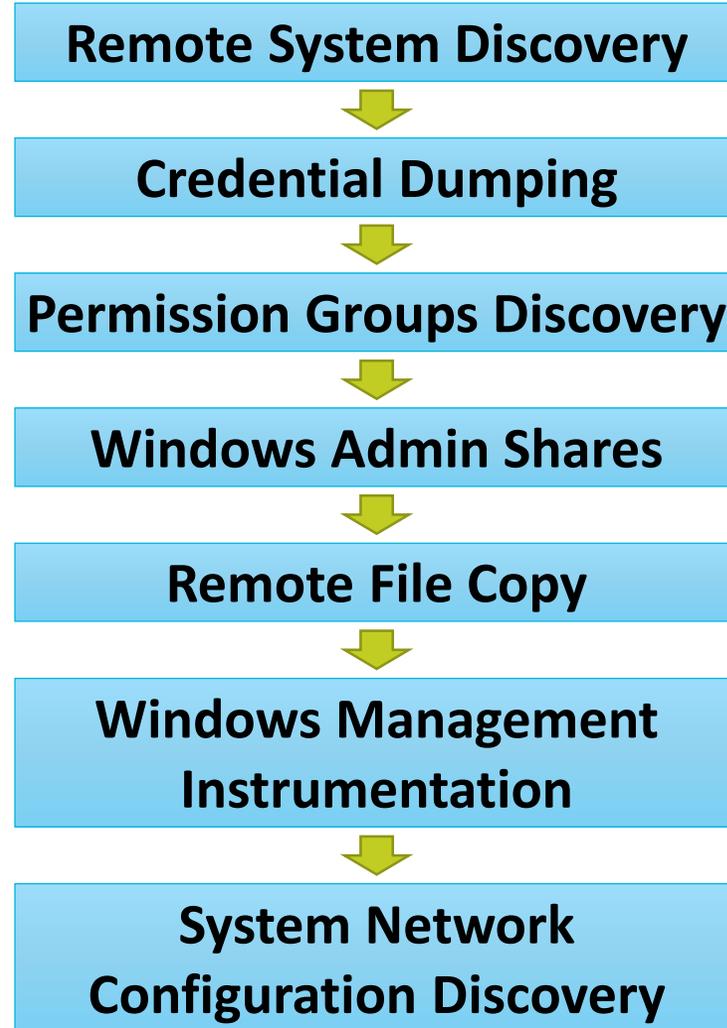
Technique Sequence Enumeration (2)

- **Given an adversary profile, can we figure out the ways in which the adversary's actions might be actuated?**
- **Yes! Leverage pre and postconditions to construct technique sequences**
- **Sequence 2:**
 - Network Configuration (provides “Domain”)
 - Remote System Discovery (provides “Host”)
 - Credential Dumping (provides “Credential”)
 - Permissions Groups (provides “Host Admins”)
 - Windows Admin Shares, Remote File Copy, and Windows Management Instrumentation last



Technique Sequence Enumeration (3)

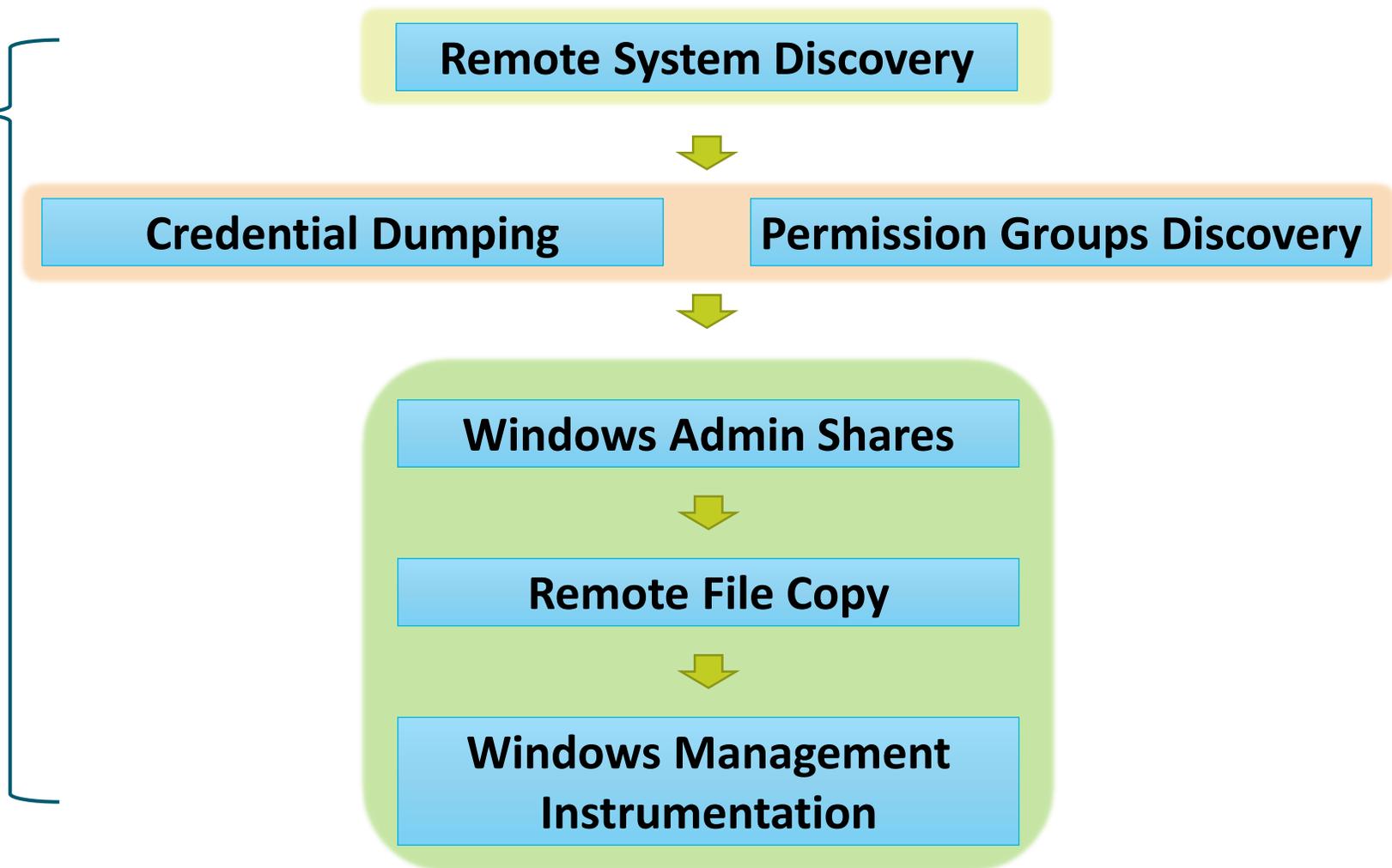
- **Given an adversary profile, can we figure out the ways in which the adversary's actions might be actuated?**
- **Yes! Leverage pre and postconditions to construct technique sequences**
- **Sequence 3:**
 - Remote System Discovery (provides “Host”)
 - Credential Dumping (provides “Credential”)
 - Permissions Groups (provides “Host Admins”)
 - Windows Admin Shares, Remote File Copy, and Windows Management Instrumentation
 - Network Configuration (provides “Domain”)



Technique Sequence Enumeration: Creating Flowcharts

System Network Configuration Discovery

- Remote System Discovery must be first
- Credential Dumping + Permissions Groups executed in same block
- Admin Shares, File Copy, WMI fall into a sequence
- Net Conf Discovery can be executed any time
- Total: $2 * 7 = 14$ sequences



Summary: Using Semantic Analysis

Mandatory Dependencies

- Ideal for finding critical dependencies
- Useful for initial hypothesis development + high-priority analytics
- Can inform security architecture (preventing critical dependencies)

Set Enhancement

- Shows dependencies as well as alternatives
- Best use case: at run-time, filling in gaps in hypothesis
- Can also be used for analytic correlation

Sequence Enumeration

- Shows dependencies, alternatives, and time-sequencing
- Best use case: developing high-fidelity analytics with correlation
- Can inform hunts + security architecture

Caution!

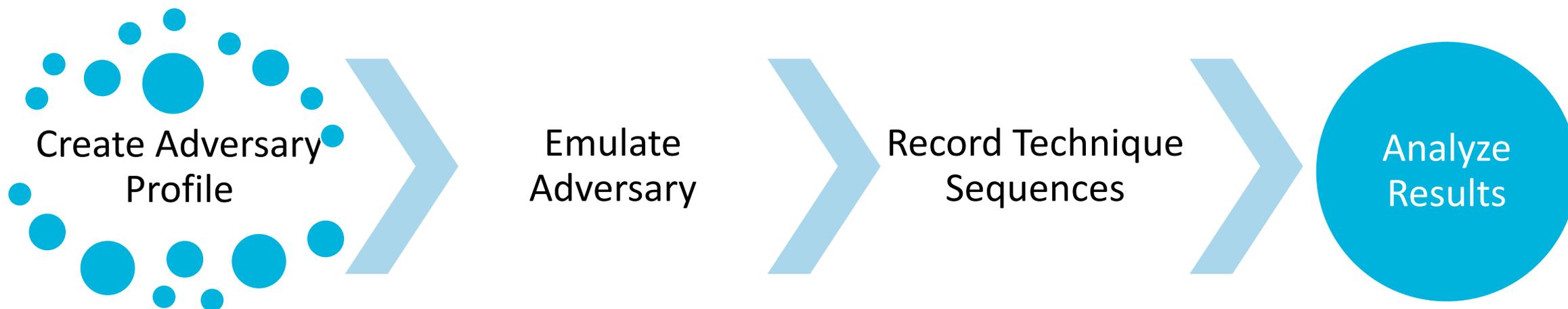
Semantic models are hard to make – and they're often incomplete

Finding Related Techniques

Experimental Results

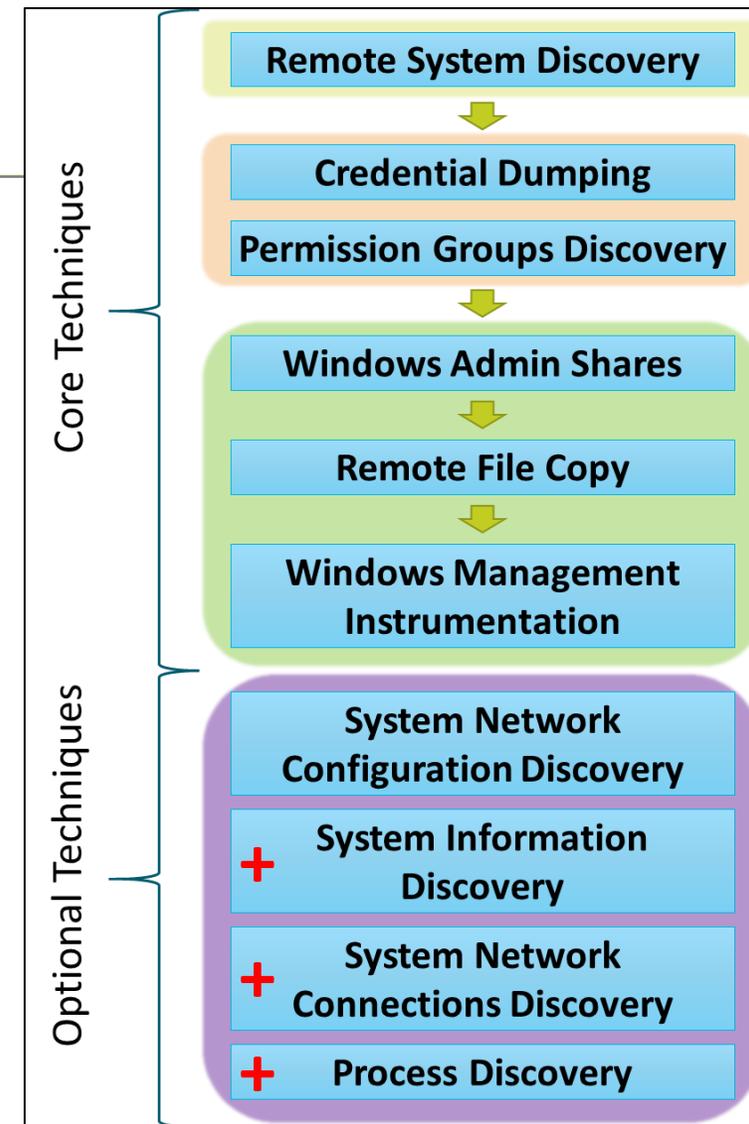
Idea

- **Both semantic modeling and threat report analysis have shortcomings**
 - Threat report analysis suffers from bias and descriptiveness
 - Semantic modeling requires an upfront time investment and can be lossy
- **What could we learn if we just simulate an adversary?**



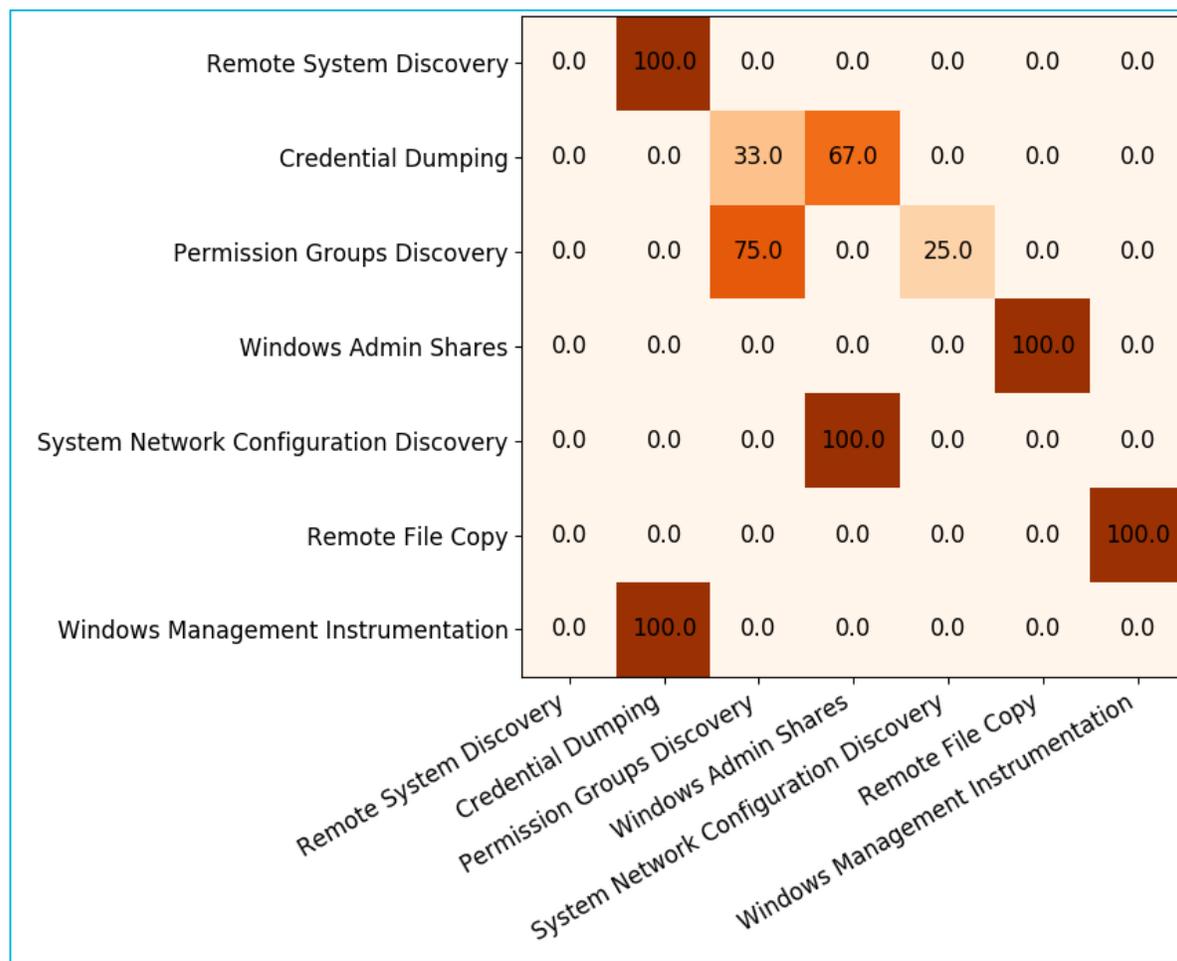
Experiment Design

- **Setup sample test network**
 - 4 Windows 10 workstations
 - 1 Domain Controller
 - 1 “admin” account seeded on start box
 - Enables easy lateral movement + TTP execution
- **Run CALDERA with 2 profiles:**
 - Alice (built-in): 6 key actions, 1 optional
 - Alice+: 6 key actions, 4 optional
- **Vary decision making capabilities**
 - Deterministic: using CALDERA’s scoring algorithm
 - Random: choosing actions randomly whenever execution possible



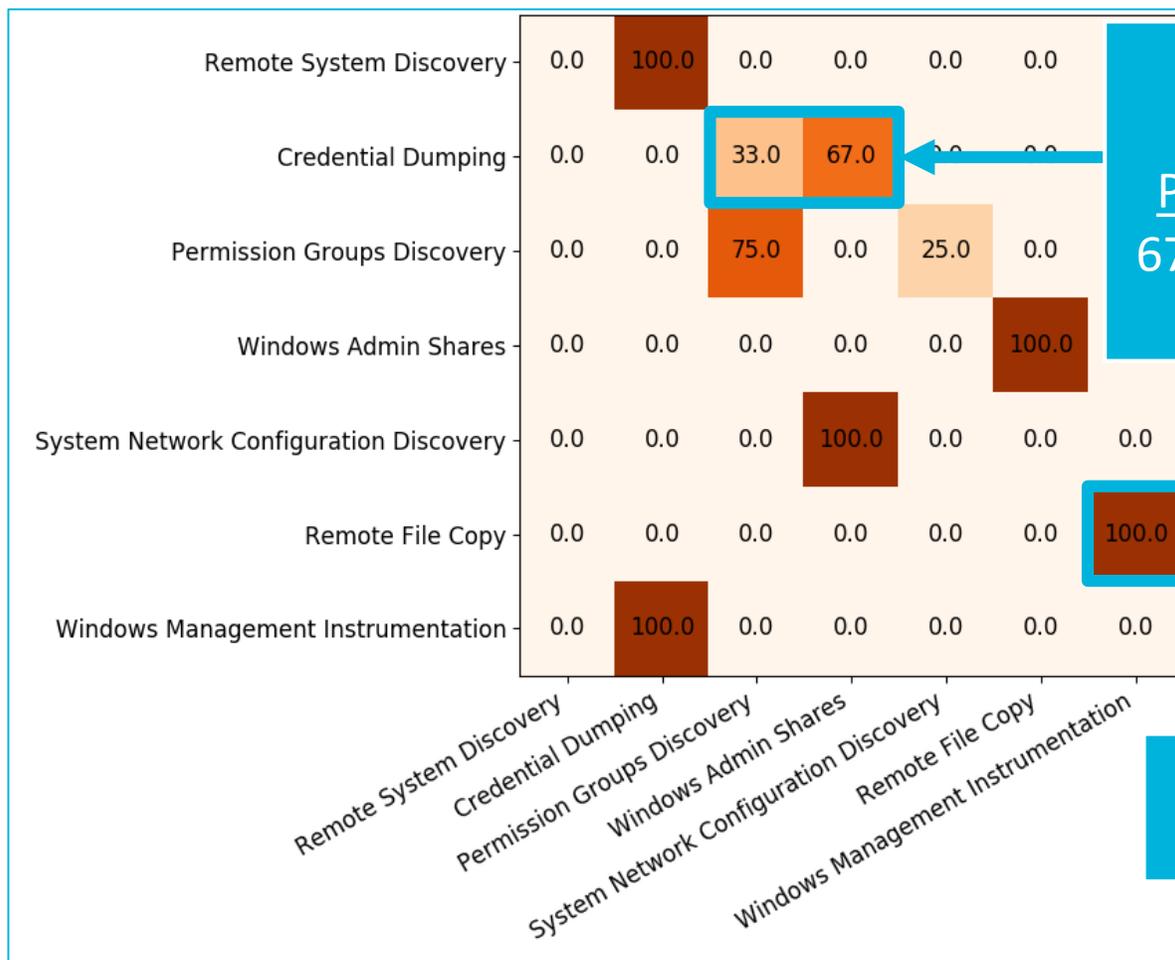
Alice With Determinism

How Often Technique A Followed Technique B



Alice With Determinism

How Often Technique A Followed Technique B

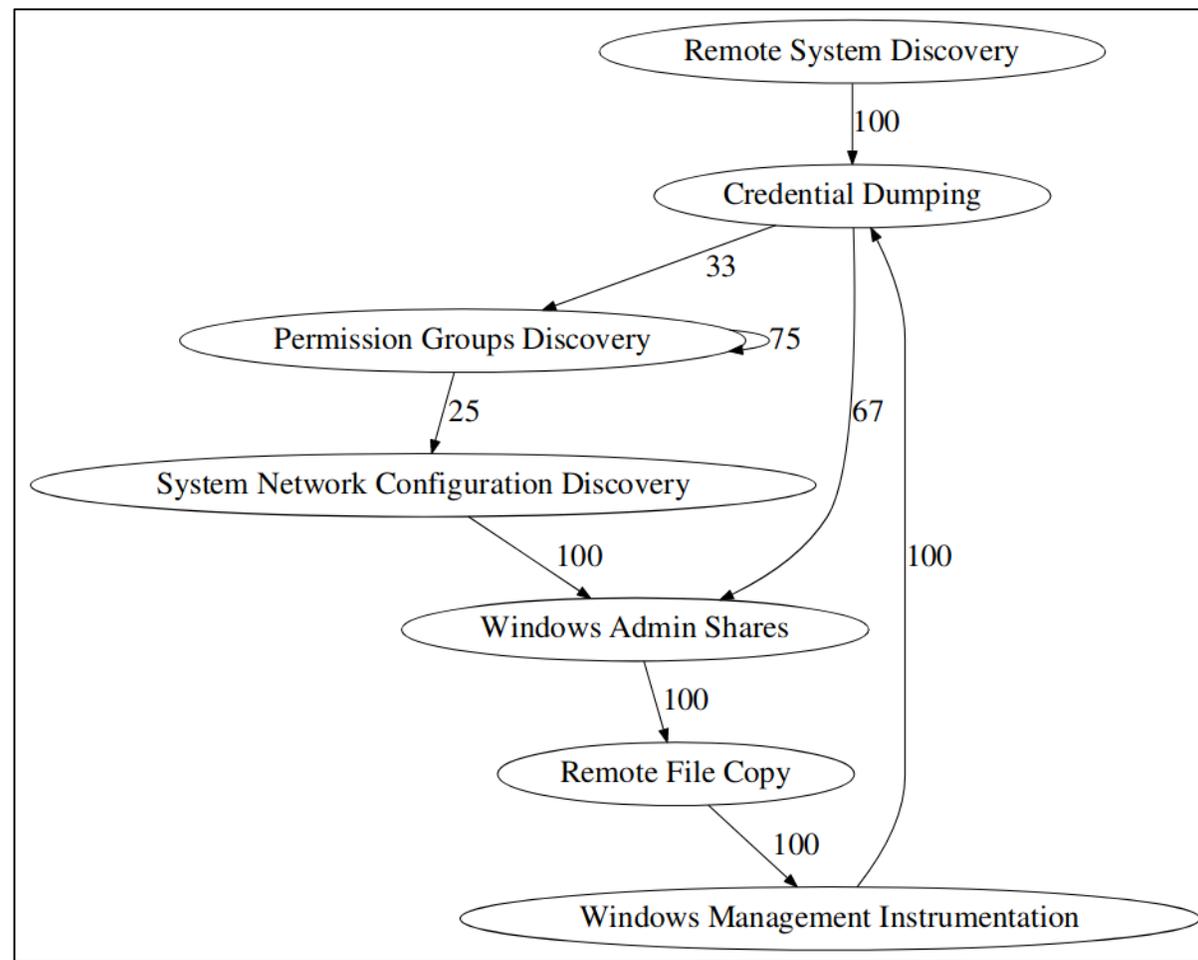
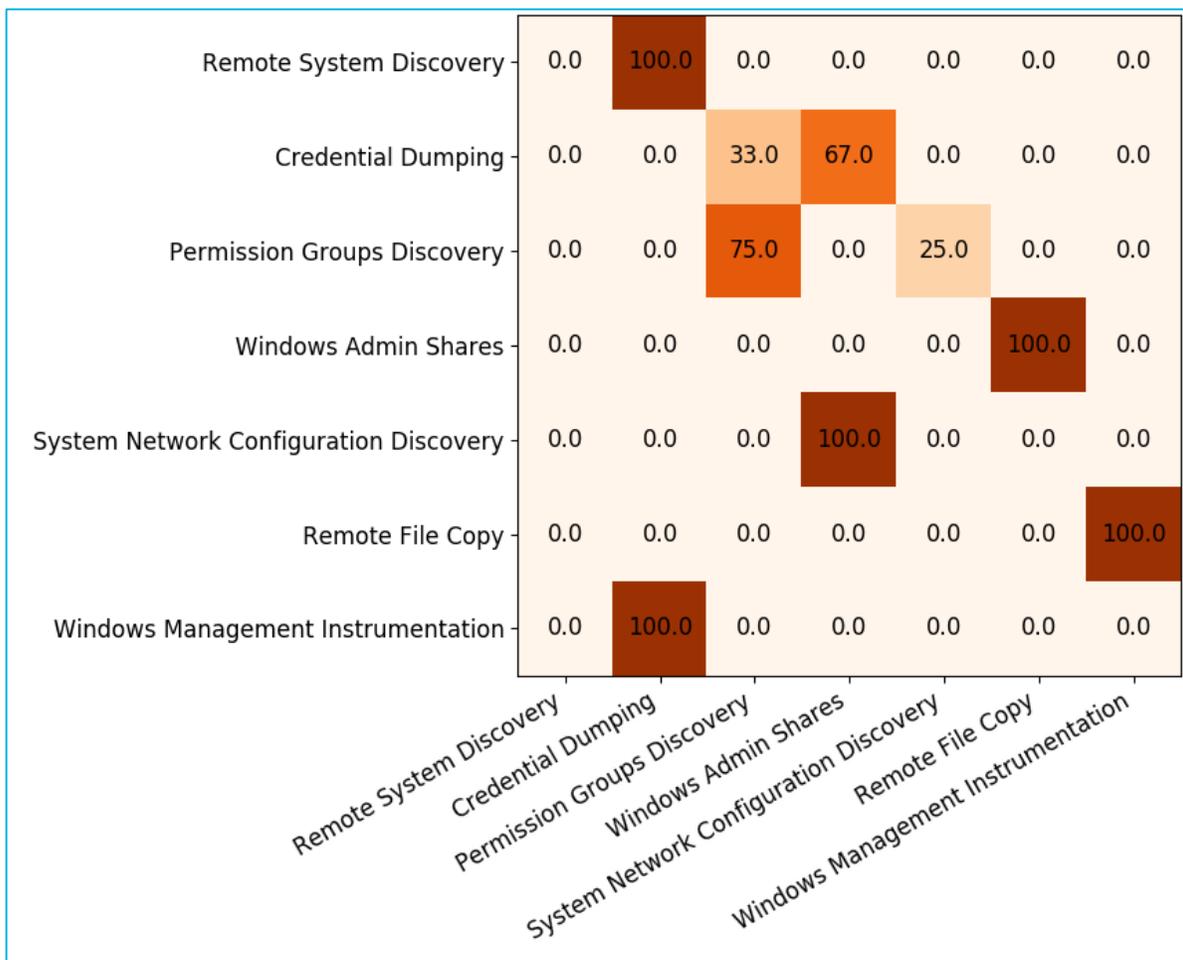


33% of the time Credential Dumping was followed by Permission Groups Discovery; 67% of the time it was followed by Windows Admin Shares

100% of the time Remote File Copy was followed by WMI

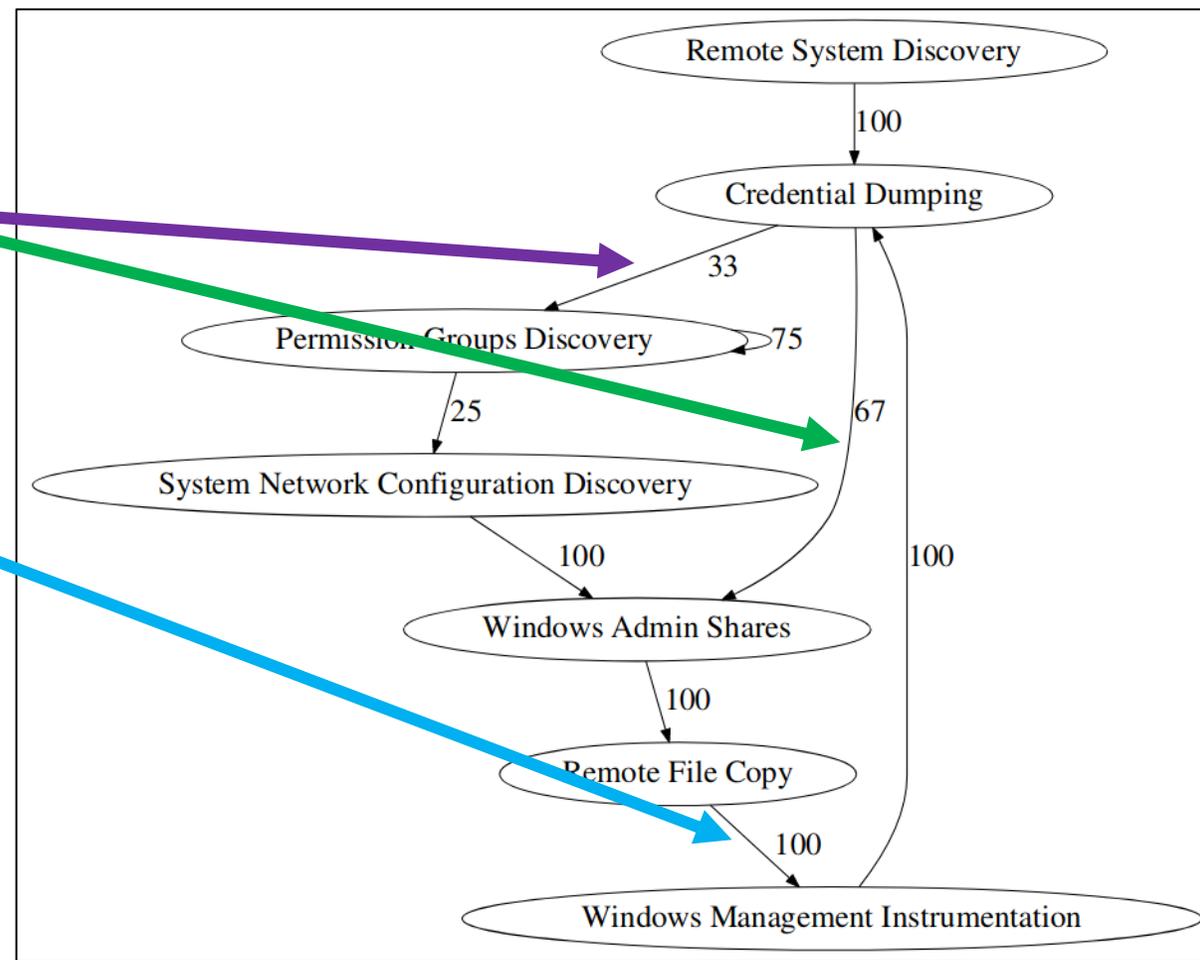
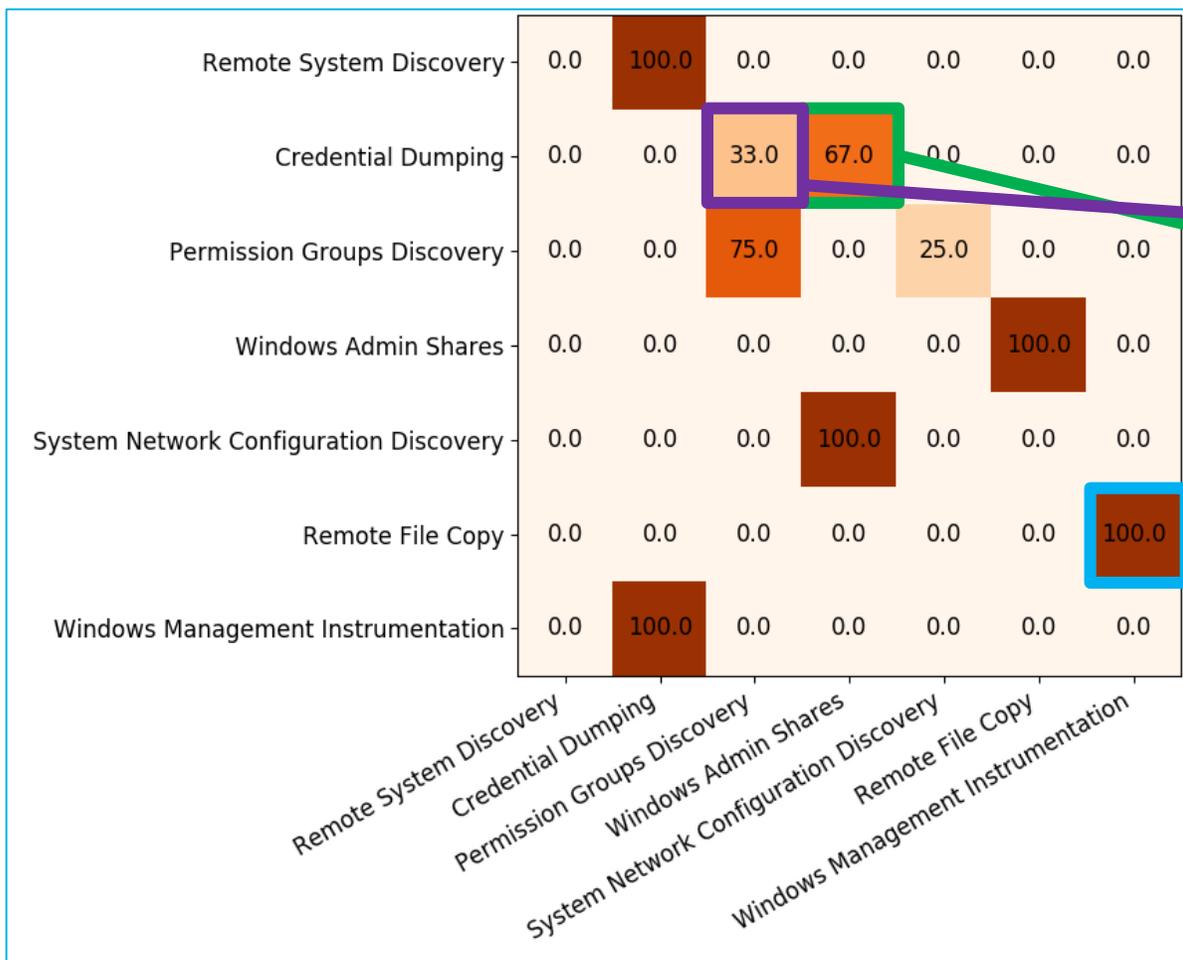
Alice With Determinism

How Often Technique A Followed Technique B



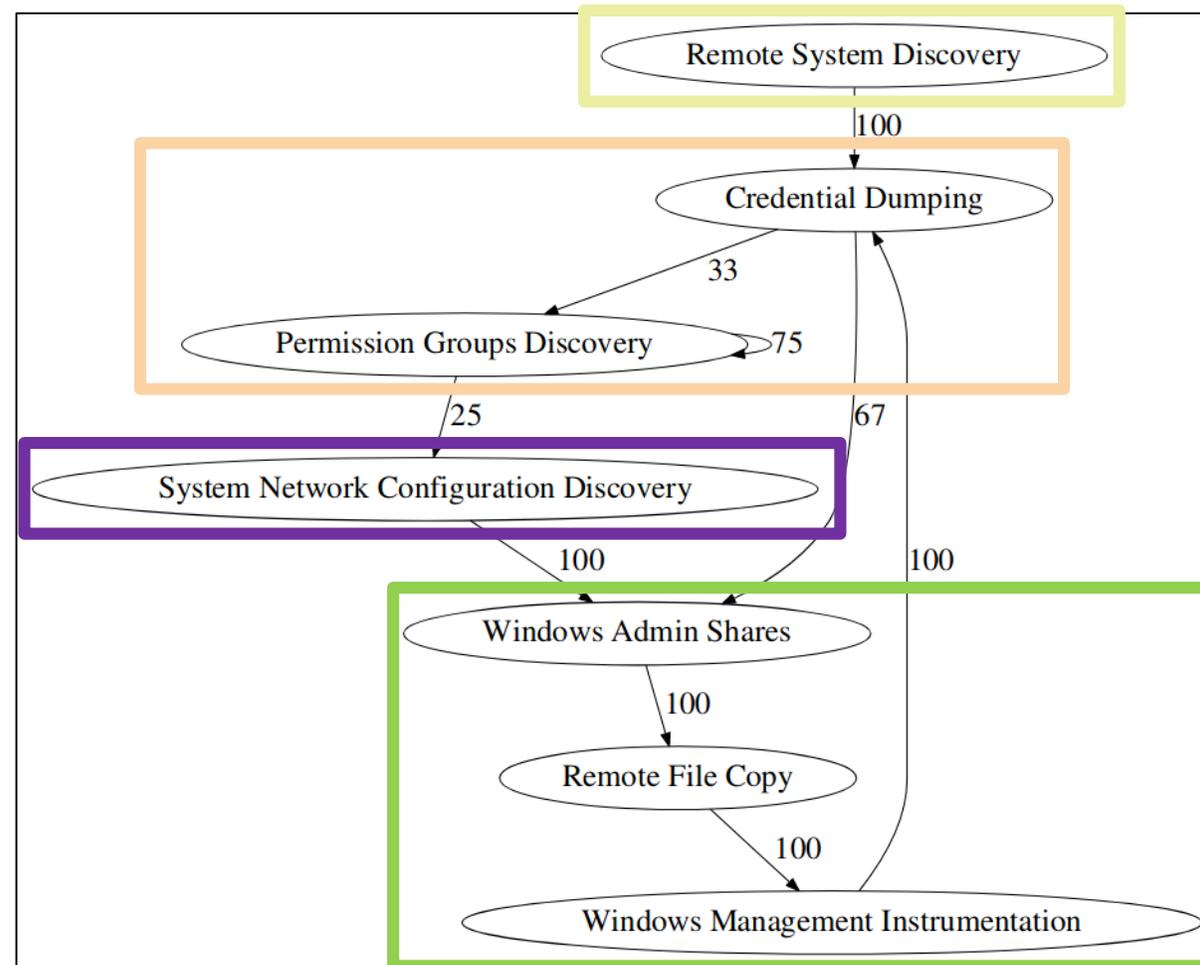
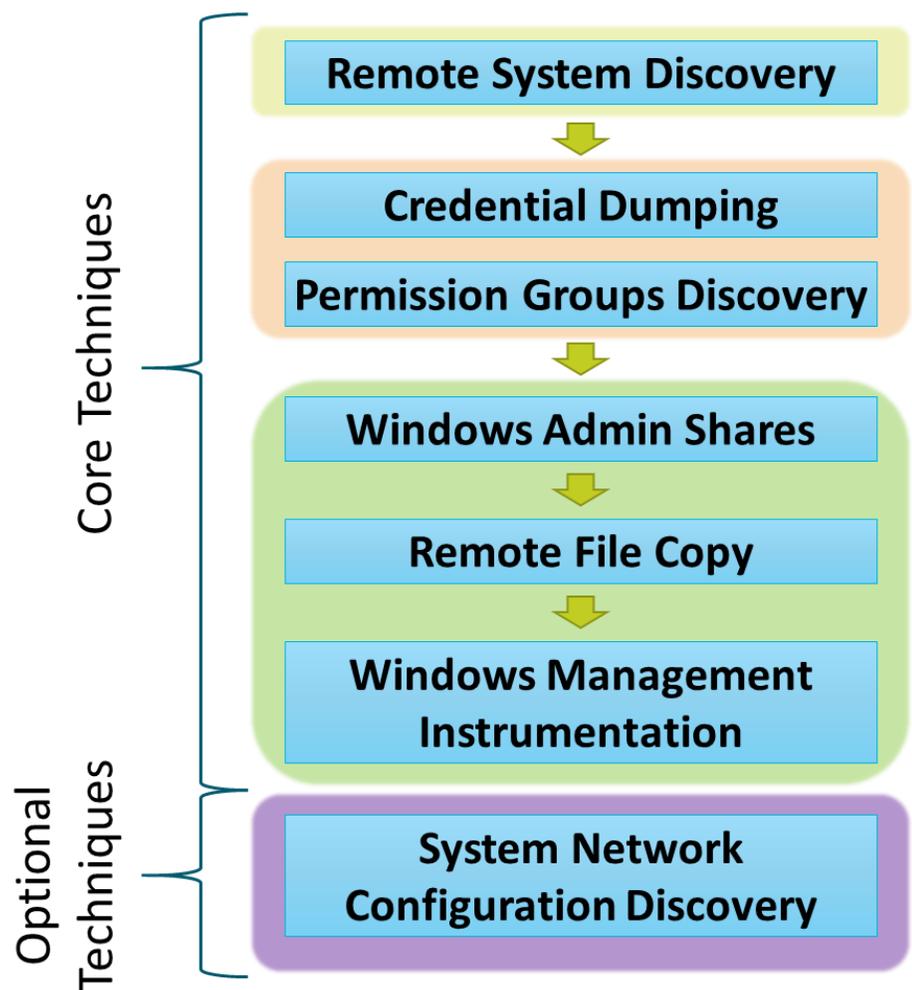
Alice With Determinism

How Often Technique A Followed Technique B



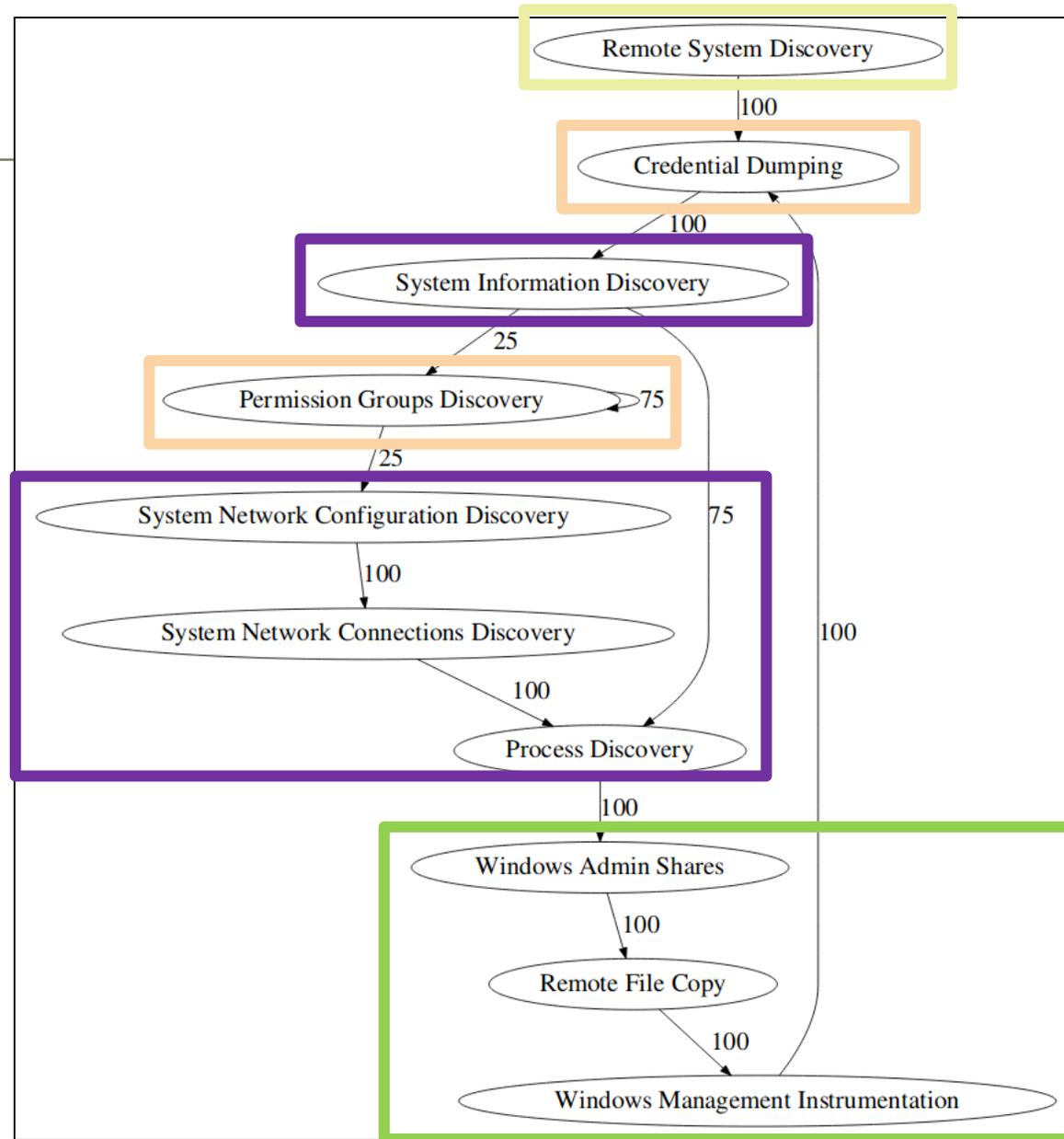
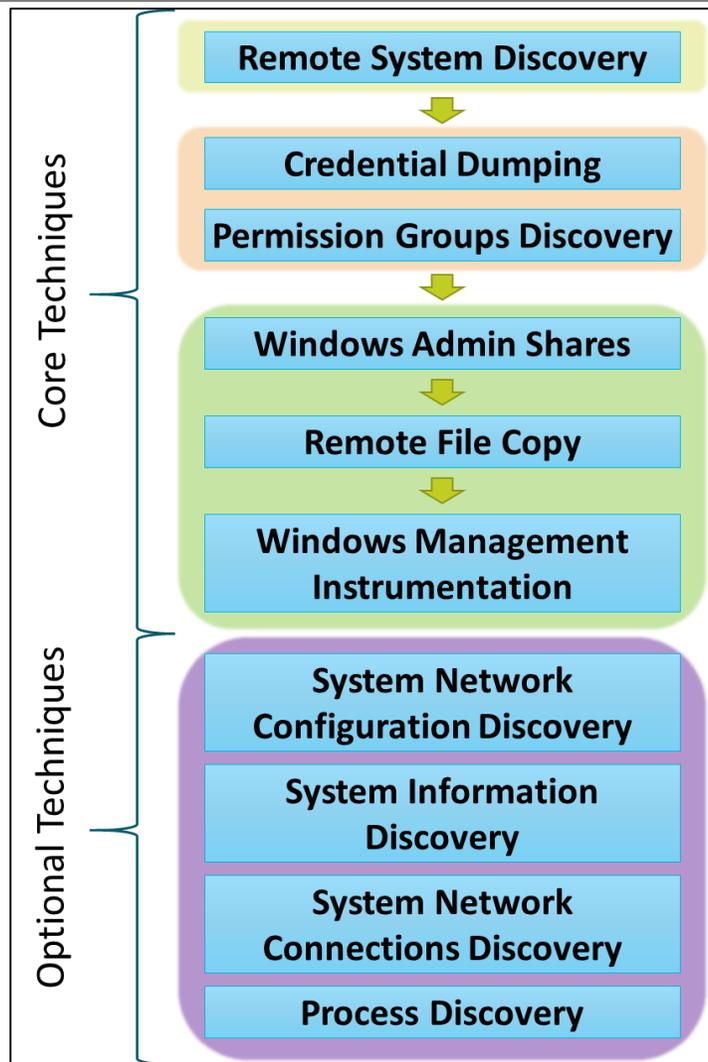
Alice With Determinism

How Often Technique A Followed Technique B

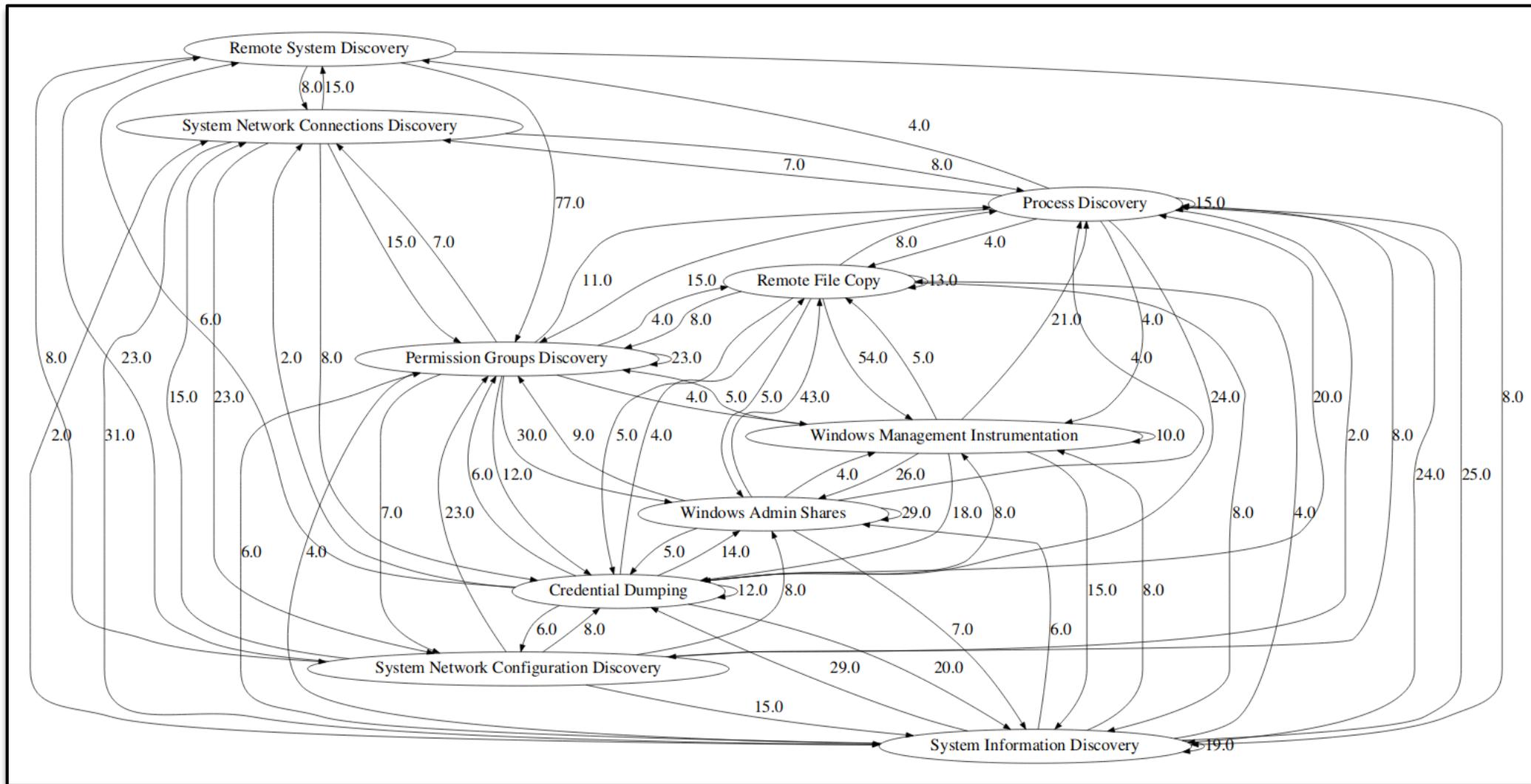


Alice+: With Determinism

Impact of More Techniques



The Flowchart for Alice+ Is Even Harder to Read

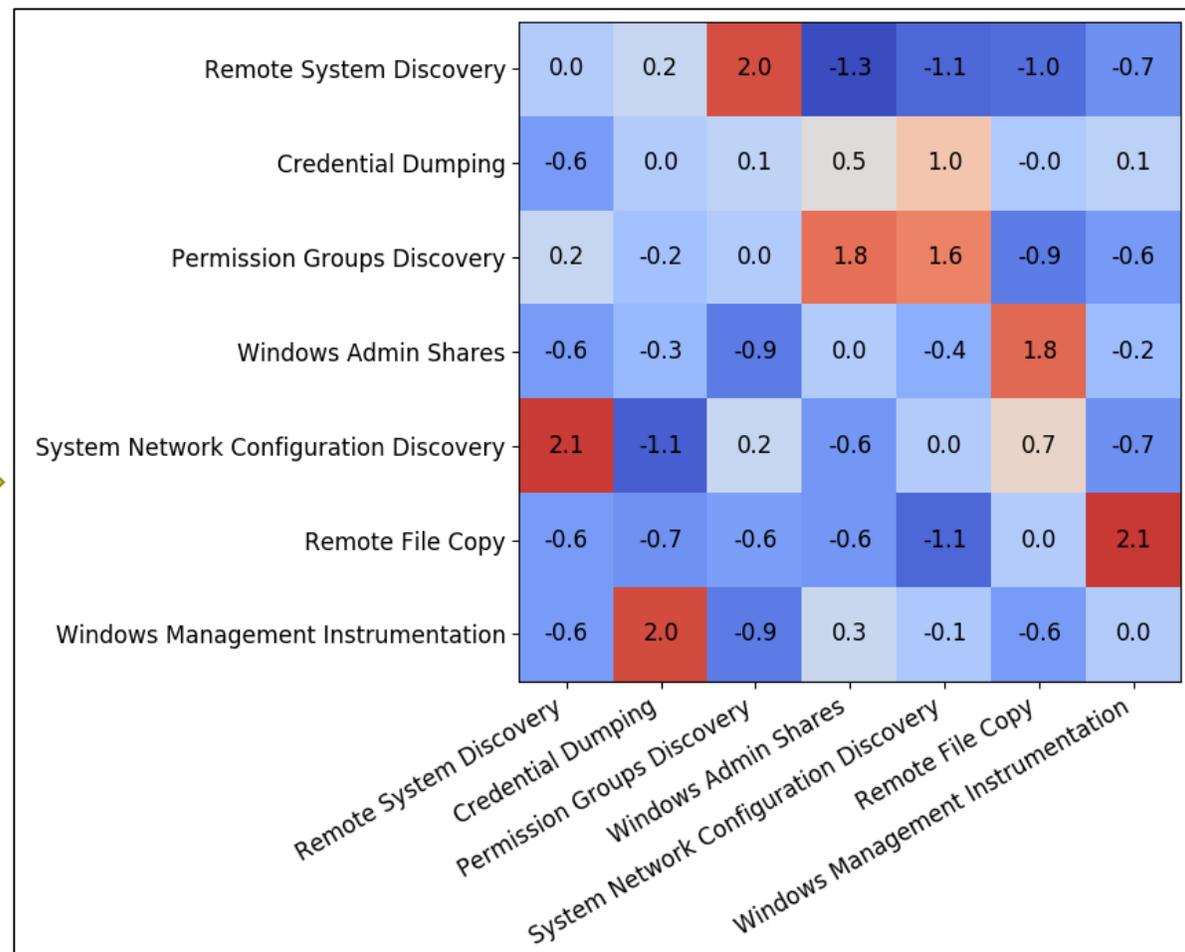
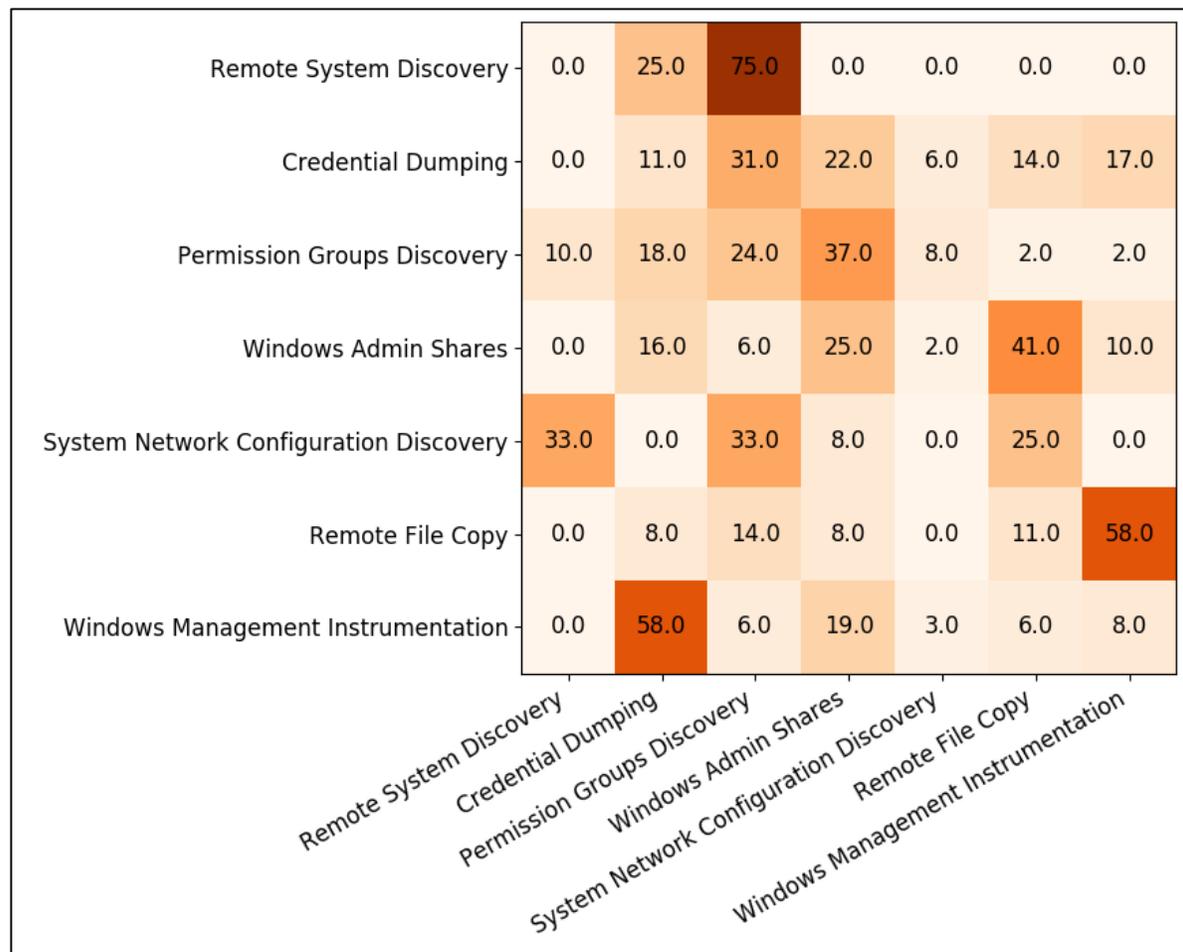


The Challenge in Using Simulations

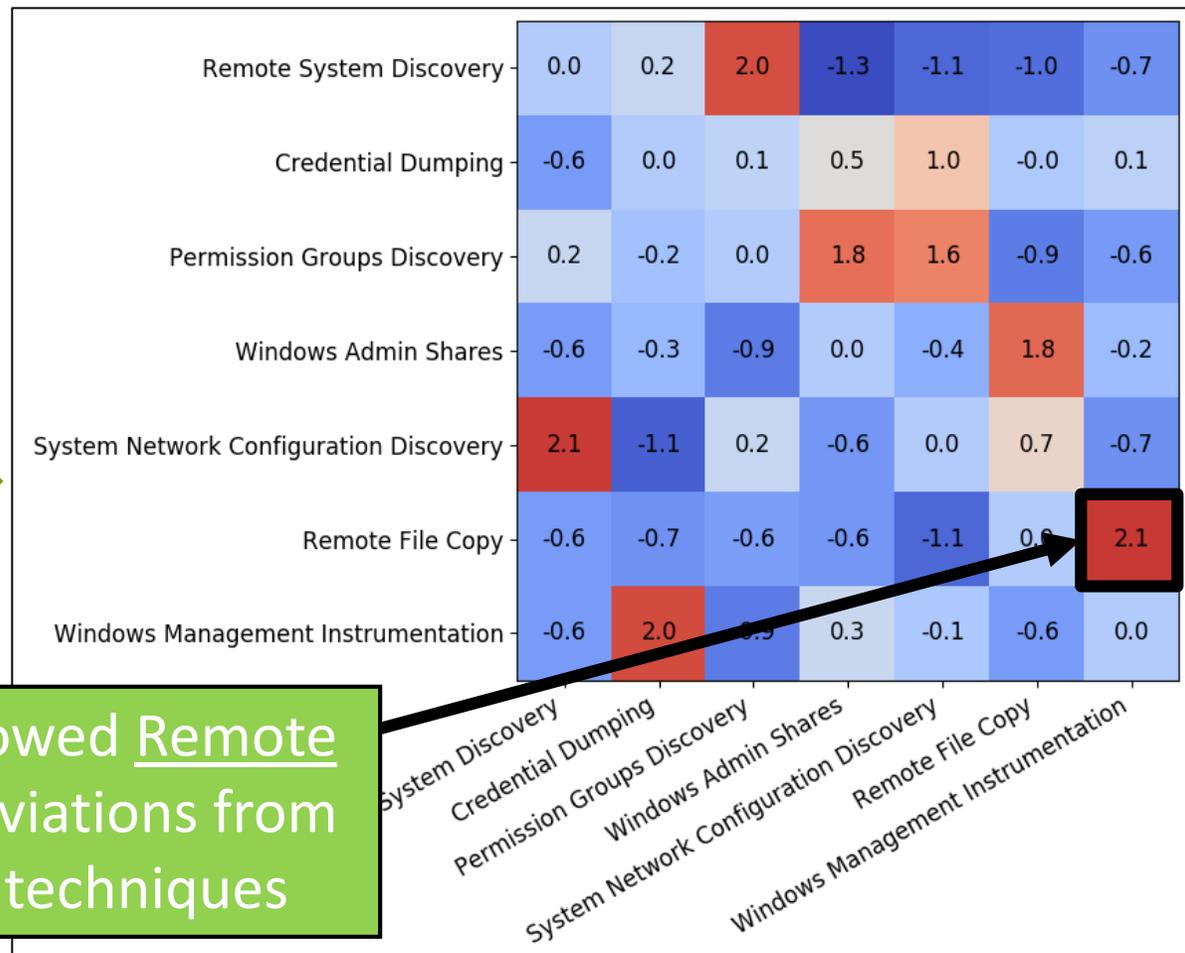
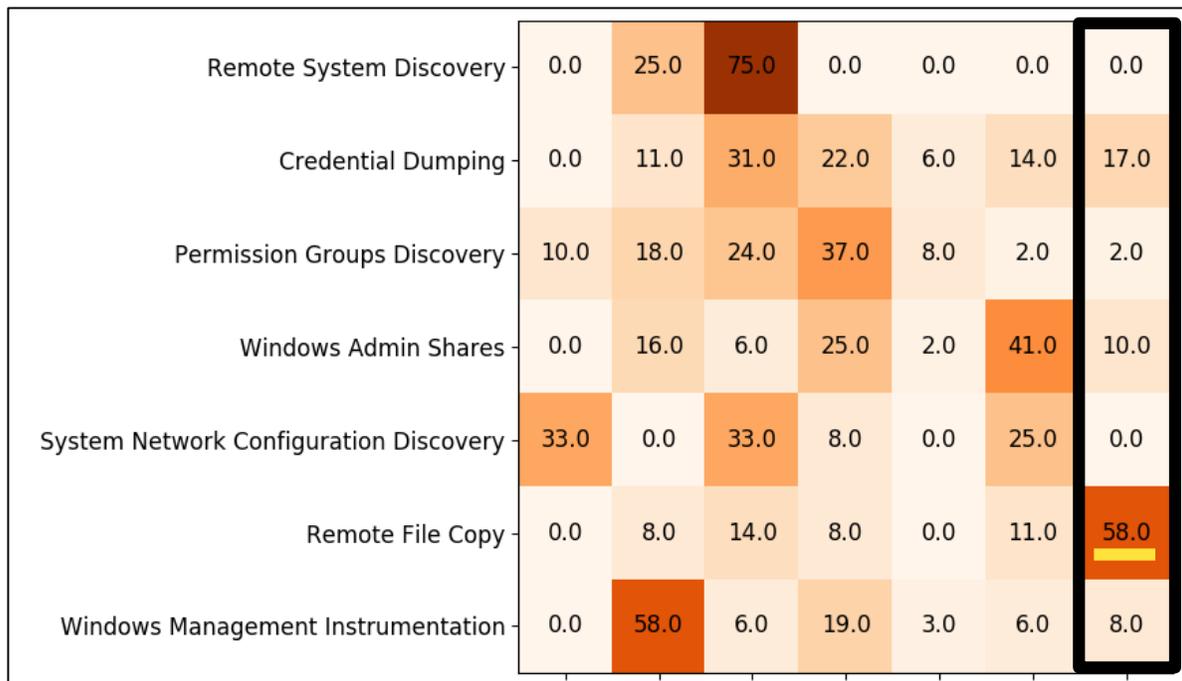
- **Controlling for adversary decision behavior is hard!**
 - Even for somewhat “forced” adversaries there can be significant variance in technique sequencing
 - This problem gets exponential very, very quickly when actions don’t have a well-defined execution structure
- **These kind of charts can be useful for understanding generic technique relationships (e.g., alternatives), but not for technique sequencing**
- **(note: not a problem if we know the decision behavior beforehand!)**

- **So what can we do?**
 - Reusing our work: instead of raw percentages, use deviations from the mean for each column

Sequencing: Deviations from the Mean



Sequencing: Deviations from the Mean



The percentage of time WMI followed Remote File Copy was greater than 2.1 deviations from how often WMI followed other techniques

Sequencing: Deviations from the Mean

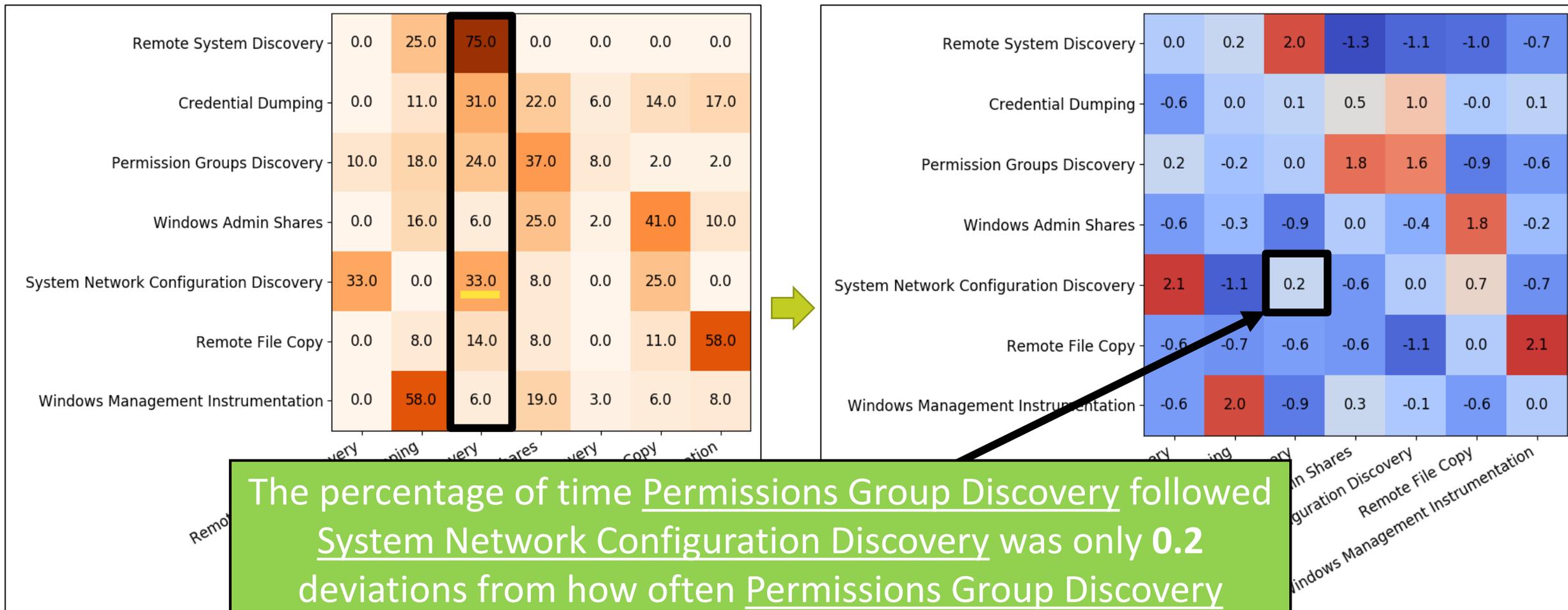
Remote System Discovery	0.0	25.0	75.0	0.0	0.0	0.0	0.0
Credential Dumping	0.0	11.0	31.0	22.0	6.0	14.0	17.0
Permission Groups Discovery	10.0	18.0	24.0	37.0	8.0	2.0	2.0
Windows Admin Shares	0.0	16.0	6.0	25.0	2.0	41.0	10.0
System Network Configuration Discovery	33.0	0.0	33.0	8.0	0.0	25.0	0.0
Remote File Copy	0.0	8.0	14.0	8.0	0.0	11.0	58.0
Windows Management Instrumentation	0.0	58.0	6.0	19.0	3.0	6.0	8.0



Remote System Discovery	0.0	0.2	2.0	-1.3	-1.1	-1.0	-0.7
Credential Dumping	-0.6	0.0	0.1	0.5	1.0	-0.0	0.1
Permission Groups Discovery	0.2	-0.2	0.0	1.8	1.6	-0.9	-0.6
Windows Admin Shares	-0.6	-0.3	-0.9	0.0	-0.4	1.8	-0.2
System Network Configuration Discovery	2.1	-1.1	0.2	-0.6	0.0	0.7	-0.7
Remote File Copy	-0.6	-0.7	-0.6	-0.6	-1.1	0.0	2.1
Windows Management Instrumentation	-0.6	2.0	-0.9	0.3	-0.1	-0.6	0.0

The percentage of time Permissions Group Discovery followed Remote System Discovery was greater than 2 deviations from how often Permissions Group Discovery followed other techniques

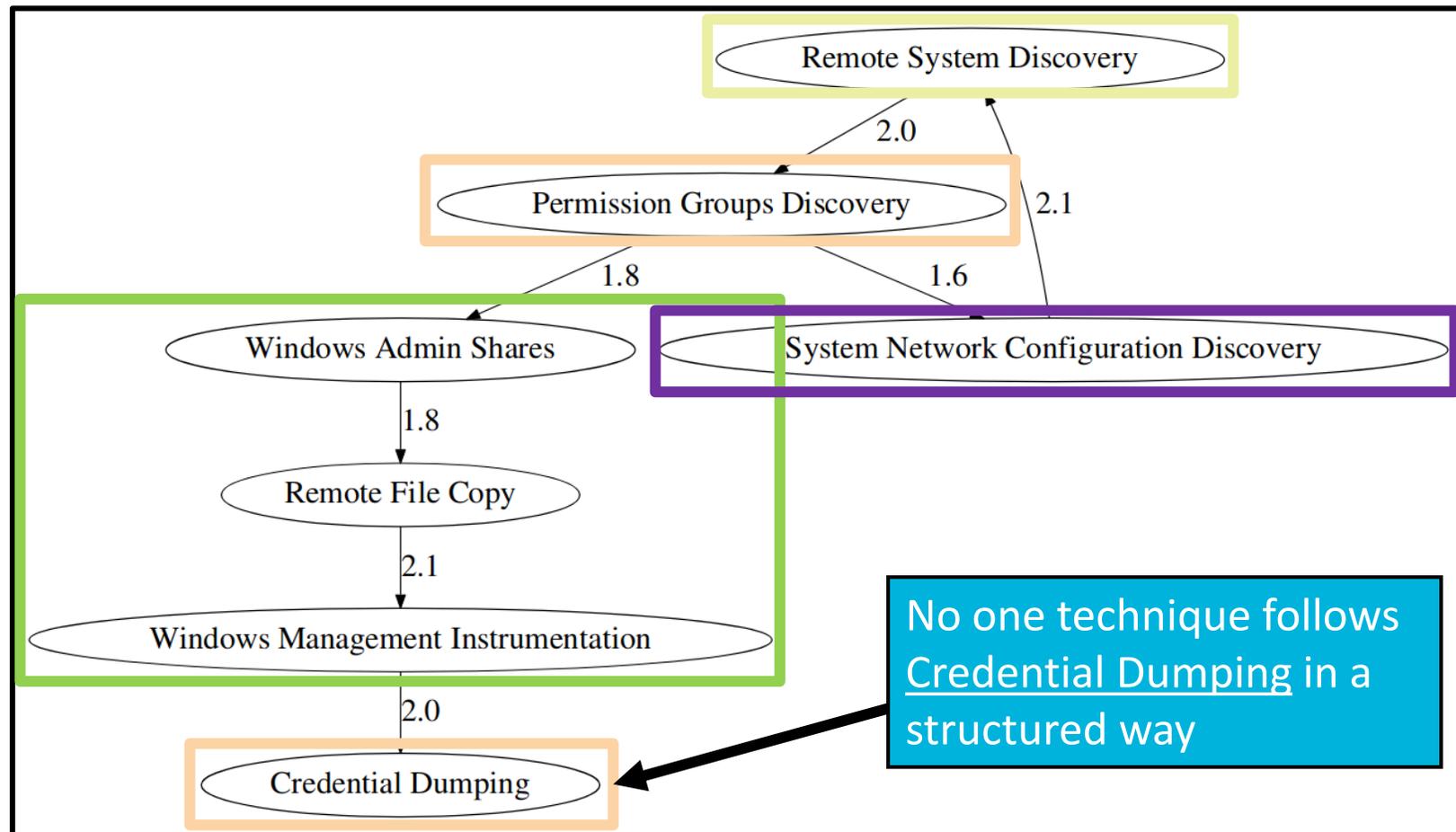
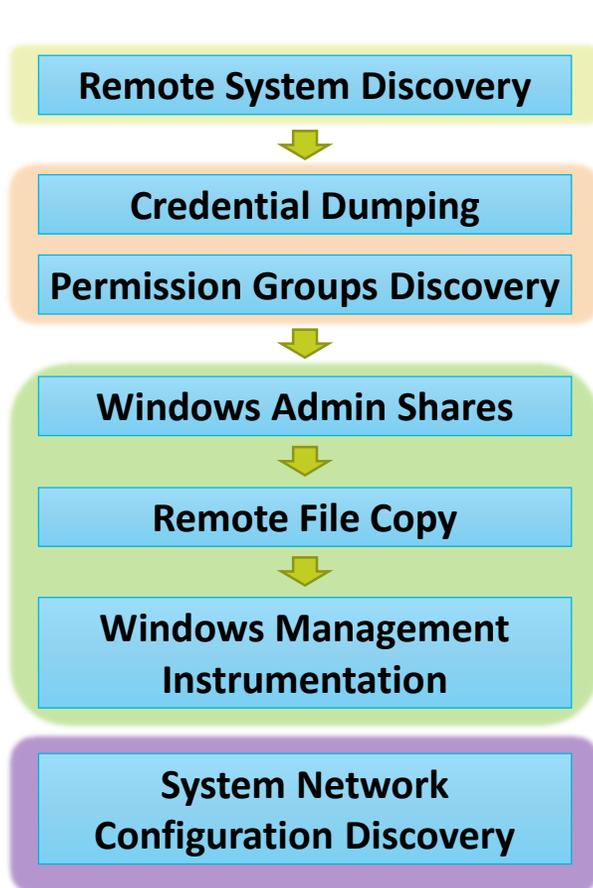
Sequencing: Deviations from the Mean



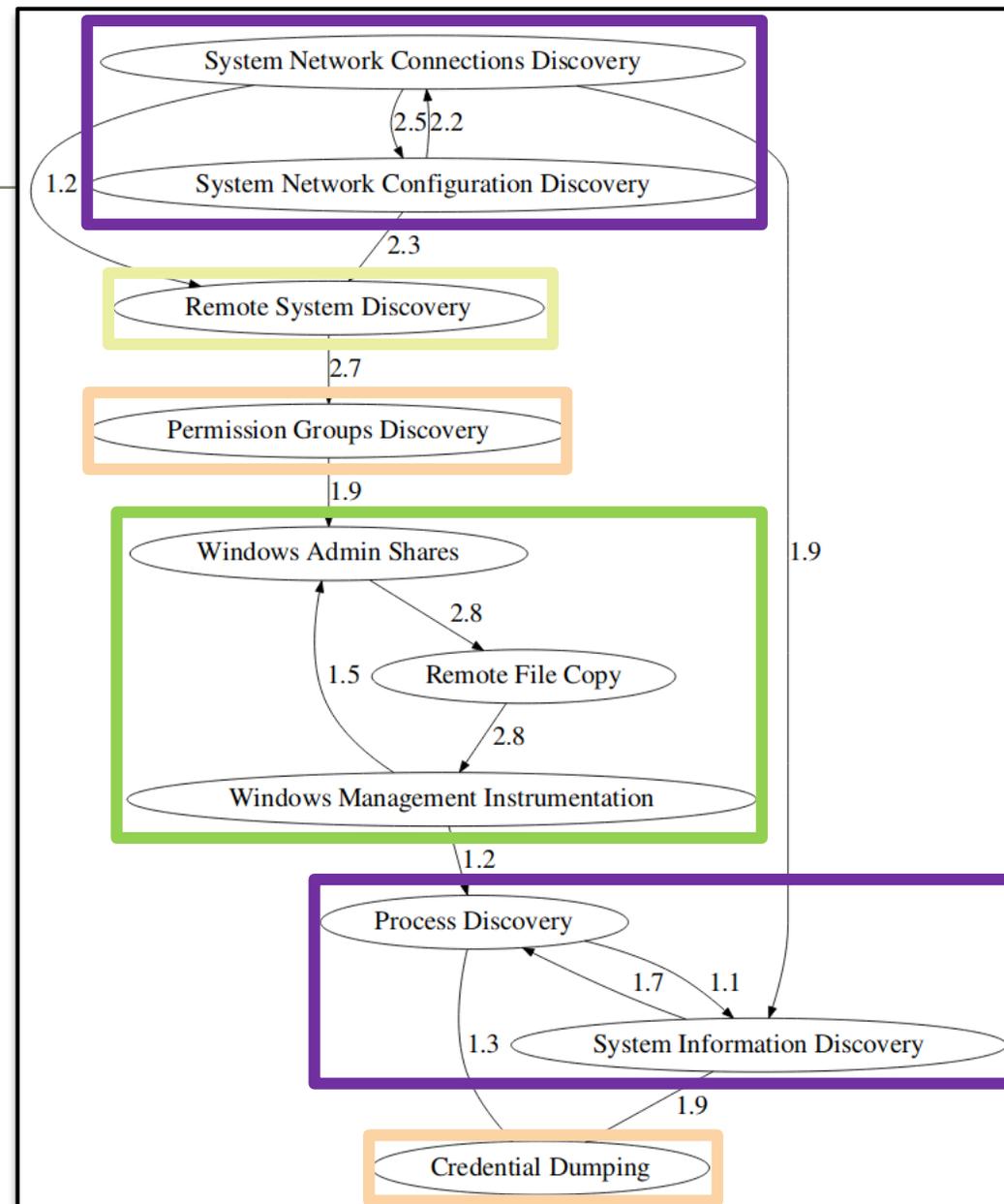
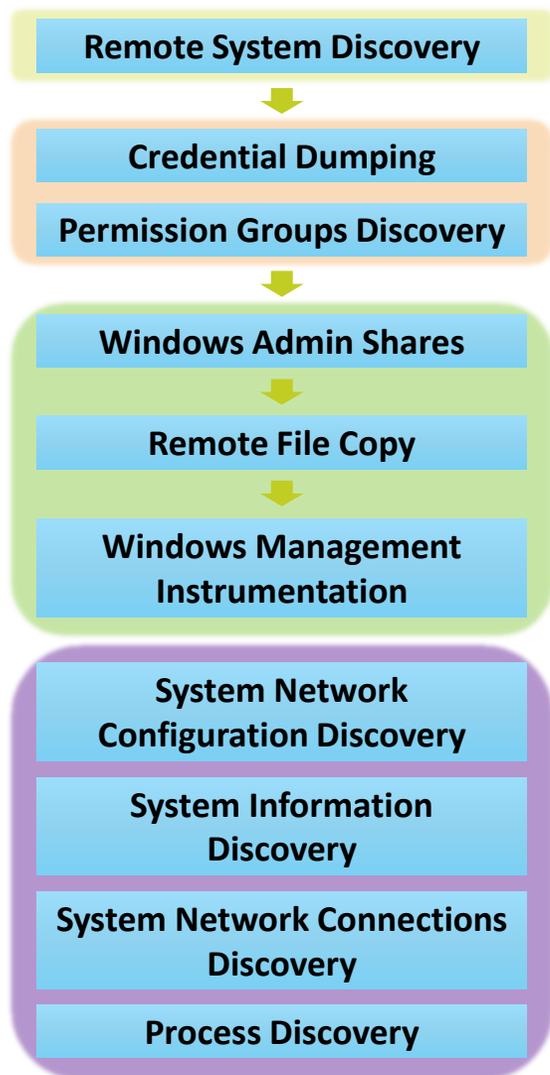
The percentage of time Permissions Group Discovery followed System Network Configuration Discovery was only 0.2 deviations from how often Permissions Group Discovery followed other techniques

Alice: Selective Flow Chart

- Only draw edges with >1 deviation

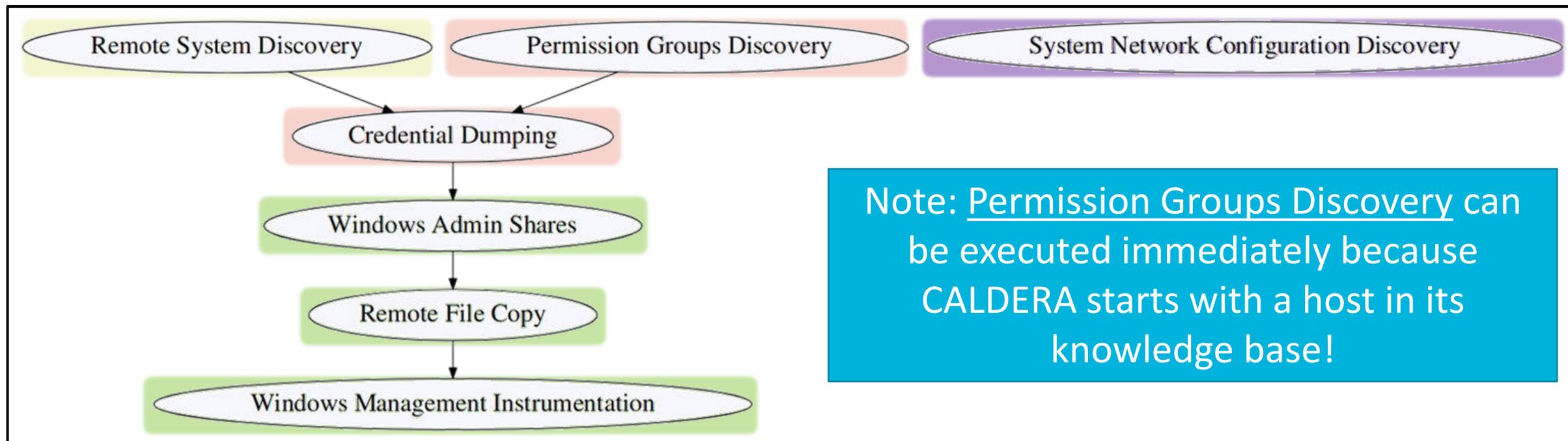


Flow Chart: Alice+



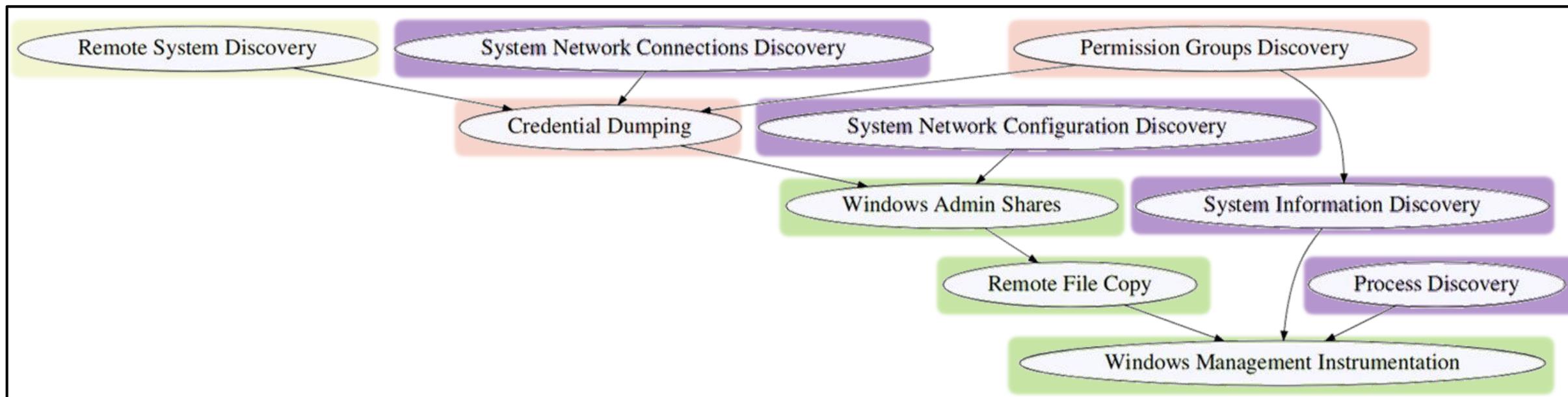
Creating Flowcharts: Technique First Use (Alice)

- If a technique is *always* seen before another it is probably a dependency
- Can trim techniques that are not *always* seen before
 - Some exceptions around alternatives; can look at technique subsets instead



Creating Flowcharts: Technique First Use (Alice+)

- **Works well for Alice! But not for Alice+**
 - Retains core “structure” (yellow -> orange -> green)
 - Show dependencies that are not true
 - Purple techniques showing as mandatory
 - By the time WMI gets executed, most purples have been executed by random choice



Summary: Using Experimental Results

- **Looked at two ways to understand experimental results:**
 1. Techniques immediately following each other
 2. Technique first use inference
- **Both offer insights into technique relationships:**
 - Method 1) can show sequences/dependencies as well as alternatives
 - Method 2) will not show alternatives, but will show sequences/dependencies
- **Both have shortcomings:**
 - Method 1) isn't perfectly accurate, and requires cutoffs
 - Method 2) needs more trials to work better (only considers *first usage*)
 - Method 2) does not work for a deterministic adversary
- **Choose Method 1) if you're looking for sequences + general relationships**
- **Choose Method 2) if you're working with a structured but semi-random profile**

Closing Thoughts

Summary of Approaches

Data/Threat Reporting Analysis

Low barrier to entry; easy to automate, extend, or customize

Suffers from bias; some inaccuracy; lack of specificity

Captures most technique relationships, including implementation overlap

Semantic Modeling Analysis

Very accurate when modeled right; shows lots of relationship info

High barrier to entry (logical modeling); hard to maintain/extend

Captures dependencies + alternatives; provides utility across functions

Experimental Analysis

No need for logical models; less bias than reports; easy to customize

Accuracy dependent on decision-making model; have to encode TTPs

Captures dependencies + sequences

Unsolicited Recommendations – Which Approach is Best?

Hunting

Dependent: hunt for techniques that enable your hypothesis

Alternative: if the hypothesis fails, hunt for a reasonable alternative

Semantic

Data

Detection

Dependent: develop high-fidelity rules by correlating dependent and independent techniques

Alternative: correlate technique execution failures with follow-up alternatives

Experimental

Experimental

Security Engineering

Dependent: configure endpoints to prevent techniques that enable others

Alternative: collect appropriate logs to cover related sets of alternatives

Semantic

Data

But really it depends on your setup!

Links and Contact

- **Andy Applebaum**

- aapplebaum@mitre.org
- @andyplayse4

- **ATT&CK**

- <https://attack.mitre.org>
- @MITREattack
- attack@mitre.org

- **Data + Code**

- <https://github.com/mitre/cti> (STIX data)
- <https://github.com/mitre-attack> (code)

- **CALDERA**

- <https://github.com/mitre/caldera>

- **ATT&CK-based Product Evals**

- <https://attacker.mitre.org/>

- **ATT&CKcon**

- <https://www.mitre.org/attackcon>

- **Blog**

- <https://medium.com/mitre-attack>