Velociraptor Digging Deeper in Linux!



Velociraptor

- □ An advanced, open source endpoint visibility tool.
- Open architecture that is flexible and adaptable to new requirements.
- Implements advanced digital forensics techniques in the form of "Artifacts"
- □ Automates the boring stuff
- □ Scales to your entire endpoint fleet !



Who am I?

Dr Michael Cohen

- Experienced digital forensic software developer.
- Developer of foundation forensic tools including Volatility and Rekall.
- Former lead developer of Grr Rapid Response at Google Inc.





What will you need today?

A Linux computer or virtual machine, with admin access. A copy of Velociraptor from our official release page: https://github.com/Velocidex/velociraptor/releases

A hunting frame of mind.

Velociraptor supports Windows, Linux and MacOS but today we will focus on **Linux!**



What is Velociraptor?

Velociraptor is a unique DFIR tool, giving <u>you</u> power and flexibility through the Velociraptor Query Language (VQL)

5

VQL is used for everything:

- Collecting information from endpoints (also called *clients*)
- Controlling monitoring and response on endpoints
- Controlling and managing the Velociraptor server.





Velociraptor overview

Everything uses the same binary - both clients and server.

- The server is controlled via the server configuration file.
- The client is controlled via the client configuration file.

In this lab, we run the server *and* client on the same machine. In real cases, we typically deploy a Velociraptor server in the cloud.

6







Installing Velociraptor

Download the latest Linux binary from our releases page:

https://github.com/Velocidex/velociraptor/releases

It is just an ELF binary that should work on all Linux distributions later than about 2018.

wget https://github.com/Velocidex/velociraptor/releases/download/v0.3.8/velo ciraptor-v0.3.8-linux-amd64

8

chmod +x velociraptor-v0.3.8-linux-amd64



Download Velociraptor

<u>^ _ D X</u>

ssh.cloud.google.com/projects/velocidex-199308/zones/australia-southeast1-b/instances/linux-conf-vm?authuser=0&hl=en_US&projectNumber...

scudette@linux-conf-vm:~\$ wget https://github.com/Velocidex/velociraptor/releases/download/v0.3.7/velocirantory 0.3.7-linux-amd64

--2020-01-05 00:58:03-- https://github.com/Velocidex/velociraptor/releases/download/v0.3.7/velociraptor-v0.3.7linux-amd64

Resolving github.com (github.com)... 13.236.229.21

Connecting to github.com (github.com) | 13.236.229.21 |: 443... connected.

HTTP request sent, awaiting response... 302 Found

Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/126576769/8033f200-18fa-11ea-8731-a89b 7b969711?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20200105%2Fus-east-1%2Fs3%2Faw s4_request&X-Amz-Date=20200105T005803Z&X-Amz-Expires=300&X-Amz-Signature=6deeeb28a281dc64812c210afbab17029561b4f cf24926e15653411978dfc3ad&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filen ame%3Dvelociraptor-v0.3.7-linux-amd64&response-content-type=application%2Foctet-stream [following]

--2020-01-05 00:58:03-- https://github-production-release-asset-2e65be.s3.amazonaws.com/126576769/8033f200-18fa -11ea-8731-a89b7b969711?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20200105%2Fus-e ast-1%2Fs3%2Faws4_request&X-Amz-Date=20200105T005803Z&X-Amz-Expires=300&X-Amz-Signature=6deeeb28a281dc64812c210a fbab17029561b4fcf24926e15653411978dfc3ad&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attach ment%3B%20filename%3Dvelociraptor-v0.3.7-linux-amd64&response-content-type=application%2Foctet-stream

Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.ama zonaws.com)... 52.216.107.180

Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3 .amazonaws.com)|52.216.107.180|:443... connected.

HTTP request sent, awaiting response... 200 OK

Length: 31981400 (30M) [application/octet-stream]

Saving to: 'velociraptor-v0.3.7-linux-amd64'

2020-01-05 00:58:08 (7.62 MB/s) - 'velociraptor-v0.3.7-linux-amd64' saved [31981400/31981400]

scudette@linux-conf-vm:~\$ chmod +x velociraptor-v0.3.7-linux-amd64
scudette@linux-conf-vm:~\$



Configuring Velociraptor

Everything is controlled by a pair of configuration files.

The configuration files contain key data, making them unique (and secure) to your deployment.

The server configuration file contains private keys - *make sure to secure it!*

Genering new configuration files is easy:

\$ velociraptor config generate -i



scudette@linux-conf-vm:~\$./velociraptor-v0.3.7-linux-amd64 config generate -i
?
Welcome to the Velociraptor configuration generator

I will be creating a new deployment configuration for you. I will begin by identifying what type of deployment you need.

Self Signed SSL Generating keys please wait.... ? Enter the frontend port to listen on. 8000 ? What is the public DNS name of the Frontend (e.g. www.example.com): localhost ? Path to the datastore directory. /tmp ? Path to the logs directory. /tmp ? GUI Username or email address to authorize (empty to end): mic ? GUI Username or email address to authorize (empty to end): ? GUI Username or email address to authorize (empty to end): ? Where should i write the server config file? server.config.yaml ? Where should i write the client config file? client.config.yaml scudette@linux-conf-vm:~\$



Starting the server

The same binary acts as a server or client depending on configuration options.

The previous step generated two files:

client.config.yaml
 server.config.yaml

Start the Velociraptor server and frontend:

velociraptor --config server.config.yaml frontend -v





scudette@linux-conf-vm:~\$./velociraptor-v0.3.7-linux-amd64 --config server.config.yaml frontend -v [INF0] 2020-01-05T01:10:357 Starting Frontend {"build time":"2010-12-07T12:56:30+10.00" "commit":"ff4d5f7"." sion":"0.3.7"} [INFO] 2020-01-05T01:10:36Z Loaded 137 built in artifacts [INF0] 2020-01-05T01:10:36Z Increased open file limit to 999999 [INF0] 2020-01-05T01:10:36Z Launched Prometheus monitoring server on 127.0.0.1:8003 [INF0] 2020-01-05T01:10:36Z Frontend is ready to handle client TLS requests at 0.0.0.0:8000 [INF0] 2020-01-05T01:10:36Z Starting hunt manager. [INF0] 2020-01-05T01:10:362 Launched gRPC API server on 127.0.0.1:8001 [INF0] 2020-01-05T01:10:35Z GUI is ready to handle TLS requests {"listenAddr":"127.0.0.1:8889"} [INF0] 2020-01-05T01:10:30Z Starting Hunt Dispatcher Service. [INF0] 2020-01-05T01:10:36Z Starting Stats Collector Service. [INF0] 2020-01-05T01:10:36Z Starting Server Monitoring Service [INF0] 2020-01-05T01:10:36Z Starting Server Artifact Runner Service [INF0] 2020-01-05T01:10:36Z Starting Client Monitoring Service [INF0] 2020-01-05T01:10:36Z Collecting Client Monitoring Artifact: Generic.Client.Stats [INFO] 2020-01-05T01:10:36Z Collecting Client Monitoring Artifact: Windows.Events.ProcessCreation [INFO] 2020-01-05T01:10:36Z Initial user mic not present, creating [INF0] 2020-01-05T01:10:36Z Starting VFS writing service. [INF0] 2020-01-05T01:10:36Z Collecting Server Event Artifact: Server.Monitor.Health/Prometheus [INF0] 2020-01-05T01:10:36Z Starting interrogation service.

3





Connect to the GUI address mentioned previously:

https://localhost:8889/

Note the certificate error - *this is OK*. It's because we chose self-signed SSL mode. You can click through the warning for now.

In real deployments we use proper SSL certificates.



Your Velociraptor server is ready.

Now let's configure some clients.





Creating a debian package

To permanently install the client and server we need to create debian packages

velociraptor --config server.config.yaml debian server velociraptor --config server.config.yaml debian client



Creating a debian package

scudette@linux-conf-vm (10.152.0.3) - byobu - Google Chrome ^ _ D X sh.cloud.google.com/projects/velocidex-199308/zones/australia-southeast1-b/instances/linux-conf-vm?authuser=0&hl=en US&projectN... scudette@ lnux-conf-vm:~\$./velociraptor-v0.3.7-linux-amd64 --config server.config.yaml debian server.... m:~\$./velociraptor-v0.3.7-linux-amd64 --config server.config.yaml debian client scudette@linux-cont-v n:~\$ ls -l *.deb scudette@linux-conf--rw-r--r-- 1 scudette scudette 13914570 Jan 5 01:57 velociraptor 0.3.7 client.deb -rw-r--r-- 1 scudette scudette 13923280 Jan 5 01:56 velociraptor 0.3.7 server.deb scudette@linux-conf-vm:~\$ sudo dpkg -i velociraptor 0.3.7 server.deb (Reading database ... 38225 files and directories currently installed.) Preparing to unpack velociraptor 0.3.7 server.deb ... Removed /etc/systemd/system/multi-user.target.wants/velociraptor server.service. Unpacking velociraptor-server (0.3.7) over (0.3.7) ... Setting up velociraptor-server (0.3.7) ... Created symlink /etc/systemd/system/multi-user.target.wants/velociraptor server.service \rightarrow /etc/systemd/syst em/velociraptor server.service. scudette@L m:~\$ sudo dpkg -i velociraptor 0.3.7 client.deb Selecting previously unselected package velociraptor-client. (Reading database ... 38225 files and directories currently installed.) Preparing to unpack velociraptor 0.3.7 client.deb ... Unpacking velociraptor-client (0.3.7) ... Setting up velociraptor-client (0.3.7) ... Created symlink /etc/systemd/system/multi-user.target.wants/velociraptor client.service → /etc/systemd/syst em/velociraptor client.service. scudette@L :-\$



The Dashboard

The **Dashboard** shows the current state of the installation:

- How many clients are connected
- Current CPU load and memory footprint on the server.

When running hunts or intensive processing, memory and CPU requirements will increase but not too much.

You can customize the dashboard - it's also just an artifact.



Search Box

Q

🗐 Last Day 🔹

Ξ 🖏

p

Server status

Currently there are 1 clients connected.



Users

Show 10 • entries		Search:
Name	ReadOnly	\$
mic	false	

Clients have a persistent connection to the server.

They're ready to receive your commands.



Currently there are 74 clients connected.

۲

A







Interactively investigate individual clients





To work with a specific client we need to search for it.

Press the **Search** icon to see all the clients



Search for clients hostname, label or client ID.

	linux-co	onf-vm c velocidex-199308	internal		
	Online	ClientID	Host	OS Version	Labels
0	•	C.77772029617b302a	linux-conf-vm.c.velocidex- 199308.internal	debian9.11	





The server collects some high level information about each endpoint.

Click **VQL Drilldown** to see more detailed information:

- Client version
- Client footprint (memory and CPU)

You can customize the information collected and shown by editing the **Generic.Client.Info** artifact.



	linux-conf-vm.c.velocidex-199	R Q linux-conf-vm.c.velocidex-199308.internal Oconnected
	④ Interrogate	O Collected
Ľ		
	linux-conf-vm.c.velocidex-199308	.internal
	Client ID	C.77772029617b302a
	Agent Version	2020-01-05111:52:54+10:00
	Agent Name	velociraptor
ł.	Last Seen At	2020-01-06 00:07:26 UTC
	Last Seen IP	[::1]:55070
	Operating System	linux
	Hostname	linux-conf-vm.c.velocidex-199308.internal
	Release	debian9.11
	Architecture	amd64



10	
	(mar
IVI	Sea

Ξ

1

φ

۶ (۱) (۱)

9

O VFS

(Interrogate

linux-conf-vm.c.velocidex-199308.internal 🥥 connected

0

Overview VQL Drilldown

linux-conf-vm.c.velocidex-199308.internal (C.77772029617b302a) @ 2020-01-05 01:55:39 +0000 UTC

Name 🔺	BuildTime	\$	Labels 🝦	Hostname 🝦	os 🛊	Architecture	Platform 🝦	PlatformVersion	KernelVersion 🝦	Fqdn	\$
velociraptor	2020-01- 05T11:52:54+10:00	22	<	linux-conf-vm	linux	amd64	debian	9.11	4.9.0-11-amd64	linux-conf-vm.c.velocidex- 199308.internal	

Memory and CPU footprint over the past 24 hours

Collected





The Virtual File System (VFS)

The VFS visualizes some server-side information we collect about the clients.

Top level corresponds to the type of information we collect:

- **File** Access the file system using the filesystem API
- NTFS Access the file system using raw NTFS parsing (Windows Only)
- Registry Access the Windows Registry using the Registry API (Windows Only)
- Artifacts A view of all artifacts collected from the client.



Ξ Search Box

🕀 🗀 sbin

+ C srv 🕀 🗀 sys ⊕- 🗀 tmp

🕀 🗀 usr

🕀 🗀 var 🗄 🗀 vmlinuz + C vmlinuz.old

+ ntfs

🖨 🗀 file

*

 ϕ

R

۲

Э

A

1

6

linux-conf-vm.c.velocidex-199308.internal Oconnected Q

m	ic

	0	2	0

>	file > home > scudette	Refresh dire	ctory fro	m		
Down	load Name	endpoint ^{size}	Mode	mtime	atime	ctime
(H)	.bash_history	546	-rw	2020-01-06T00:56:43Z	2020-01-06T04:03:50Z	2020-01-06T04:03
	.bash_logout	220	-rw-rr	2017-05-15T19:45:32Z	2020-01-05T01:29:30Z	2020-01-05T00:55
	.bashrc	3613	-rw-rr	2020-01-05T01:13:13Z	2020-01-06T04:03:12Z	2020-01-05T01:13:
	.byobu	4096	drwxr-xr-x	2020-01-05T01:13:27Z	2020-01-05T01:20:09Z	2020-01-05T01:13:
	.lesshst	48	-rw	2020-01-05T01:28:42Z	2020-01-05T01:28:42Z	2020-01-05T01:28
	.nano	4096	drwxr-xr-x	2020-01-05T01:28:38Z	2020-01-05T01:28:38Z	2020-01-05T01:28
	.profile	675	-rw-rr	2017-05-15T19:45:32Z	2020-01-06T04:03:12Z	2020-01-05T00:55
	.ssh	4096	drwx	2020-01-06T04:07:05Z	2020-01-05T01:42:39Z	2020-01-06T04:07
	.viminfo	1175	-rw	2020-01-05T01:27:06Z	2020-01-05T01:27:06Z	2020-01-05T01:27
H	.wget-hsts	165	-rw-rr	2020-01-05T00:57:35Z	2020-01-05T00:58:03Z	2020-01-05T00:57
	client.config.yaml	2229	-rw	2020-01-05T01:06:14Z	2020-01-06T04:44:45Z	2020-01-05T01:06
> 1 St	file > home > scudette > .b tats TextView HexView	ash_history w CSVView Report	S			
/hc	ome/scudette/ bash_history			Properties		

/home/scudette/.bash_history	§
Size	546
Mode	-TW
Mtime	2020-01-06T00:56:43Z
Atime	2020-01-06T04:03:50Z
Ctime	2020-01-06T04:03:50Z
Last Collected	2020-01-06 05:06:24 UTC ADownload
Fetch from Client	Collect from the client
Peter nonicient	Collect from the client



0 -

Fetch file contents from endpoint





Task: We suspect a user account had been compromised. Determine what the attacker had done.

- Useful artifacts include
 - □ Bash history
 - □ Less history
 - □ VIM history





Ξ	Search Box		linux-conf-vm.c.v	elocidex-199308.	internal 🔵 d	connected	
*	Ġ- Ċ⊐ file ↓ ∲- C⊐ bin						
0	Et Ca boot		.5511		4030	UIWA	2020-01-00104.01.032
		H H	.viminfo		1175	-rw	2020-01-05T01:27:06Z
anc .	⊕ ⊡ etc	н	.wget-hsts		165	-rw-rr	2020-01-05T00:57:35Z
۲	- C home	12 0 15	78186866 "wa"		0000		2020 01 05701-00-147
_	🕀- 🗀 mic	12,0,10	, or				
	🕞 🖿 scudette	# Searc	h String History (new	est to oldest)	:		
	🕁- 🗀 .byobu	# Expre	ssion History (newest	to oldest):			
	🕀- 🗀 .nano	# Input	Line History (newest	to oldest):			
	⊕- 🗀 .ssh	# Debug	Line History (newest	to oldest):			
Э	🕂 🗀 initrd.img	# Regis	ters:				
	+- 🗀 initrd.img.old	Timesan			-		
1	🕀 🗀 lib	U Z	309 ~/.ssh/avthorize	d_keys			
	🕂 🗀 lib64	4,48,2	, 389, 1578187626, "~ .s 0 ~7. ssh/author 120d	sh/authorized_ keys	_keys"		
	🕂 🗀 lost+found	4,49,2	,0,1578186866,"~/.ssh	/authorized_ke	eys"		
	🕂 🗀 media	# Jump1	ist (newest first):				
	🕀 🗀 mnt	4,39,2	389 ~/.ssh/authorize ,389,1578187626,"~/.s	ed_keys sh/authorized_	keys"		
	🕂 🗀 opt	-' 2	0 ~/.ssh/authorized_ 0 1578186866 "~/ ssh	keys	ave"		
	🕂- 🗀 proc	-' 1	0 ~/.ssh/authorized_	keys	sy s		
	🕂 🗀 root	4,39,1	,0,1578186832,"~/.ssh 0 ~/.ssh/authorized	/authorized_ke kevs	eys"		
	🕀- 🗀 run	4,39,1	,0,1578186832,"~/.ssh	/authorized_ke	eys"		
	🕂 🗀 sbin	# Histo	ry of marks within fi	les (newest to	oldest):		
	🕀- 🗀 srv	> ~/.ss	h/authorized keys				
	🕀 🗀 sys		* 1578187625	Θ			
	🕂- 🗀 tmp		^ 2 389 ^ 2 390				
	🕂 🗀 usr		. 2 348 + 1 389				
	🕂 🗀 var		+ 2 348				
	🕂 🗀 vmlinuz						



rises

Use Velociraptor artifacts to automate everything



Automate everything!

We can collect information about *many* things in DFIR cases:

Registry keys, files, WMI queries, sqlite databases …

But we really just want to answer specific questions:

- What program did the attacker run?
- What files were downloaded?
- Did the user connect to a known malicious C&C?
- Did a particular file exist on a client?



Velociraptor uses expert knowledge to find the evidence

A key objective of Velociraptor is encapsulating DFIR knowledge into the platform, so you don't need to be a DFIR expert.

- We have high level questions to answer
- We know where to look for evidence of user / system activities

We build artifacts to collect and analyze the evidence in order to answer our investigative questions.



Velociraptor's superpower: user specified artifacts

An artifact is a YAML file ...

- (therefore user-readable, shareable and editable)
- ☐ ... that answers a question ...
 - ... by collecting data from the endpoint ...
- □ ... and reporting on this data in a human readable way.

Artifacts encode expert knowledge into human reusable components.



+

Linux.Ssh.AuthorizedKeys

Ŵ

Search Box

1

autho

Windows.Detection.Impersonation

Q

Windows.Sys.CertificateAuthorities

Windows.Sys.Programs

Linux.Ssh.AuthorizedKeys

Type: client

Find and parse ssh authorized keys files.

Parameters



Exercise: Collect all users' authorized keys

Select **Collected Artifacts** to view all artifacts previously collected. Click **Collect More Artifacts** to open the **New Artifact Wizard**.

36

Search for an artifact that fetches authorized keys.

Click Add to add the artifact to the list for collection.

Click Next to start the collection.

This allows us to determine which keys provide access to which user account


Search Box Q linux-conf-vm.c.velocidex-199308	3.internal 😝 connected	0 mic
+ 🛍 🔳 🖓 💠		
New Artifact Collection - Select Artifacts to col	lect	X Creato mic
autho Linux, Ssk-AuthorizedKeys Windows.Detection.Impersonation	Linux.Ssh.AuthorizedKeys	mic
Windows.Sys.CertificateAuthorities	Find and parse ssh authorized keys files.	mic
Windows.393.Programs	Name Type Default	mic
Selected Artifacts:	sshKeyFiles .ssh/authorized_keys*	mic
Linux.Ssh.AuthorizedKeys	Source	
Clear	<pre>1 LET authorized_keys = SELECT * from foreach(2 row={ 3 SELECT Uid, User, Homedir from Artifact.Linux.Sys.Users() 4 }, 5 query={ 6 SELECT FullPath, Mtime, Ctime, User, Uid from glob(7 globs=Homedir + '/' + sshKeyFiles) 8 })</pre>	
sshKeyFiles .ssh/authorized_keys* Ops/Sec		
		Next

Ξ		Search Box	Q linux-conf-vm.c.velocidex-199308.internal Oconnected			0 mic
*	+		\$			
\$	State	FlowId	Artifacts Collected	Creation Time	Last Active	Creator
x	*	F.BO9D4KSOOCB00	Linux.Ssh.AuthorizedKeys	2020-01-06 06:23:15 UTC	2020-01-06 06:23:56 UTC	mic
1	~	F.BO9C9PHE3IIR2	System: VFS. Download File	2020-01-06 05:25:58 UTC	2020-01-06 05:25:58 UTC	mic
	~	F.BO9C9M83GREH6	System.VFS.DownloadFile	2020-01-06 05:25:45 UTC	2020-01-06 05:25:45 UTC	mic
	~	F.BO9C10SSHHPOS	System.VFS.ListDirectory	2020-01-06 05:07:15 UTC	2020-01-06 05:07:15 UTC	mic
<u>م</u>	Lin	ux.Ssh.AuthorizedKeys w 10 • entries			Search:	•
A	Uid	d 📥 User 🍦 FullPath	Key			
	1000	0 mic /home/mic/.s	ssh/authorized_keys AAAAB3NzaC1yc2EAAAADAQABAAABAQDELIxd5FPC	DcE4CAEAvUNSQVNxHobfOdtZHYI	kSRUYxcq6TZrigPmpQztF9MUU/mKT	5x22MHnOZOc
	100	1 scudette /home/scude	ette/.ssh/authorized_keys AAAAB3NzaC1yc2EAAAADAQABAAABAQDELIxd5FPC	DcE4CAEAvUNSQVNxHobfOdtZHYI	kSRUYxcq6TZrigPmpQztF9MUU/mKT	5x22MHnOZOc
	Sho	wing 1 to 2 of 2 entries			Previous	1 Next

© 2020 Velocidex Enterprises

V



The artifact is running a VQL query - returning one row per key.

Click **Prepare Download** to download the results of this artifact collection through your web browser (see next slide).

The result is a ZIP file with any collected files and a CSV file of the collection results.



	Search Box	Q linux-conf-vm.c.velocidex-199308.interna	al 🦳 connected			0 mid		
-		•						
State	FlowId	Artifacts Collected		Creation Time	Last Active	Creato		
~	✓ F.BO9D4KSOOCB00 Linux.Ssh.AuthorizedKeys			2020-01-06 06:23:15 UTC	2020-01-06 06:23:56 UTC	mic		
~	F.BO9C9PHE3IIR2	System.VFS.DownloadFile		2020-01-06 05:25:58 UTC	2020-01-06 05:25:58 UTC	mic		
~	F.BO9C9M83GREH6	System.VFS.DownloadFile		2020-01-06 05:25:45 UTC	2020-01-06 05:25:45 UTC	mic		
~	F.BO9C10SSHHPOS	System.VFS.ListDirectory		2020-01-06 05:07:15 UTC	2020-01-06 05:07:15 UTC	mic		
0\	rerview		Results					
	Artifact Names	Linux.Ssh.AuthorizedKeys	Ar	tifacts with Results ["Linux.Ssh.A	uthorizedKeys"]			
	Flow ID	F.BO9D4KSOOCB0O		Uploaded Bytes 0 / 0				
	Creator Start Time	MIC 2020-01-06-06-22:15 LITC		Files uploaded				
	Last Active	2020-01-06 06:23:15 0 TC		Prepare Do	ownload			
	State	TERMINATED	A	vailable Downloads	1657 2020-01-06			
	Ops/Sec	Unlimited		F.BO9D4KS	Bytes 09:28:09.99740 UTC	4581 +0000		
Pa	rameters							
	sshKeyFiles	.ssh/authorized_keys*						

2



The ZIP file contains a directory structure for each client with the collected artifacts stored in csv files.





Scenario - Admin is leaving

- Bob is an IT administrator with a lot of access.
- Bob just sent his resignation letter.
- Which machines of our 25,000 servers/laptops/cloud instances, does Bob have access to?
- □ What systems has Bob logged into in the last 2 months?

BTW

We don't have SSO or centralized logging (\underline{v})



Query local SSH logs

SSH logins are normally recorded in /var/log/auth.log which is rotated periodically. e.g.:

Jan 5 09:56:55 DevBox sshd[1953]: Accepted password for mic from 192.168.0.5 port 36836 ssh2

We want to parse the logs for this specific message. We could write a regular expression

... but ...

Grok is a de facto standard in the world of log parsing



Grokking authorization logs

Grok expressions are used to apply regex to log lines and capture into structured JSON dict

Jan 5 09:56:55 DevBox sshd[1953]: Accepted password for mic from 192.168.0.5 port 36836 ssh2

```
%{SYSLOGTIMESTAMP:Timestamp}
%{SYSLOGHOST:logsource} %{SYSLOGPROG}:
%{DATA:event} %{DATA:method} for (invalid
user )?%{DATA:user} from %{IPORHOST:ip}
port %{NUMBER:port} ssh2(:
%{GREEDYDATA:system.auth.ssh.signature})?
```

Linux.Syslog.SSHLogin

Type: client

Parses the auth logs to determine all SSH login attempts.

Parameters

Name	Туре	Default
syslogAuthLogPath		/var/log/auth.log*
SSHGrok		%{SYSLOGTIMESTAMP:Timestamp} (?:%{SYSLOGFACILITY})?%{SYSLOGHOST

Source

1	<pre>SELECT timestamp(string=Event.Timestamp) AS Time,</pre>
2	Event.IP AS IP,
3	Event.event AS Result,
4	Event.method AS Method,
5	Event.user AS AttemptedUser,
6	FullPath
7	FROM foreach(
8	row={
9	SELECT FullPath FROM globs=syslogAuthLogPath)
10	}, query={
11	SELECT grok(grok=SSHGrok, data=Line) AS Event, FullPath
12	FROM parse_lines(filename=FullPath)
13	WHERE Event.program - "sshd"
14	3)
15	

syslog Linux.Syslog.SSHLogin	syslogAuthLogPath	/var/log/auth.log*
	SSHGrok	%{SYSLOGTIMESTAMP:Timestamp} (?:%{SYSLOGFACILITY})?%{SYS
	Source	
Selected Artifacts: Linux.Syslog.SSHLogin Clear	Add 2 Event.IP 3 Event.eve 4 Event.eve 5 Event.uet 5 Event.uet 5 Event.uet 6 FullPath 7 FROM foreach(8 row={ 9 SELECT Full 10 }, query={ 11 SELECT grownerse 13 WHERE Even 14 }) Remove	AS IP, nt AS Result, hod AS Method, r AS AttemptedUser, lPath FROM glob(globs=syslogAuthLogPath) k(grok=SSHGrok, data=Line) AS Event, FullPath _lines(filename=FullPath) t.program = "sshd"
yslogAuthLogPath	/var/log/auth.log*	
SHGrok Ops/Sec Maximum Time 600	%{SYSLOGTIMESTAMP:Timestam	p} (?:%{SYSLOGFACILITY})?%{SYSLOGHOST:logsource} %{SYSLOGPRO



+] 💠					
tate	FlowId	Artifacts Collected			Creation Time	Last Active	Creato
~	F.BOAK95NB8HITQ	Linux.Syslog.SSHLog	in		2020-01-08 02:55:18 UTC	2020-01-08 02:55:20	UTC mic
	E BOA JEENEOEUEO	Custom Linux Cus CU			2020 01 00 01 40 02 UTC	2020 01 08 01:40:06	EUTO mia
Ar	tifact Collection Uple	oaded Files Reque	ests Results	Log Reports			
Sh	ow 10 V entries					Search:	
Т	me	IP	♦ Result ♦	Method	4	AttemptedUser	FullPath
202	20-01-05T00:55:43Z	74.125.41.103	Accepted	publickey		scudette	/var/log/auth.log
202	20-01-05T01:29:28Z	122.109.172.84	Accepted	password		scudette	/var/log/auth.log
202	20-01-05T01:29:36Z	122.109.172.84	Accepted	pas Legitimate	ogins (maybe?)	cudette	/var/log/auth.log
202	20-01-05T01:29:58Z	122.109.172.84	Accepted	password		scudette	/var/log/auth.log
202	20-01-05T01:33:06Z	178.128.52.97	Failed	password		user	/var/log/auth.log
202	20-01-05T01:41:45Z	159.65.159.81	Eailed	password		vuy	/var/log/auth.log
202	20-01-05T01:42:15Z	122.109.172.84	Accepted	Brute force cr	ackers	scudette	/var/log/auth.log
202	20-01-05T01:45:04Z	122.109.172.84	Accepted	password		scudette	/var/log/auth.log
202	20-01-05T01:54:40Z	59.10.5.156	Failed	password		ftp01	/var/log/auth.log
202	20-01-05T01:54:59Z	27,150,169,223	Failed	password		webapp	/var/log/auth.loc





Velociraptors job is just to collect and preserve evidence:

- Collect files to the server
- Run queries on the endpoint and store result sets on the server

Velociraptor **does not** index or analyze the results of the queries!

You can export data as Zip files containing the CSV or JSON files that were collected for offline analysis. You can also export data to Elasticsearch and Kibana (ELK stack)



State FlowId	Artifacts Collected	Creation Time	Last Active	Creator	
✓ F.BOAK95NB8HITQ	Linux.Syslog.SSHLogin	2020-01-08 02:55:18 UTC	2020-01-08 02:55:20 UTC	mic	
	Custom Linux Suo SLIID	2020 01 00 01:40:02 UTC	2020 01 00 01-40-06 LITC	mio	
Artifact Collection Upl	paded Files Requests Results Log Repor	rts			
Overview		Results			
Artifact Names	Linux.Syslog.SSHLogin	Artifacts with Results ["Linu:	x.Syslog.SSHLogin"]		
Flow ID	F.BOAK95NB8HITQ	Uploaded Bytes 0 / 0			
Creator	mic	Files uploaded 0			
Start Time	2020-01-08 02:55:18 UTC	Download Results Pre	pare Download		
Last Active	2020-01-08 02:55:20 UTC				
State	TERMINATED	Available Downloads	14797 202	0-01-08	
Ops/Sec	Unlimited	F.BO	AK95NB8HITQ.2ip Sytes +00	00 UTC	
Parameters					
Parameters syslogAuthLogPath	/var/log/auth.log*				

Velociraptor command line

Velociraptor offers many command line tools. e.g.:

List content of the zip files

velociraptor unzip --list F.BOAK95NB8HITQ.zip

Extract and filter collected data as JSON

velociraptor unzip --csv F.BOAK95NB8HITQ.zip clients/C.77772029617b302a/artifacts/Linux.Syslog.SSHL ogin/F.BOAK95NB8HITQ.csv

--where "AttemptedUser =~ 'mic'"



```
<u>mic@trek:~/Downloads</u>$velociraptor unzip --list F.BOAK95NB8HITQ.zip
  "Filename": "FlowDetails",
  "Size": 1119
},
  "Filename": "clients/C.77772029617b302a/collections/F.BOAK95NB8HITQ/logs",
  "Size": 360
},
  "Filename ": "clients/C.77772029617b302a/artifacts/Linux.Syslog.SSHLogin/F.BOAK95NB8HITQ.csv",
  "Size": 92106
]mic@trek:~/Downloads$ velociraptor unzip --csv E_BOAK95NB8HHTQ.zip_clients/C.77772029617b302a/artifacts/Li
nux.Syslog.SSHLogin/F.BOAK95NB8HITQ.csv --where "Attem<u>ptedUser =~ 'mic'"</u>
  "Time": "2020-01-07T00:52:10Z",
  "IP": "122.109.172.84",
  "Result": "Accepted",
  "Method": "publickey",
  "AttemptedUser": "mic",
  "FullPath": "/var/log/auth.log"
},
  "Time": "2020-01-07T00:52:29Z",
  "IP": "122.109.172.84",
  "Result": "Accepted",
  "Method": "publickey",
  "AttemptedUser": "mic",
  "FullPath": "/var/log/auth.log"
mic@trek:~/Downloads$
```



Hunting across the entire deployment



What is hunting?

Any artifact that can be collected on a single computer, can be simultaneously hunted across the entire network.

A hunt can cover a group of clients, or the whole network.

A hunt will continue running until it expires, or is stopped.

As new machines appear, they automatically join in the hunt.

Downloading the hunt results generates a ZIP file with all the uploaded files as well as a large CSV or JSON file with combined results from each client in the same file.



Search Box Q linux-conf-vm.c.velocidex-199308.	nternal Oconnected	0 mic
+ > = =		
New Hunt - Select Artifacts to collect		x
author Haux, Ssh.AuthorizedKeys Windows.Detection.Impersonation Windows.Sys.CertificateAuthorities Windows.Sys.Programs Selected Artifacts:	Linux.Ssh.AuthorizedKeys Type: client Find and parse ssh authorized keys files. Parameters Name Type Default 	
Linux.Ssh.AuthorizedKeys Clear Remove	<pre>Source 1 LET authorized_keys = SELECT * from foreach(2 row={ 3 SELECT Uid, User, Homedir from Artifact.Linux.Sys.Users() 4 }, 5 query={ 6 SELECT FullPath, Mtime, Ctime, User, Uid from glob(7 globs=Homedir + '/' + sshKeyFiles) 8 })</pre>	
sshKeyFiles .ssh/authorized_keys* Ops/Sec Timeout		

+		Ö						
tatus	Hunt ID	Description	Create Time	Start Time	Expires	Client Limit	Clients Scheduled	Cr
X	H.ca1a7fa6	All Authorized Keys	2020-01-06 06:35:17 UTC	2020-01-06 06:35:54 UTC	2020-01-13 06:35:17 UTC	Unlimited	1	mi
	Artifact N	amos Linux Sch Auth	orizedkevs		Total scheduled 1			
	Artifact N Hi Ci	ames Linux.Ssh.Authount ID H.ca1a7fa6 reator mic	orizedKeys		Total scheduled 1 Finished clients 1 Download Results	Download		
	Artifact N Hi Ci Creation	ames Linux.Ssh.Authe unt ID H.ca1a7fa6 reator mic Time 2020-01-06 06:	orizedKeys 35:17 UTC	A	Total scheduled 1 Finished clients 1 Download Results Prepare	Download		
	Artifact N Hi Creation Expiry	ames Linux.Ssh.Authe unt ID H.ca1a7fa6 mic Time 2020-01-06 06: Time 2020-01-13 06: State RELINING	35:17 UTC 35:17 UTC	Av	Total scheduled 1 Finished clients 1 Download Results Prepare ailable Downloads	Download		
	Artifact N Hi Creation Expiry Op	ames Linux.Ssh.Autho unt ID H.ca1a7fa6 mic Time 2020-01-06 06: Time 2020-01-13 06: State RUNNING s/Sec Unlimited	35:17 UTC 35:17 UTC	Av	Total scheduled 1 Finishea clients 1 Download Results Prepare allable Downloads	Download		



The ZIP file contains a single CSV file with combined output from all clients

Extract +	H.ca1a7fa6.zip		٩ =	: _ • ×
< > 🟠 Location: 🖿 /				
🔹 🖥 H.ca1a7fa6.zip	Name	Size -	Туре	Modified
 clients C.77772029617b302a artifacts Linux.Ssh.AuthorizedKeys collections F.BO9DAIO9RBO6I 	clients HuntDetails All Linux.Ssh.AuthorizedKeys.csv	1.2 kB 815 bytes 1.1 kB	Folder unknown CSV docu	30 November 30 November



What access does a Key have?

A single key can give access to multiple user accounts of many systems. Our hunt collects all the authorized keys on our endpoints in a large CSV file. We can search for the hosts that the key will allow access to.

```
]mic@trek:-/Downloads$ velociraptor unzip --csv H.ca1a7fa6.zipຶAll Linux.Ssh.AuthorizedKeys.csv"> --where "Key =- 'Gu5BusVC5VWlPA7x86hdXU'
  "Uid": "1000"
  "User(: "mic",
  "FullPath": "/home/mic/.ssh/authorized keys",
 "Key": "AAAAB3NzaC1yc2EAAAADAQABAAABAQDELIxd5FP0cE4CAEAvUNSQVNxHobf0dtZHYkSRUYxcq6TZriqPmpQztF9MUU/mKT5x22MHn0Z0ceY2/lAn5ngMtYUEB6sHjSgFVHshKsr
Hk8PsU/gWqjId80IvMjRwwL9mJozpuyaTMwR5bWkqbJPJ4+V9mNLnGd9st+p0WFJFKVuX0NJYJ1BgYXgHRdY6oy/2maKtEI3CJeLSXz87K515tHBepVk6hSSRe8Mi0SV9EmcbH2k0aDwi74N6
d7s0d+I60Xo2K06E0rBpKEtd8nRRqPbL56ICbGu5BusVC5VWlPA7x86hdXU+qfx+Pj".
  "Comment": "mic@trek\n",
 "Mtime": "2020-01-05T00:52:08Z",
 "FlowId": "F.B09DAI09RB06I",
  "ClientId": "C.77772029617b302a"
  "Fqdn" 🗲 linux-conf-vm.c.velocidex-199<u>308.internal</u>"
  "Uid": "1001".
  "User": "scudette",
  "FullPath": "/home/scudette/.ssh/authorized_keys",
 "Key": "AAAAB3NzaC1yc2EAAAADAQABAAABAQDELIxd5FP0cE4CAEAvUNSQVNxHobf0dtZHYkSRUYxcq6TZrigPmpQztF9MUU/mKT5x22MHn0Z0ceY2/lAn5ngMtYUEB6sHjSgFVHshKsr
ik8PsU/gWgjId80IvMjRwwL9mJozpuyaTMwR5bWkgbJPJ4+V9mNLnGd9st+p0WFJFKVuX0NJYJ1BgYXgHRdY6oy/2maKtEI3CJeLSXz87K515tHBepVk6hSSRe8Mi0SV9EmcbH2k0aDwi74N6
d7s0d+I6QXo2K06E0rBpKEtd8nRRaPbL56ICbGu5BusVC5VWlPA7x86hdXU+afx+Pj",
 "Comment": "mic@trek\n",
  "Mtime": "2020-01-06T05:42:15Z",
  "FlowId": "F.B09DAI09RB06I",
  "ClientId": "C.77772029617b302a",
  "Fadn": "linux-conf-vm.c.velocidex-199308.internal"
```

Surgical collection of evidence



Trying to keep one step ahead

- Proactive DFIR work involves trying to keep ahead of current Tools Technique and Procedures (TTP)
- Reading a lot of blog posts
- Following the <u>Mitre Att&ck framework</u>

Ultimately we are looking for signals we can use to alert when an endpoint is compromised.



MITRE	ATT&CK	<u>Matrices</u> Blog I∕* (Tactics Contribute	Tech	niques -	Mitigation	is ▼ Gi	oups So	oftware F	lesources	•	Search site	
M A T R I C E PRE-ATT&C Enterprise Windows	S K	Home > Ma	ntrices > L	^{inux} atrix							Launo	ch the ATT&C	K™ Navigator ⊡
macOS Linux Cloud	~	Below are the platform.	he tactics	and techni	que repres	enting the M	ITRE ATT&	CK Matrix™ f	for Enterprise	e. The Matr	ix contains in	formation fo	r the Linux
Mobile	~	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
		Drive-by Compromise	Command- Line Interface	.bash_profile and .bashrc	Exploitation for Privilege Escalation	Binary Padding	Bash History	Account Discovery	Application Deployment Software	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
		Exploit Public- Facing Application	Exploitation for Client Execution	Bootkit	Process Injection	Clear Command History	Brute Force	Browser Bookmark Discovery	Exploitation of Remote Services	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
		Hardware Additions	Graphical User Interface	Browser Extensions	Setuid and Setgid	Compile After Delivery	Credential Dumping	File and Directory Discovery	Internal Spearphishing	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
		Spearphishing Attachment	Local Job Scheduling	Create Account	Sudo	Connection Proxy	Credentials from Web Browsers	Network Service Scanning	Remote File Copy	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
		Spearphishing Link	Scripting	Hidden Files and Directories	Sudo Caching	Disabling Security Tools	Credentials in Files	Network Sniffing	Remote Services	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
		Spearphishing via Service	Source	Kernel Modules and Extensions	Valid Accounts	Execution Guardrails	Exploitation for Credentia Access	Password Policy Discovery	SSH Hijacking	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe

MITRE	ſ&CK™	Matrices Blog I2ª → C	Tactics Techniques Mitigations Groups Software Resources Search site
ENTERPRISE	.▼ S	Home > Tec Priva	chniques > Enterprise > Private Keys
	~	Private cryp	otographic keys and certificates are used for authentication, (decryption, and digital signatures. ^[1] ID: T1145
Execution Persistence Privilege Escalation Defense Evasion Credential Access Account Manipulation	* * *	Adversaries Remote Ser Common ke .cer, .p7b, .a SSH keys of Private keys also use Inp Adversary to relating to c	Tactic: Credential Access Platform: Linux, macOS, Windows Permissions Required: User Data Sources: File monitoring Contributors: Itzik Kotler, SafeBreach Version: 1.0 Created: 14 December 2017 Last Modified: 18 July 2019
Bash History Brute Force		Proce	dure Examples
Cloud Instance Metadata API		Name	Description
Credential Dum	nping	Ebury	Ebury has intercepted unencrypted private keys as well as private key pass-phrases. [6]
Credentials fro Web Browsers	m	Empire	Empire can use modules like Invoke-SessionGopher to extract private key and session information. ^[5]
Credentials in F	Files	jRAT	jRAT can steal keys for VPNs and cryptocurrency wallets. ^[7]

Machata Machata has scanned and looked for cryptographic koys and cortificate file extensions [8]

Credentials in

Find all un-encrypted private keys

SSH private key files reside in ~/.ssh/id_rsa or ~/.ssh/id_dsa
 Encrypted files have a line like:

----BEGIN RSA PRIVATE KEY----

Proc-Type: 4, ENCRYPTED

DEK-Info: AES-128-CBC, 278B4B1765B49ECF24679ABF589A60CC



Linux.Ssh.PrivateKeys

FROM glob(globs=KeyGlobs)

Type: client

SSH Private keys can be either encryted or unencrypted. Unencrypted private keys are more risky because an attacker can use them without needing to unlock them with a password.

This artifact searches for private keys in the usual locations and also records if they are encrypted or not.

Parameters

4

56

Name	Туре	Default	
KeyGlobs		/home/*/.ssh/id_{rsa,dsa}	
Source			
1 SELECT FL	llPath, imestamp(epoch=Mti	me.Sec) AS Mtime,	

}, then="Yes", else="No") AS Encrypted

SELECT * from yara(rules="wide ascii:ENCRYPTED", files=FullPath)





State	FlowId	Artifacts Collected		Creation Time	Last Active	Creator
~	F.BOA0KDTVVQFIC	Linux.Ssh.PrivateKeys		2020-01-07 04:33:59 UTC	2020-01-07 04:34:00 UTC	mic
~	F.BO9V25TU2VMFC Linux.Syslog.SSHLogin F.BO9DAIO9RBO6I Linux.Ssh.AuthorizedKeys			2020-01-07 02:46:47 UTC	2020-01-07 02:53:05 UTC	mic
~				2020-01-06 06:35:55 UTC	2020-01-06 06:35:55 UTC	H.ca1a7fa6
~	F.BO9D4KSOOCB0O	Linux.Ssh.AuthorizedKeys		2020-01-06 06:23:15 UTC	2020-01-06 06:23:56 UTC	mic
~	F.BO9C9PHE3IIR2	System.VFS.DownloadFile		2020-01-06 05:25:58 UTC	2020-01-06 05:25:58 UTC	mic
Ar	ifact Collection Uploade	ed Files Requests Resul	ts Log Reports			
Ar	iifact Collection Uploade	ed Files Requests Resul	ts Log Reports			Ţ
Ar Lir Sho	tifact Collection Uploade nux.Ssh.PrivateKeys w 10 ▼ entries	ed Files Requests Resul	ts Log Reports		Search:	·
Ar Lir Sho	tifact Collection Uploade nux.Ssh.PrivateKeys w 10 ▼ entries	ed Files Requests Resul	ts Log Reports		Search:	
Ar Lir Sho Fu	tifact Collection Uploade nux.Ssh.PrivateKeys w 10 ▼ entries IllPath me/mic/.ssh/id_rsa	ed Files Requests Resul	ts Log Reports Mtime 2020-01-07T04:33:45Z		Search: Encrypted Yes	· ·



ATT&CI	Blog 2 Contribute	Search site
ENTERPRISE - ECHNIQUES Iverview	Home > Techniques > Enterprise > Setuid and Setgid Setuid and Setgid When the setuid or setgid bits are set on Linux or macOS for an application, this means that the application will run with the privileges of the owning user or group respectively. ^[1]	,
xecution versistence	Normally an application is run in the current user's context, regardless of which user or group owns the application. There are instances where programs need to be executed in an elevated context to function properly, but the user running them doesn't need the elevated Tactic: Property of the current user's context, regardless of which user or group owns the application. There are instances where programs need to be executed in an or platform.	ivilege Escalation, ce : Linux, macOS
.bash_profile and .bashrc Accessibility	privileges. Instead of creating an entry in the sudoers file, which must be done by root, any user can specify the setuid or setgid flag to be set for their own applications. These bits are indicated with an "s" instead of an "x" when viewing a file's attributes via 1s -1. The chmod	ons Required: User ons: Administrator, root
Features Account	program can set these bits with via bitmasking, chmod 4777 [file] or via shorthand Data Sou naming, chmod u+s [file]. Process n command	r ces : File monitoring, nonitoring, Process I-line parameters
AppCert DLLs AppInit DLLs	An adversary can take advantage of this to either do a shell escape or exploit a vulnerability in an application with the setsuid or setgid bits to get code running in a different user's context. Additionally, adversaries can use this mechanism on their own malware to make use they're able to execute in alcosted contexts in the future [2].	1.0 14 December 2017 lified: 24 June 2019
Application Shimming		

Mitigations -

Groups

Software

Resources -

Matrices

Tactics 🔻

Techniques 🔻

Create a SUID backdoor!

echo 'int main() { setresuid(0,0,0);
system("/bin/sh"); }' > privshell.c

gcc -o privshell privshell.c

rm privshell.c

chown root:root privshell

chmod u+s privshell

https://medium.com/@airman604/9-ways-to-backdoor-a-linux-box-f5f83bae5a3c





```
root@linux-conf-vm:/tmp# echo 'int main() { setresuid(0,0,0); system("/bin/sh"); }' > privshell.c
root@linux-conf-vm:/tmp# gcc -o privshell privshell.c
privshell.c: In function 'main':
privshell.c:l:14: warning: implicit declaration of function 'setresuid' [-Wimplicit-function-declaration]
int main() { setresuid(0,0,0); system("/bin/sh"); }
privshell.c:l:32: warning: implicit declaration of function 'system' [-Wimplicit-function-declaration]
int main() { setresuid(0,0,0); system("/bin/sh"); }
root@linux-conf-vm:/tmp# rm privshell.c
root@linux-conf-vm:/tmp# rm privshell.c
root@linux-conf-vm:/tmp# chown root:root privshell
root@linux-conf-vm:/tmp# chown root:root privshell
root@linux-conf-vm:/tmp# chown u+s privshell
```

```
scudette@linux-conf-vm:~$ /tmp/privshell
# id
uid=0(root) gid=1002(scudette) groups=1002(scudette),4(
#
```



Hunt for SUID backdoors

Ξ

Glob recursively for all files in a directory

Filter only those files with SUID bit enabled.

Could take a while on large system - throttle the endpoint if needed

Search Box Q III	ux-conf-vm.c.velocidex-199308.internal Oconnected		0
F 🖋 🛍			
suid	Linux.Svs.SUI	D	
Linux.Sys.SUID	Type: client		
	When the setuid or setgid bits will run with the privileges of th current user's context, regard programs need to be execute need the elevated privileges. user can specify the setuid or "s" instead of an "x" when view bitmasking, chmod 4777 [file]	are set on Linux or macOS for an e owning user or group respective less of which user or group owns t in an elevated context to function instead of creating an entry in the setgid flag to be set for their own a ving a file's attributes via Is -I. The or via shorthand naming, chmod u	application, this means that the applicatio ely [1]. Normally an application is run in th he application. There are instances where n properly, but the user running them does sudoers file, which must be done by root, applications. These bits are indicated with chmod program can set these bits with vi 1+s [fule].
	An adversary can take advant with the setsuid or setgid bits use this mechanism on their o future [2].	age of this to either do a shell esc to get code running in a different u wn malware to make sure they're	ape or exploit a vulnerability in an applicat iser's context. Additionally, adversaries ca able to execute in elevated contexts in the
	References:		
	 https://attack.mitre.org/ 	techniques/T1166/	
	Parameters		
	Name	Туре	Default
	GlobExpression		/usr/**
	Source		
	1 SELECT Mode.String A 2 FullPath, Siz 3 timestamp(epo 4 Sys.Uid AS 00 5 Sys.Gid AS 06 5 EPON 210/210/2007	AS Mode, te, sch=Mtime.Sec) AS Mtime, wherID, supposed by MURBE Mede == 1	

suid Linux.Sys.SUID		execute in elevated contexts in References: • https://attack.mitre.org/t Parameters	the future [2]. echniques/T1166/		
		Name	Туре	Default	
Selected Artifacts:	Add	GlobExpression		/usr/**	
Linux.Sys.SUID					
		Source			
Clear	Remove	1 SELECT Mode.String As 2 FullPath, Siz 3 timestamp(epor 4 Sys.Uid AS Own 5 Sys.Gid AS Gro 6 FROM glob(globs=Globb) 7	S Mode, 2, ch=Mtime.Sec) AS Mtime, nerID, pupID Expression) WHERE Mode	=~ '^u'	
GlobExpression	/**				
Maximum Time					



Search Box	Q linux-conf-vm.c.velocid	lex-199308.internal 🔵 connected			0 mi
+ 🛍 🔳 d	2 4				
State FlowId	Artifacts Collected		Creation Time	Last Active	Creator
✓ F.BOAJ2E48NTSQA	Linux.Sys.SUID		2020-01-08 01:32:40 UTC	2020-01-08 01:32:47 UTC	mic
Artifact Collection U	ploaded Files Requests Results	Log Reports			
Linux.Sys.SUID				Search:	
Mode 📥 Fi	ullPath	Size Mtime		wnerID 🔶 GroupID	
urwxr-xr-x /bir	n/mount	44304 2018-03-0713	L8:29:09Z 0	0	
urwxr-xr-x /bir	n/ping	61240 2016-11-10T0	06:23:32Z 0	0	
urwxr-xr-x /bir	n/su	40536 2017-05-1713	11:59:59Z 0	0	
urwxr-xr-x /bin	n/umount	31720 2018-03-0712	L8:29:09Z 0	0	
urwxr-xr-x /us	sr/bin/chfn	50040 2017-05-1713	11:59:59Z 0	0	
urwxr-xr-x /us	r/bin/chsh	40504 2017-05-1713	11:59:59Z 0	0	
urwxr-xr-x /us	sr/bin/gpasswd	75792 2017-05-1713	11:59:59Z 0	0	
urwxr-xr-x /us	sr/bin/newgrp	40312 2017-05-1713	11:59:59Z 0	0	
urwxr-xr-x /us/	sr/bin/passwd	59680 2017-05-17T	1:59:59Z 0	0	
urwxr-xr-x /us/	sr/bin/sudo	140944 2019-10-12T	L4:20:21Z 0	0	
urwxr-xr-x /usr	sr/lib/openssh/ssh-keysign	440728 2019-07-15T	L3:32:09Z 0	0	
urwxr-xr-x //tm	np/privshell	8704 2020-01-08T0	01:23:45Z 0	0	

Customizing artifacts





Artifacts simply contain VQL statements.

- It's easy to modify existing artifacts to your needs.
- As you learn VQL, you can easily write your own.
- Custom artifacts start with the **Custom** prefix.
- You can use official or custom artifacts interchangeably.
- You can also contribute your artifacts to the Velociraptor project





Customize Artifacts

The SUID artifact is great but we don't know if someone replaced one of the standard binaries

Lets calculate the hash of each SUID binary

This will show us outliers - some machines might have a different hash which is not known. Maybe VirusTotal know about it?

We need to edit the Linux.Sys.SUID artifact to add a hash column


= 🔯

۲

D

Ph.

Search Box

Linux.Sys.SUID

Q

0 mic

Linux.Sys.SUID

Type: client

When the setuid or setgid bits are set on Linux or macOS for an application, this means that the application will run with the privileges of the owning user or group respectively [1]. Normally an application is run in the current user's context, regardless of which user or group owns the application. There are instances where programs need to be executed in an elevated context to function properly, but the user running them doesn't need the elevated privileges. Instead of creating an entry in the sudoers file, which must be done by root, any user can specify the setuid or setgid flag to be set for their own applications. These bits are indicated with an "s" instead of an "x" when viewing a file's attributes via Is -1. The chmod program can set these bits with via bitmasking, chmod 4777 [file] or via shorthand naming, chmod u+s [file].

An adversary can take advantage of this to either do a shell escape or exploit a vulnerability in an application with the setsuid or setgid bits to get code running in a different user's context. Additionally, adversaries can use this mechanism on their own malware to make sure they're able to execute in elevated contexts in the future [2].

References:

https://attack.mitre.org/techniques/T1166/

Parameters

Na	me	Туре	Default	
Gl	obExpression		/usr/**	
Sc	ource			
1	SELECT Mode.String AS Mode,			
3	timestamp(epoch=Mtime_Se	AS Mtime		
4	Svs.Uid AS OwnerID.	ioj no ricano,		
5	Sys.Gid AS GroupID			
6	FROM glob(globs=GlobExpression)	WHERE Mode =~ '	^u'	



Add/Modify an artifact

3

4

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Linux.Sys.Sl

1 hame: Custom.Linux.Sys.SUID 2 description: |

When the setuid or setgid bits are set on Linux or macOS for an application, this means that the application will run with the privileges of the owning user or group respectively [1]. Normally an application is run in the current user's context, regardless of which user or group owns the application. There are instances where programs need to be executed in an elevated context to function properly, but the user running them doesn't need the elevated privileges. Instead of creating an entry in the sudoers file, which must be done by root, any user can specify the setuid or setgid flag to be set for their own applications. These bits are indicated with an "s" instead of an "x" when viewing a file's attributes via ls -l. The chmod program can set these bits with via bitmasking, chmod 4777 [file] or via shorthand naming, chmod u+s [file]. An adversary can take advantage of this to either do a shell escape

or exploit a vulnerability in an application with the setsuid or setgid bits to get code running in a different user's context. Additionally, adversaries can use this mechanism on their own malware to make sure they're able to execute in elevated contexts in the future [2].

24 ## References:

25 - https://attack.mitre.org/techniques/T1166/

Sat	ve	Ar	ti.	fa	ct
20	••				~ ~

Name	Туре	Default
GlobExpression		/usr/**
Source		

that the application cation is run in the instances where unning them doesn't be done by root, any are indicated with an these bits with via

ility in an application y, adversaries can ed contexts in the

Search Bo				
	1000			

i

Linux.Sys.SL

suid

+

0

Add/Modify an artifact

16 17 18 19 20 21 22 23	An adversary can take advantage of this to either do a shell escape or exploit a vulnerability in an application with the setsuid or setgid bits to get code running in a different user's context. Additionally, adversaries can use this mechanism on their own malware to make sure they're able to execute in elevated contexts in the future [2].	
24	## References:	
25 26	- https://attack.mitre.org/techniques/T1166/	
27 -	parameters.	
28 -	- name: GlobExpression	
29	default: /usr/**	
30		
31 -	sources+	
32 -	- aueries	
22	- Select Mode String AS Mode	
24	- Select Houe, set ing AS Houe,	
25	timestamp(opsk=Mtime_Sec) AS Mtime	
25	Such a felore and the second s	
27	Sys. old AS Group ID	
20	Sys.old AS droupid,	
20		
39	PROM groungrous-ocodexpression, where mode =~ u	
40		
	Save Artifact	
	Name	Type

*

ility in an application y, adversaries can ed contexts in the



Default

/usr/**



New Artifact Collection - Select A Step 1 out of 2		in me nume [z].		
Custom.Linux.Sys.SUID	• https://attack.mitre.org	g/techniques/T1166/		
	Name	Туре	Default	1a7fa
Selected Artifacts:	Add GlobExpression		/usr/**	
Custom.Linux.Sys.SUID	Source			
Clear	1 SELECT Mode.String 2 FullPath, S: 3 timestamp(e) 4 Sys.Uid AS 5 Sys.Gid AS 6 FROM gtob(globs=Globs) 8	AS Mode, Lze, boch=Mtime.Sec) AS Mtime, boch=Mtime.Sec) AS Mtime, softward,	=~ '∧u'	
GlobExpression	<i>J**</i>			
Ops/Sec Maximum Time 600				

×-	Search Box	Q	linux-conf-vm.c.vel	ocidex-199308.internal	\ominus connected			0
+	1 C	\$						
tate F	lowid	Artifacts Colle	cted			Creation Time	Last Active	Crea
✓ F	BOAJ5SNSQEU5O	Custom.Linux.S	ys.SUID			2020-01-08 01:40:02 UTC	2020-01-08 01:40:06 UTC	mic
		Linux Sve SLID	6			2020-01-08 01-32-40 LITC	2020-01-08 01:32:47 LITC	mic
Artifa	act Collection Upload	led Files Re	quests Results	Log Reports				
Cust Show	tom.Linux.Sys.SUID						Search:	
Mod	le 🔺 FullPath	\$ Size	e 🍦 Mtime		GroupID 🝦	Hash		
urwx x	-xr- /tmp/privshell	8704	2020-01- 08T01:23:45Z	0	•	MD5 : 040abfbcd907f8cbabdadd SHA1 : 3e6b921491d007d11855a SHA256 : 9bf1e07e1d06c739e924b7ca1bc	8ec2ef5c8c ae14ce2da2d6738d2827 726c2d61a8fe24d4b2dda60515d95	52780cad
urwxr x urwxr x	-xr- /tmp/privshell -xr- /usr/bin/chfn	5004	2020-01- 08T01:23:45Z 0 2017-05- 17T11:59:59Z	0	0	MD5 : 040abfbcd907f8cbabdadd SHA1 : 3e6b921491d007d11855a SHA256 : 9bf1e07e1d06c739e924b7ca1bc MD5 : 8f665d117a350a3d26c450 SHA1 : 443540cd3c1eaacb967fc SHA256 : c25505dbc0c17abb8d1bf59e640	Bec2ef5c8c ae14ce2da2d6738d2827 726c2d61a8fe24d4b2dda60515d95 09783eabd5 7242787428f8224f4d3 20c3ee0ca3ad4ccefe3a90480fa95c	52760cad
urwxr x urwxr x urwxr	-xr- /tmp/privshell -xr- /usr/bin/chfn -xr- /usr/bin/chsh	> 8704 5004 4050	2020-01- 08T01:23:45Z 0 2017-05- 17T11:59:59Z 4 2017-05- 17T11:59:59Z	0 0 0	0	MD5 : 040abfbcd907f8cbabdadd SHA1 : 3e6b921491d007d11855a SHA256 : 9bf1e07e1d06c739e924b7ca1bc MD5 : 8f665d117a350a9d26c450 SHA1 : 443540cd3c1eaacb967fc SHA256 : c25505dbc0c17abb8d1bf59e640 MD5 : 2a65613bd111f2dbc9b94d SHA1 : 81ff60c5a9eef7ad1ffc6d0 SHA256 : 7ef5f2f8a8460950fa1a37736c226	Bec2ef5c8c ae14ce2da2d6738d2827 726c2d61a8fe24d4b2dda60515d95 09783eabd5 7242787428f8224f4d3 20c3ee0ca3ad4ccefe3a90480fa95cl 3290ce787a 889c5a10b70c63819 5cdcc435d7bcf85e15ab90830ce445	52200cad b574843 423b3d

Process injection

🗘 atomic-red-team/T10: × 🛛 +

→ C 🔒 github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1055/T1055.md

Atomic Test #3 - Shared Library Injection via /etc/ld.so.preload

This test adds a shared library to the ld.so.preload list to execute and intercept API calls. This te threat actor Rocke during the exploitation of Linux web servers. This requires the glibc package.

Supported Platforms: Linux

Inputs

Name	Description	Туре	Default Value
path_to_shared_library	Path to a shared library object	Path	/bin/T1055.so

Run it with bash ! Elevation Required (e.g. root or admin)

echo #{path_to_shared_library} > /etc/ld.so.preload

Let's test for this! How can we detect it?



Sample code

/*

Atomic Red Team Shared Object Library Uses code inspired by Zombie Ant Farm (https://github.com/dsnezhkov/zombieant)

Compilation

```
gcc -shared -fPIC -o T1055.so T1055.c
```

*/

#include <stdio.h>

```
static void init(int argc, char **argv, char **envp) {
    printf("Loaded Atomic Red Team Library successfully!\n");
}
static void fini(void) {
    printf("Unloading Atomic Red Team preload...\n");
```

__attribute__((section(".init_array"), used)) static typeof(init) *init_p = init; __attribute__((section(".fini_array"), used)) static typeof(fini) *fini_p = fini;

79

Copy/Paste this code and compile it into a shared object

© 2020 Velocidex Enterprises





How can we detect such process injection?

mic@DevBox:~/projects/atomic-red-team/atomics/T1055/src/Linux\$ ls -l Loaded Atomic Red Team Library successfully! total 20 -rw-rw-r-- 1 mic mic 602 Jan 7 15:37 T1055.c -rwxrwxr-x 1 mic mic 16064 Jan 7 15:38 T1055.so mic@DevBox:~/projects/atomic-red-team/atomics/T1055/src/Linux\$

The Linux.Search.FileFinder artifact can be used to search for /etc/ld.so.preload - it should not normally be there



© 2020 Velocidex Enterprises

New Artifact Collection - Sele	ct Artifacts to collect	×
Step 1 out of 2		
Selected Antifacts:	This artifact is useful in the following scenarios:	-
Linux.Search.FileFinder	We need to locate all the places on our network where customer data has been copied.	
	 We've identified malware in a data breach, named using short random strings in specific folders and need to search for other instances across the network. 	
	We believe our user account credentials have been dumped and need to locate them.	
	We need to search for exposed credit card data to satisfy PCI requirements.	
	We have a sample of data that has been disclosed and need to locate other similar files	- 118
Clear	Remove Parameters	- 1
SearchFilesGlob	/etc/ld.so.preload	
Keywords		
Upload_File		
Calculate_Hash		H
MoreRecentThan		- 8
ModifiedBefore		- 1
Ops/Sec		- 18
Maximum Time 600		- U
152		*

2 N N

C 🔒 github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1055/T1055.md

Atomic Test #4 - Shared Library Injection via LD_PRELOAD

This test injects a shared object library via the LD_PRELOAD environment variable to execute. This technique was used by threat actor Rocke during the exploitation of Linux web servers. This requires the glibc package.

Supported Platforms: Linux

Inputs

Name	Description	Туре	Default Value
path_to_shared_library	Path to a shared library object	Path	/opt/AtomicRedTeam/atomics/T1055/bin/T1055.so

Run it with bash !

LD_PRELOAD=#{path_to_shared_library} ls

Let's launch a long lived process

LD_PRELOAD=/tmp/T1055.so bash

Now delete the shared object to make detection even harder!



Sea	rch Box		Q linux-conf-vr	n.c.velocidex-199308.	internal 🥥	connected				0
+	D	@ ♦								
state Flow	/ld	Artif	acts Collected				Creation	Time	Last Active	Creator
✓ F.BO	A7UO6CG4L	G0 Linux	k.Sys.Maps				2020-01-	07 12:53:52 UTC	2020-01-07 12:53:53 UTC	mic
Artifact (Linux.Sy Show 10	Collection ys.Maps	Uploaded Files	s Requests R	Results Log	Reports				Search:	
Pid 👙	Name 🍦	Username	StartHex	EndHex	Perm	♦ Size	SizeHex	Filename		Deleted
10423	bash	scudette	0x7f37630f5000	0x7f37630f6000	rp	0	0x00000000	/tmp/T1055.so		true
10423	bash	scudette	0x7f37630f6000	0x7f37630f7000	r-xp	0	0x00001000	/tmp/T1055.50		true
10423	bash	scudette	0x7f37630f7000	0x7f37630f8000	rp	0	0x00002000	/tmp/T1055.so		true
10423	bash	scudette	0x7f37630f8000	0x7f37630f9000	rp	0	0x00002000	/tmp/T1055.so		true
10423	bash	scudette	0x7f37630f9000	0x7f37630fa000	rw-p	0	0x00003000	/tmp/T1055.so		true

We see that bash is linking a suspicious binary from /tmp which is also deleted!

Note: A binary linking a deleted library is not uncommon - it means the library was upgraded without the binary restarting. It also means the binary is still vulnerable if the library was updated to fix a security issue so it might be interesting to hunt for.





The GUI is limited to show only 500 rows.

For larger result sets prepare and download the zip file containing the collected CSV files and parse them offline.



Scenario: Chrome extensions

Chrome extensions can be very dangerous.

They could access all website data including cookies and login creds.

They can create XSS opportunities for complete compromise.

Exfil is difficult to spot, since all communications occur over SSL.

Many Chrome extensions have been found to be malicious or vulnerable.



chrome Linux.Applications.Cl	hrome.Extensions	Type: client Fetch Chrome extensions.
Linux.Applications.Cl Windows.Application Windows.Application	hrome.Extensions.Upload s.Chrome.Cookies s.Chrome.Extensions	Chrome extensions are installed into the user's home directory. We search for manifest json files in a known path within each system user's home directory. We then parse the manifest file as JSON. Many extensions use locale packs to resolve strings like name and description. In this case we detect the default locale and load those locale files. We then resolve the extension's name and description from there.
Windows.Application	s.Chrome.History	Add Name Type Default
	Cin Unite. Extensions	extensionGlobs Source 1 /* For each user on the system, search for extension manifests 2 in their home directory. */ 3 LET extension manifests
Clear extensionGlobs Ops/See	Remu I.config/g	3 SELECT Vice User, Homedir from Artifact.Linux.Sys.Users() 5 SELECT Uid, User, Homedir from Artifact.Linux.Sys.Users() 6 }, google-chrome/*/Extensions/*/*/manifest.json

	Search Box		Q trek	connected						0 mi
+		2	\$							
State	FlowId		Artifacts Collected				Creation T	ime	Last Active	Creator
X	F.BOAIDNPSG	DVF0	Linux.Applications.C	Chrome.Extensio	ns.Upload		2020-01-08	3 00:48:31 UTC		mic
~	F.BOAID25FF0	IMI	Linux.Applications.C	Chrome.Extensio	ns		2020-01-08	3 00:47:04 UTC	2020-01-08 00:47:07 UTC	mic
~	F.BOAICAG1U	E59O	Linux.Ssh.Authorize	dKeys			2020-01-08	3 00:45:30 UTC	2020-01-08 00:45:31 UTC	H.ca1a7fa
12										
Art	ifact Collection	Uploaded	d Files Request	s Results	Log Reports					
Lin	ux.Applications.	.Chrome.Exte	nsions							v
Sho	w 10 🔻 entrie	S							Search:	
Ui	d 🔺 User 🛔	Name	Description	Identifier		Version	Author	Persistent	Path	
100	0 mic	SSH for Google Cloud Platform	Enriches SSH for Google Cloud Platform, including support for all keyboard shortcuts.	ojillmhjhibplnp	pnamldakhpmdnibd	1.13.1_0		false	/home/mic/.config/google- chrome/Default/Extensions/ojillim	hjhibpInppnamIdał
100	0 mic	TeamViewer	TeamViewer - the All-In-One Software for Remote Support and Home Office	oooiobdokpcfd	llahlmcddobejikcmkfo	14.0.47319_0			/home/mic/.config/google- chrome/Default/Extensions/oooio	bdokpcfdlahlmcdd



Suspicious extensions?

Chrome extensions ask for permissions - some are dangerous!

- videoCapture allows the extension to take videos!
- □ socket allows extensions to make outbound connections!

It is worthwhile auditing chrome extensions in your organization with powerful permissions!

© 2020 Velocidex Enterprises



Search Box	Q trek Connected				0 mic
+ 🛍 🔳 🖄	\$				
State FlowId	Artifacts Collected		Creation Time	Last Active	Creator
F.BOAIDNPSGDVF0	Linux.Applications.Chrome.Extensions.Upload		2020-01-08 00:48:31 UTC	2020-01-08 00:59:51 UTC	mic
✓ F.BOAID25FF0IMI	Linux.Applications.Chrome.Extensions		2020-01-08 00:47:04 UTC	2020-01-08 00:47:07 UTC	mic
✓ F.BOAICAG1UE590	Linux.Ssh.AuthorizedKeys		2020-01-08 00:45:30 UTC	2020-01-08 00:45:31 UTC	H.ca1a7fa6
Artifact Collection Uploaded Linux Applications.Chrome.Exten Show 10 • entries	Files Requests Results Log Reports				Search: videoCapture
Uid User 🔶 Name 🔺	Description Identifier 🗍 Version 🖗 Author 🔶 Persistent 🖗 Path 🗍 Scopes ኞ	Permissions 🍦	Key		
1000 mic Zoom	Coom Cloud Aeetings for Chrome	0 : ['fileSystem": ['write','directory']) 1 : storage 2 : alwaycOnTopWindows 3 : AudioCapture 4 : desktopCapture 4 : desktopCapture 5 : unlimitedStorage 7 : system.cpu 8 : system.memory 9 : system display 10 : power 11 : ['socket']['tcp- connect', "resolve- host", "udp- bind:", "udp-send- to; *** ", "resolve- host", "udp- bind:", "udp-send- to; *** ", "resolve- proxy']} 12 : *//* zoom.us/* 13 : 14 : *//* zoomdev.us/ 15 : ***/* zoom.com	MIIBIJANBgkqhkiG9w0BAQEFAAOCAQ8	AMIIBCgKCAQEAjXFAHp3jJP5VXqg4Zcvhw	v20dD7vDIPB6HGUbrlpQB4sRHWB0PwVKqpllNVuwvd6gRN
Showing 1 to 1 of 1 entries (filtered	from 47 total entries)				Previous 1 Next



Suspicious chrome extension!

So you found a suspicious looking chrome extension?

Upload the extension to the server for closer inspection, using the artifact:

Linux.Applications.Chrome.Extensions.Upload

Warning! This artifact gets all the files in all the extensions this could be huge!

On my system it yields over 400Mb!



e Flowld Artifacts Collected	Creation Time	Last Active	Creato
F.BOAIDNPSGDVF0 Linux.Applications.Chrome.Extensions.Upload	2020-01-08 00:48:31 UTC	2020-01-08 00:59:51 UTC	mic
F.BOAID25FF0IMI Linux.Applications.Chrome.Extensions	2020-01-08 00:47:04 UTC	2020-01-08 00:47:07 UTC	mic
Flow ID F.BOAIDNPSGDVF0 Creator mic	Uploaded Bytes 39252359 Files uploaded 2386	56 / 392523556	500 4
Flow ID F.BOAIDNPSGDVF0	Uploaded Bytes 39252355	56 / 392523556	ra 1
Start Time 2020-01-08 00:48:31 UTC	Download Results Prepare	e Download	
Last Active 2020-01-08 00:59:51 UTC	Available Downloads	Downoad	
State ERROR			
Enter Overstimed out offer 600 seconds			





Search Box	Q trek Oconnected				0 п
+ 🖿 🔳 🤻	b 🔶				
State FlowId	Artifacts Collected		Creation Time	Last Active	Creator
F.BOAIDNPSGDVF0	Linux.Applications.Chrome.Extension	s.Upload	2020-01-08 00:48:31 UTC	2020-01-08 00:59:51 UTC	mic
✓ F.BOAID25FF0IMI	Linux.Applications.Chrome.Extension	S	2020-01-08 00:47:04 UTC	2020-01-08 00:47:07 UTC	mic
Artifact Collection Up	loaded Files Requests Results	Log Reports			
Show 10 • entries				Search:	
Timestamp	time	message			
1578444512145919	2020-01-08 00:48:32 +0000 UTC	vql: scope.go:318: Starting	query execution.		
1578444639194794	2020-01-08 00:50:39 +0000 UTC	Time 36: Linux.Applications	Chrome.Extensions.Upload: Sending respo	onse part 0 522 kB (1001 rows).	
1578444755677138	2020-01-08 00:52:35 +0000 UTC	Time 136: Linux.Application	s.Chrome.Extensions.Upload: Sending resp	oonse part 1 158 kB (299 rows).	
1578444816554226	2020-01-08 00:53:36 +0000 UTC	Time 236: Linux.Application	s.Chrome.Extensions.Upload: Sending resp	oonse part 2 637 B (1 rows).	
1578444945361871	2020-01-08 00:55:45 +0000 UTC	Time 336: Linux.Application	s.Chrome.Extensions.Upload: Sending resp	oonse part 3 639 B (1 rows).	
1578445039980507	2020-01-08 00:57:19 +0000 UTC	Time 437: Linux.Application	s.Chrome.Extensions.Upload: Sending resp	oonse part 4 199 kB (425 rows).	
1578445150907548	2020-01-08 00:59:10 +0000 UTC	Time 537: Linux.Application	s.Chrome.Extensions.Upload: Sending resp	oonse part 5 374 kB (704 rows).	
1578445191100218	2020-01-08 00:59:51 +0000 UTC	vql: scope.go:318: Query tir	med out after 600 seconds		
1578445191102209	2020-01-08 00:59:51 +0000 UTC	While processing job Query	timed out after 600 seconds		
1578445191104489	2020-01-08 00:59:51 +0000 UTC	Time 609: Linux.Application	s.Chrome.Extensions.Upload: Sending resp	oonse part 6 449 kB (854 rows).	

Showing 1 to 10 of 11 entries

Next 1 2

Previous

Velociraptor resource management

- Velociraptor is careful about the resource usage on the endpoint
 - Queries are timed out in 10 minutes by default
 - Queries can be rate limited on the endpoint a notional ops/sec can be specified.
 - ❑ An op is a row or 1mb of scanned data



New Artifact Collection - Select Artifacts to collect

Step 1 out of 2

chrome			ations Chroma Extensions Unload
Linux.Applications.Chrome.Extensions	<u>^</u>	Linux.Applica	ations.Chrome.Extensions.Opioad
Linux.Applications.Chrome.Extensions.Upload		Type: client	
Windows.Applications.Chrome.Cookies		Upload all users chrome e	extension.
Windows.Applications.Chrome.Extensions	_	We dont bother actually pa directory	arsing anything here, we just grab all the extension files in user's home
Windows.Applications.Chrome.History		uncolory.	
	•	Parameters	
Selected Artifacts:	Add	Name T	ype Default
Linux.Applications.Chrome.Extensions.Upload		extensionGlobs	<pre>/.config/google-chrome/*/Extensions/**</pre>
		Source	
Clear	Remove	1 /* For each user 2 in their home 3 SELECT * from for 4 row={ 5 SELECT Vid	r on the system, search for extension files e directory and upload them. */ oreach(, User, Homedir from Artifact.Linux.Sys.Users()
extensionGlobs	/.config/googl	e-chrome/*/Extensions/**	
Ops/Sec 30			
Maximum Time 1200			

Event artifacts and endpoint monitoring



© 2020 Velocidex Enterprises

What are event artifacts?

Event artifacts are never-ending VQL queries that watch for events on clients and stream those events to the server.

Example:

Generic.Client.Stats

Samples CPU and Memory footprint every 10 seconds and streams to the server





Client Footprint for DESKTOP-6CBJ8MJ

8

۲

-

Э

A

The client has a client ID of C.e57080a99511ee58. Clients report the Velociraptor process footprint to the server every 10 seconds. The data includes the total CPU utilization, and the resident memory size used by the client.

The following graph shows the total utilization. Memory utilization is measured in Mb while CPU Utilization is measured by Percent of one core .

We would expect the client to use around 1-5% of one core when idle, but if a heavy hunt is running this might climb substantially.





Monitor all endpoints for successful SSH connections.

As soon as a new SSH session starts, stream an event log to the server.

Now even if the attacker removes local log files we have this important information in a central location.





Search Box

Q

• 🖋 🛍

ssh

Linux.Events.SSHBruteforce

Linux.Events.SSHLogin

Linux.Ssh.AuthorizedKeys

Linux.Ssh.KnownHosts

Linux.Ssh.PrivateKeys

Linux.Syslog.SSHLogin

Linux.Events.SSHLogin

Type: client_event

This monitoring artifact watches the auth.log file for new successful SSH login events and relays them back to the server.

Parameters

Nan	те Тур	e Default
sysle	ogAuthLogPath	/var/log/auth.log
SSH	IGrok	%{SYSLOGTIMESTAMP:timestamp} (?:%{SYSLOGFACILITY})?%{SYSLOGHOST:logs
Soi	urce	
1 2 2	LET success_login FROM watch_syslog	= SELECT grok(grok=SSHGrok, data=Line) AS Event, Line (filename=syslogAuthLogPath)
4	SELECT timestamor	am = sshu AND Event.event Accepted
5	Event.user	AS User,
6	Event.metho	d AS Method,
7	Event.IP AS	SourceIP,
8	Event.pid A	S Pid
9 10	FROM SUCCESS_LOG1	n

Ξ	Search Box Q	linux-conf-vm.c.velocidex-199	9308.internal 🔵 connected			0 mic
*	Linux.Events.SSHLogin -	2020-01-08				٩
¢ 4	Linux.Events.SSHLogin	I				
۲	Show 10 v entries				Search:	
	_ts A Time		User	Method	SourceIP	🗄 Pid 📥
	1578458312 2020-01-0	8T04:37:13Z	scudette	publickey	122.109.172.84	5814
-	1578458512 2020-01-0	8T04:41:43Z	mic	publickey	122.109.172.84	5873
9	Showing 1 to 2 of 2 entries					Previous 1 Next



Process execution logs

Collecting process execution is a very powerful technique
 It is possible to isolate many anomalies, particularly for servers with predictable loads

Examples:

- Apache spawning bash -> execve vulnerability in PHP scripts or webshell
- sshd spawning scp or sftp or rsync -> remote exfiltration of data
- On Linux process execution is logged by the kernel primarily via auditd.
- Velociraptor can parse auditd logs or act as an auditd broadcast receiver by itself.
 - Either way auditd must be installed in order to enable the audit rules



Ξ $\overline{\mathbb{O}}$

Q

execu	Linux, Events, Pro	ocessE	recutions			
Custom.Linux.Sys.SUID	Type: client event					
Demo.Plugins.Fito	This artifact collects process exec	ution logs from	the Linux kernel.			
Linux Events.ProcessExecutions	Deremetere	Ū				
Server Alerts PsExec	Parameters					
Server, Monitor, Shell	Name	Туре	Default			
Server.Powershell.EncodedCommand	posts To Auditot		/usr/sbin/auditctl			
Windows.Analysis.EvidenceOfExecution	path loAuditci					
Windows.Attack.ParentProcess	Courses					
Windows.Attack.Prefetch	Source					
Windows.Forensi Windows.Forensi	1 LET _ = SELECT * FROM 2 "exit,always", "-F" 3 LET exec_log = SELECT	<pre>1 LET _ = SELECT * FROM execve(argv=[pathToAuditctl, "-a", 2 "exit,always", "-F", "arch=b64", "-S", "execve", "-k", "p 3 LET exec_log = SELECT timestamp(string=Timestamp) AS Time, S</pre>				
Windows.Forensics.RecentApps	5 atoi(string=Proces	Process.PID) AS Pid, Process.PPID) AS Ppid,				
Windows.Packs.Autoexec	6 Process.PPID AS PP 7 atoi (string=Summar	'ID, 'y.Actor.Prim	ary) AS UserId,			
Windows.Persistence.Debug	8 Process.Title AS C	mdLine,				
Windows.Persiste Watch for events	10 Process.CWD AS CWD 11 FROM audit())				
Vindows.Registr	12 WHERE "procmon" in Tag	js AND Result	= 'success'			
Windows.Registry.UserAssist	14 FROM Artifact.Linux.Sy	/s.Users()	11g-010, A0 014			
	10 SELECT TIME, PIA PDIA	rom users WH	ERE llid = llserId] AS llser			

ts.ProcessExecutions ocess execution logs from the Linux kernel. Туре Default /usr/sbin/auditctl T * FROM execve(argv=[pathToAuditctl, "-a", ys", "-F", "arch=b64", "-S", "execve", "-k", "procmon"]) = SELECT timestamp(string=Timestamp) AS Time, Sequence, Igenroosse DTD) AS Did ng=Process.PPID) AS Ppid, PID AS PPID, ng=Summary.Actor.Primary) AS UserId, itle AS CmdLine, AC AS ENC, WD AS CWD n" in Tags AND Result = 'success'





C

Client Event N

The Velocirapt

This dashboard

2

Add client monitoring artifacts.

0

Linux.Events.ProcessExecutions	Linux.Events.ProcessExecutions Type: client_event This artifact collects process execution logs from the Linux kernel. Parameters
	Name Type Default
elected Artifacts:	Add pathToAuditctl /usr/sbin/auditctl
Generic.Client.Stats	
Linux.Events.ProcessExecutions	Source
Linux.Events.SSHLogin	5001 00
Clear	<pre>1 LET _ = SELECT * FROM execve(argv=[pathToAuditctl, "-a" 2 "exit,always", "-F", "arch=b64", "-S", "execve", "-k 3 LET exec_log = SELECT timestamp(string=Timestamp) AS Ti 4 atoi(string=Process.PID) AS Pid, 5 atoi(string=Process.PPID) AS Ppid, 6 Process.PPID AS PPID, 7 atoi(string=Summary.Actor.Primary) AS UserId, 8 Process.Title AS CmdLine, 8 Process.Title AS CmdLine, 8 Process.Exe AS Exe 8 Process.Exe AS Exe 8 Process.PETC AS Exe</pre>
	Save Client Monitoring Artifacts



Ŧ

ntries										
							Se	earch:		
Time 👙	Pid	Ppid 🌲	UserId	User	Parent \$	CmdLine	\$	Exe	÷	CWD
2020-01- 08T12:39:27.102Z	13075	10342	1000	mic	SCREEN -T screen-256color -S byobu -c /usr/share/byobu/profiles/byoburc /usr/bin/byobu-shell	SCREEN -T screen-256color -S byobu -c /usr/share/byobu/profiles/byoburc /usr/bin/byobu-shell		/bin/dash		/home/mic/artifac
2020-01- 08T12:39:27.11Z	13076	13075	1000	mic	/bin/sh /usr/bin/byobu-status screen_lower_right	sedfollow-symlinks s/// /dev/null		/bin/sed		/home/mic/artifa/
2020-01- 08T12:39:27.114Z	13078	13075	1000	mic	/bin/sh /usr/bin/byobu-status screen_lower_right	cat /proc/net/dev		/bin/cat		/home/mic/artifa/
2020-01- 08T12:39:27.118Z	13079	13075	1000	mic	/bin/sh /usr/bin/byobu-status screen_lower_right	rm -f /dev/shm/byobu-root- bQWfUSwy/status.screen/network.new		<mark>/bin/r</mark> m		/home/mic/artifa
2020-01- 08T12:39:27.122Z	13080	13075	1000	mic	/bin/sh /usr/bin/byobu-status screen_lower_right	cat /proc/net/dev		/bin/cat		/home/mic/artifa
2020-01- 08T12:39:27.122Z	13081	13075	1000	mic	/bin/sh /usr/bin/byobu-status screen_lower_right	rm -f /dev/shm/byobu-root- bQWfUSwy/status.screen/network*		/bin/rm		/home/mic/artifa
	1020-01- 18712:39:27.102Z 1020-01- 18712:39:27.11Z 1020-01- 18712:39:27.114Z 1020-01- 18712:39:27.118Z 1020-01- 18712:39:27.122Z 1020-01- 18712:39:27.122Z	020-01- 18T12:39:27.102Z 13075 020-01- 18T12:39:27.11Z 13076 020-01- 18T12:39:27.114Z 13078 020-01- 18T12:39:27.114Z 13079 020-01- 18T12:39:27.122Z 13080 020-01- 18T12:39:27.122Z 13081	1020-01- IBT12:39:27.102Z 13075 10342 1020-01- IBT12:39:27.11Z 13076 13075 1020-01- IBT12:39:27.114Z 13078 13075 1020-01- IBT12:39:27.114Z 13078 13075 1020-01- IBT12:39:27.118Z 13079 13075 1020-01- IBT12:39:27.122Z 13080 13075 1020-01- IBT12:39:27.122Z 13081 13075	1020-01- IBT12:39:27.102Z 13075 10342 1000 1020-01- IBT12:39:27.11Z 13076 13075 1000 1020-01- IBT12:39:27.114Z 13078 13075 1000 1020-01- IBT12:39:27.114Z 13078 13075 1000 1020-01- IBT12:39:27.118Z 13079 13075 1000 1020-01- IBT12:39:27.122Z 13080 13075 1000 1020-01- IBT12:39:27.122Z 13081 13075 1000	1020-01- I8T12:39:27.102Z 13075 10342 1000 mic 1020-01- I8T12:39:27.11Z 13076 13075 1000 mic 1020-01- I8T12:39:27.11Z 13076 13075 1000 mic 1020-01- I8T12:39:27.114Z 13078 13075 1000 mic 2020-01- I8T12:39:27.118Z 13079 13075 1000 mic 2020-01- I8T12:39:27.122Z 13080 13075 1000 mic 2020-01- I8T12:39:27.122Z 13081 13075 1000 mic	1020-01- I8T12:39:27.102Z 13075 10342 1000 mic SCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byoburc /usr/bin/byobu-shell 1020-01- I8T12:39:27.11Z 13076 13075 1000 mic /bin/sh /usr/bin/byobu-shell 1020-01- I8T12:39:27.11Z 13076 13075 1000 mic /bin/sh /usr/bin/byobu-status screen_lower_right 2020-01- I8T12:39:27.114Z 13078 13075 1000 mic /bin/sh /usr/bin/byobu-status screen_lower_right 2020-01- I8T12:39:27.118Z 13079 13075 1000 mic /bin/sh /usr/bin/byobu-status screen_lower_right 2020-01- I8T12:39:27.122Z 13080 13075 1000 mic /bin/sh /usr/bin/byobu-status screen_lower_right 2020-01- I8T12:39:27.122Z 13081 13075 1000 mic /bin/sh /usr/bin/byobu-status screen_lower_right 2020-01- I8T12:39:27.122Z 13081 13075 1000 mic /bin/sh /usr/bin/byobu-status screen_lower_right	SCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ screen_lower_rightSCREEN -T screen-256color -S vusr/share/byobu/profiles/byobu/c /usr/share/byobu/c /usr/share/byobu/profiles/byobu/c /usr/share/byobu/c <b< td=""><td>SCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ screen_lower_rightSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ screen_lower_rightSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ screen_lower_right2020-01- 18712:39:27.11Z13076130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightsedfollow-symlinks s/// /dev/null2020-01- 18712:39:27.11AZ13078130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.11BZ13079130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13080130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightrm -f /dev/shm/byobu-root- bQWtUSwy/status.screen/network</td><td>SCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byoburSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byoburSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byoburMin/dash1020-01- 18T12:39:27.11Z13076130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightsedfollow-symlinks s// /dev/null//bin/sed1020-01- 18T12:39:27.11Z13078130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightsedfollow-symlinks s// /dev/null//bin/sed1020-01- 18T12:39:27.114Z13078130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/sh2020-01- 18T12:39:27.118Z13079130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/m2020-01- 18T12:39:27.122Z13080130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/cat2020-01- 18T12:39:27.122Z13081130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/cat2020-01- 18T12:39:27.122Z13081130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/m2020-01- 18T12:39:27.122Z13081130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightmid /dev/shm/byobu-root- bQWHUSwy/status.screen/network*//bin/m</td><td>SCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/bin/byobu-shellSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobur /usr/bin/byobu-shellSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobur /usr/bin/byobu-shellMin/dash1020-01- 18T12:39:27.11Z13076130751000mic/bin/sh /usr/bin/byobu-shellsedfollow-symlinks s/// /dev/null/bin/sed2020-01- 18T12:39:27.11Z13078130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightsedfollow-symlinks s/// /dev/null/bin/cat2020-01- 18T12:39:27.11AZ13079130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/fm2020-01- 18T12:39:27.11BZ13080130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/fm2020-01- 18T12:39:27.12ZZ13080130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/cat2020-01- 18T12:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/cat2020-01- 18T12:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/cat2020-01- 18T12:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightmir / /dev/shm/byobu-root- bQW/USwy/status.screen/network*/bin</td></b<>	SCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ screen_lower_rightSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ screen_lower_rightSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/share/byobu/profiles/byobu/ screen_lower_right2020-01- 18712:39:27.11Z13076130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightsedfollow-symlinks s/// /dev/null2020-01- 18712:39:27.11AZ13078130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.11BZ13079130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13080130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev2020-01- 18712:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightrm -f /dev/shm/byobu-root- bQWtUSwy/status.screen/network	SCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byoburSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byoburSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byoburMin/dash1020-01- 18T12:39:27.11Z13076130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightsedfollow-symlinks s// /dev/null//bin/sed1020-01- 18T12:39:27.11Z13078130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightsedfollow-symlinks s// /dev/null//bin/sed1020-01- 18T12:39:27.114Z13078130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/sh2020-01- 18T12:39:27.118Z13079130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/m2020-01- 18T12:39:27.122Z13080130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/cat2020-01- 18T12:39:27.122Z13081130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/cat2020-01- 18T12:39:27.122Z13081130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev//bin/m2020-01- 18T12:39:27.122Z13081130751000mic//bin/sh /usr/bin/byobu-status screen_lower_rightmid /dev/shm/byobu-root- bQWHUSwy/status.screen/network*//bin/m	SCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobu/ /usr/bin/byobu-shellSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobur /usr/bin/byobu-shellSCREEN -T screen-256color -S byobu - c /usr/share/byobu/profiles/byobur /usr/bin/byobu-shellMin/dash1020-01- 18T12:39:27.11Z13076130751000mic/bin/sh /usr/bin/byobu-shellsedfollow-symlinks s/// /dev/null/bin/sed2020-01- 18T12:39:27.11Z13078130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightsedfollow-symlinks s/// /dev/null/bin/cat2020-01- 18T12:39:27.11AZ13079130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/fm2020-01- 18T12:39:27.11BZ13080130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/fm2020-01- 18T12:39:27.12ZZ13080130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/cat2020-01- 18T12:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/cat2020-01- 18T12:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightcat /proc/net/dev/bin/cat2020-01- 18T12:39:27.12ZZ13081130751000mic/bin/sh /usr/bin/byobu-status screen_lower_rightmir / /dev/shm/byobu-root- bQW/USwy/status.screen/network*/bin

Velociraptor simply reports process execution logs from the endpoint to the server

Advanced topics

- We can now forward these events to a SIEM like Elasticsearch
- We can write server monitoring artifacts to monitor the stream of events for bad patterns and alert/escalate further.



Apply what you learned

Investigations involve answering questions about our endpoints.

Velociraptor enhances your visibility into the state of your endpoints answering those questions effortlessly and quickly.

What questions do you want to answer about your network?



105

Dig Deeper!

- □ Today's workshop is an introduction level.
- Velociraptor's power comes from its open architecture and powerful query language.
- Learning how to write your own queries and integrate with Velociraptor's API opens huge possibilities for customization to your specific requirements!

See the documentation site <u>Velociraptor Docs</u>

Further training opportunities <u>Training Schedule</u>



Start hunting today

- Download Velociraptor from our GitHub repository.
- □ Join the mailing list at <u>velociraptor-discuss@googlegroups.com</u>
- Join our Discord Channel at https://www.velocidex.com/discord
- Leave feedback at http://feedback.velocidex.com/
- Contribute back with your feedback and ideas.



© 2020 Velocidex Enterprises