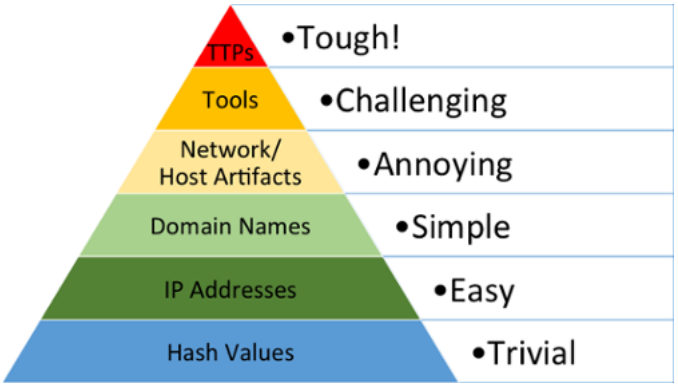
A Framework for Cyber Threat Hunting Part 3: The Value of Hunting TTPs

September 24, 2015 by
Sqrrl Team

A Framework for Cyber Threat Hunting Part 3: The Value of Hunting TTPs

In the first two parts of our "Framework for Cyber Threat Hunting" series, we discussed the heirarchy of Indicators of Compromise , the most valuable of which are an attacker's Tactics, Techniques, and Procedures (TTPs), and the benefits of using those indicators in a security feedback loop to build an Advanced Persistent Defense . This third and final part aims to provide a concrete example of how the discovery and mapping of TTPs contributes to the strength of an advanced persistent defense.

An attacker's TTPs are the actions they make as they move down the Kill Chain . Constructing the narrative of an attack, from Reconnaissance to Act on Objective, provides actionable insights into two areas. First, mapping an attacker's TTPs can be useful for attributing an attack to an actor. Given that many actors employ the same (or related) techniques in each of their attacks, attributing an attack to a specific actor helps expedite attribution and make response/remediation more effective. For example, a certain adversary group might tend to use a specific kind of spearphishing attack as their tactic, or always employ a specific kind of malware as their procedure. This makes it easier to identify their presence in the future when these TTPs are identified again. Second, understanding each of the attacker's movements down the Kill Chain makes it easier for security analysts to create detections for those TTPs, rather than always have to respond to attacks that have already happened. Taking advantage of these indicators as you gather and feed them back into your security ecosystem is a fundamental part of building an advanced persistent defense model .



TTPs are at the top of the Pyramid of Pain… The hardest indicators to get, but also the most useful. The Pyramid of Pain, originally developed by David Bianco: http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

Let's examine one example of an adversary's TTPs in an attack:

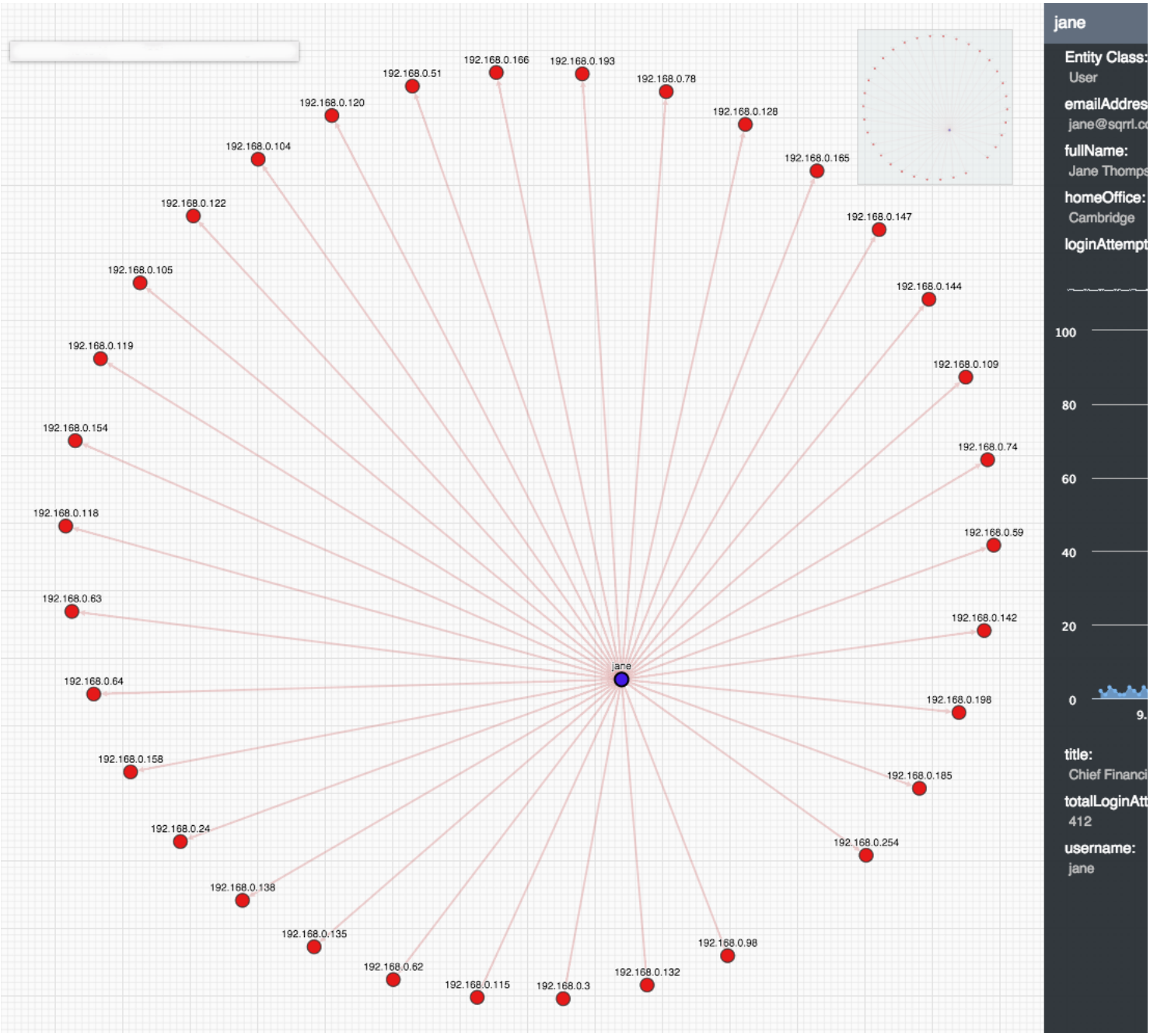
- At our example company, an attacker finds our CFO, Jane, via LinkedIn and constructs a spearphishing email that tricks Jane into downloading malware onto her laptop.
- Accessing the laptop when Jane returns to the company's network, the attacker scans the network for servers that can be accessed using Jane's credentials.
- After a number of failed attempts to log into various system, the attacker realizes that they will need higher level credentials to access servers containing customer billing histories. Using access to Jane's account, the hacker installs a piece of malware that, when IT services her machine, scrapes the hash of the "IT administrator" credentials.
- Passing the hash of the admin credentials, the hacker accesses the critical servers and exports several thousand billing histories back to Jane's computer. This avoids automated alerts that indicate data being sent from the servers to an outside source.
- Once the files have been moved to Jane's computer, they are exfiltrated to a remote server outside the network through email or other transactions which might fit her regular behavior.

So, How Do We Detect TTPs?

By definition, the TTPs of an attacker are composed of events and tools that make up the narrative of the attack. For that reason, detecting a TTP will almost always require a combination of automated detection systems and human analysts. Discovery and detection of a previously unknown TTP will always start with an exploratory hunt (a hunt based on hypothesis) or a more basic indicator.

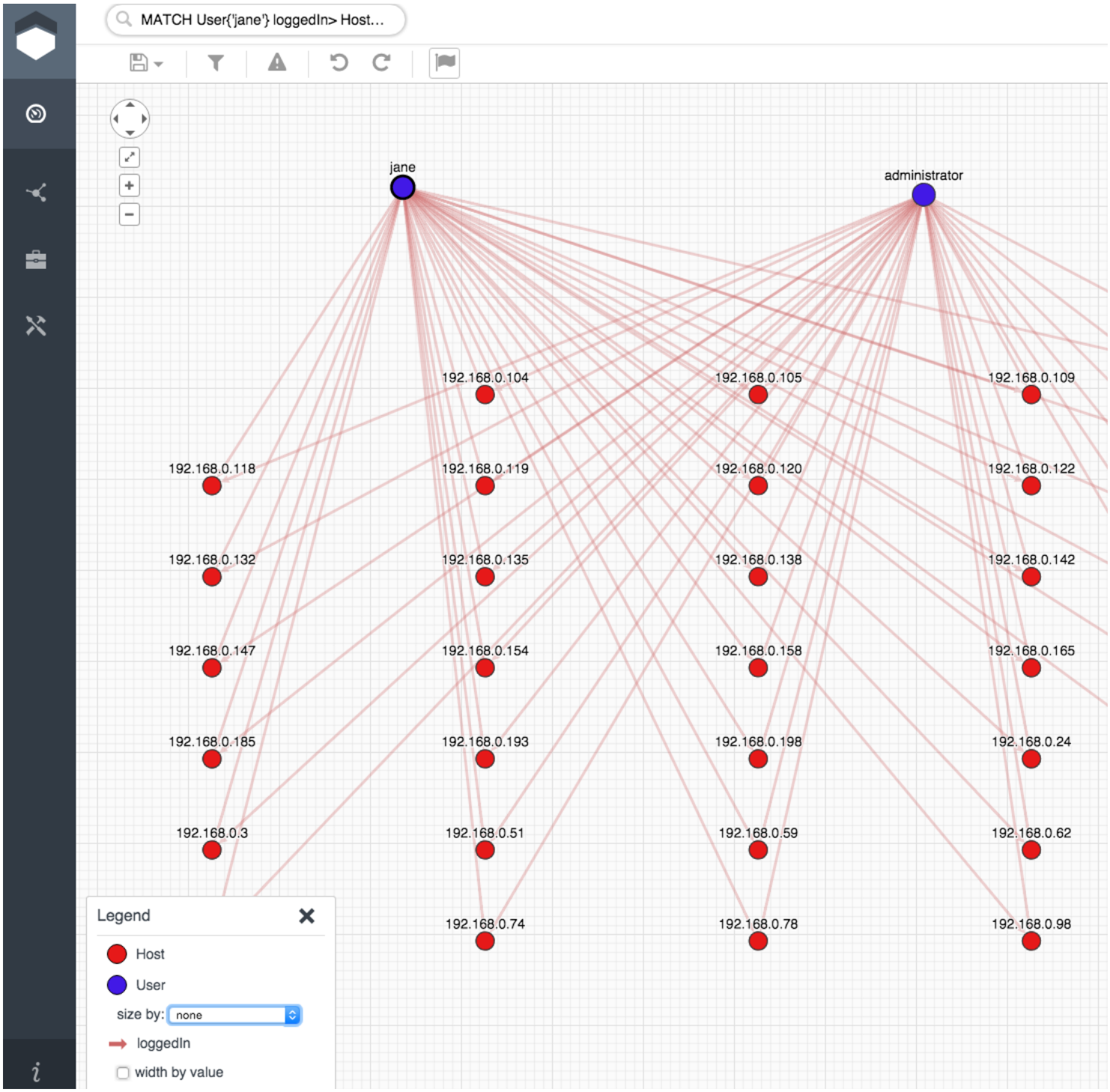
In this case, an analyst could have suspected the attack because an alert flagged an internal machine sending some amount of data to a known malicious domain. Alternatively, while investigating activity involving VIP accounts, a threat hunter might have noticed anomalies in the number and frequency of Jane's login attempts, deviating from her normal behavior patterns. If Jane's login attempts are recorded in a mixture of host logs or VPN logs, mapping the TTPs would necessitate pivoting across different security datasets to track different phases of the Kill Chain.

In log based models, any analyst who is threat hunting through the environment must manually query each dataset and rely on event correlation engines to notice anomalies. This makes it very challenging to make connections across different datasets. Pivoting through more than one or two layers is very cumbersome, and it's difficult to keep track of where you are and where you have already been in the process. However, in a linked data model, every entity is visually connected to other entities that relate to it, which gives hunters context for each event, helps track their investigative progress and allows them to easily build a picture of the attack narrative. In the case of Jane's compromised laptop, fully mapping out the TTPs of this attacker will require we pivot across at least four datasets, including email and authentication logs.

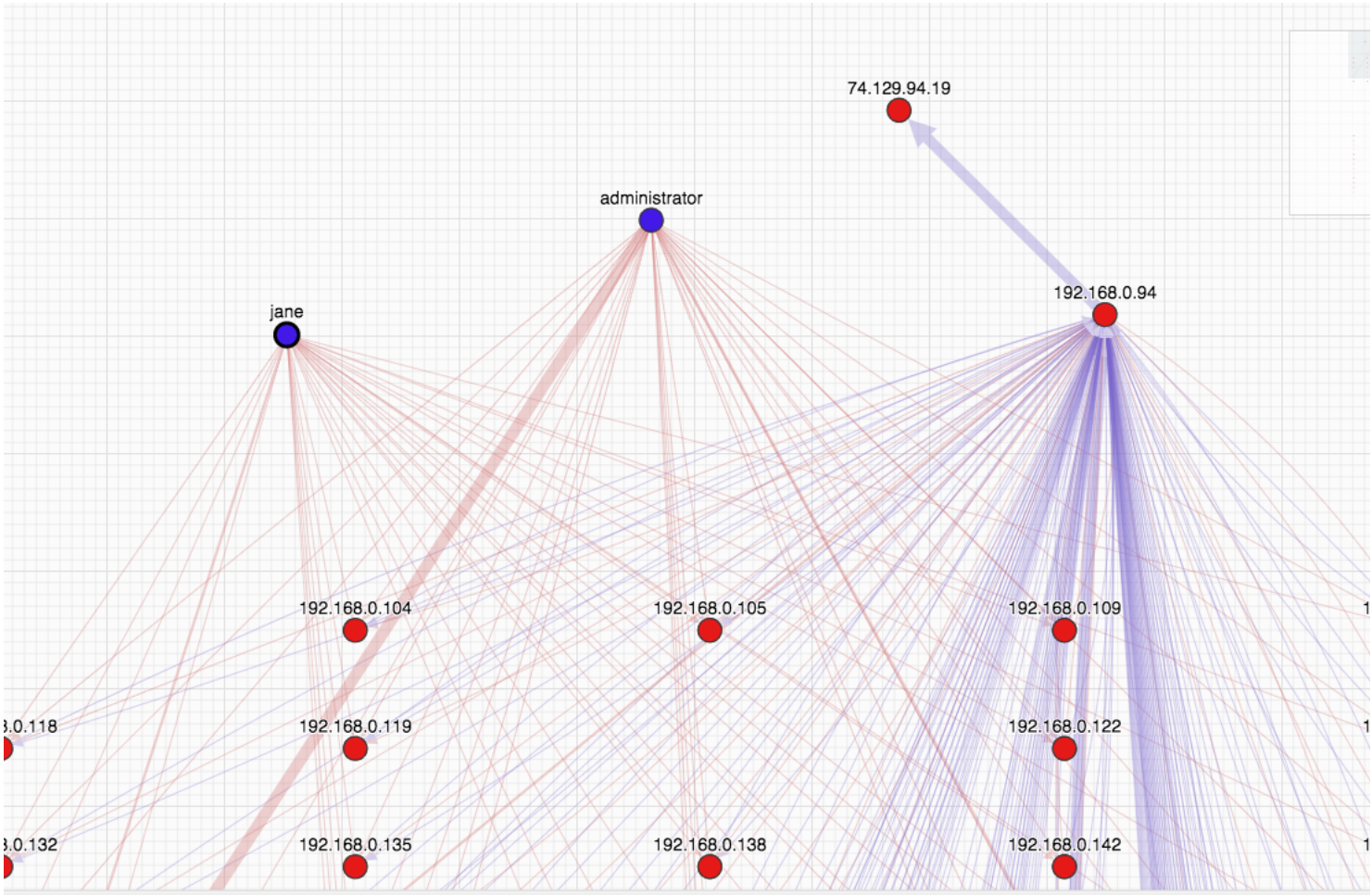


Initially, an analyst might discover something anomalous, like an unusually high number of logins for an unlikely user

For example, after constructing a timeline of Windows Security Event logs, it is easier to identify anomalous behavior, which can then be correlated with the HR profile of the anomalous user, in this case, Jane. Netflow data from our Firewall linked to the login information shows that the "IT administrator" attempted to access a large subset of the machines Jane tried to login to.



Jane and the Admin have accessed many of the same hosts



entries

	TimeCreated	IpAddress	AuthenticationPackag	Computer	EventId	IpPort	KeyLength	LmPacka
:3...	2014-09-10T04:...	192.168.0.94	NTLM	192.168.0.98	4624	40937	128	NTLM V2
:3...	2014-09-10T04:...	192.168.0.94	NTLM	192.168.0.78	4624	57981	128	NTLM V2
:3...	2014-09-10T04:...	192.168.0.94	NTLM	192.168.0.74	4624	39182	128	NTLM V2
:3...	2014-09-10T04:...	192.168.0.94	NTLM	192.168.0.62	4624	38031	128	NTLM V2
:3...	2014-09-10T04:...	192.168.0.94	NTLM	192.168.0.64	4624	60018	128	NTLM V2

Reconstructing the attackers exfiltration process: From Jane to the admin, to a host on the network, and then out to a remote IP

How Do we Use These TTPs?

There are a number of suspicious behaviors embedded in the data from the attack scenario above. These behaviors help define the TTPs of the attacker. Some examples of the observables associated with these behaviors include the following:

- A high number of login attempts, especially an anomalous number that deviates significantly from an established historical baseline or from that user's established peer group
- Machines that initiate scans often indicate internal reconnaissance
- Data transfers from a server to a user or group of machines that do not normally receive large data transfers
- Data transfers from a server to an internal machine then to an external machine

Sqrrl Enterprise helps analysts identify and make use of these behaviors in a number of ways.

- Sqrrl's integration with Apache Spark enables usage of machine learning algorithms to auto-detect malicious behaviors
- Analysts can set up dashboards that correspond to the observable behaviors associated with TTPs.
- Using Sqrrl's replay feature, analysts can share the discovery of the TTP with their team or administrator, step by step, to decrease Mean Time to Know in the future.

- Having mapped the TTP, analysts can capture the results of their hunt as indicators to facilitate automated defense

These are some of the ways that you can develop your security ecosystem to be more resiliant and adaptable, with hunting as one of the core aspects of your cyber defense efforts. If you haven't yet, make sure to check out part 1 and part 2 of this blog series, to see how everything fits together.