The ultimate Authentication Brute-Force detection using super stats

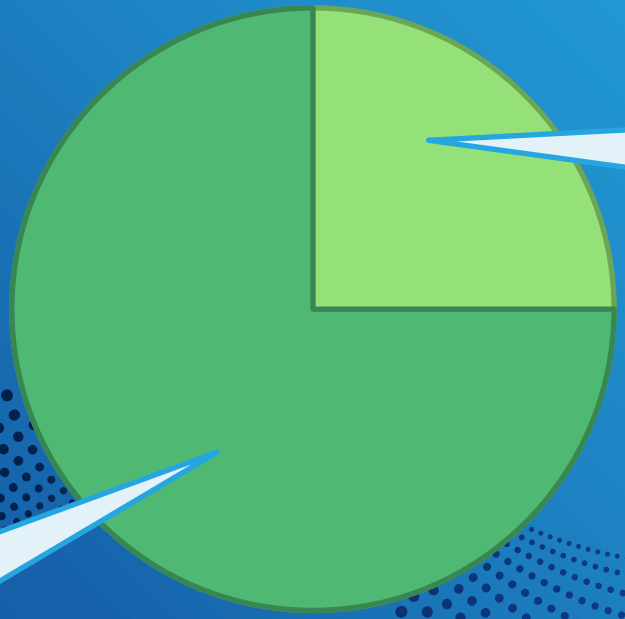
| stats values(*)

Alex Teixeira

- Over 15y of professional experience in Infosec
- Former **Splunker** (Top **PS** Sales Contributor FY'16)
- Last 4y *solely* dedicated to writing SPL (detection use cases) for enterprise Security Teams as a freelancer
- Recently joined an MDR provider based in NY as a Senior Content Engineer (SPL, KQL, Detection Research)

Why this?

Problem Statement



Because *virtually* every computer system is a potential target of brute-force attacks

Because SPL is awesome, and I'd like to share how I use it!

MITRE's ATT&CK **T1110**

- Password Guessing (.001)
- Password Spraying (.003)
- ~~Password Cracking~~
- ~~Credential Stuffing~~

Brute Force

Sub-techniques (4)

Adversaries may use brute force techniques to gain access to accounts when passwords are unknown or when password hashes are obtained. Without knowledge of the password for an account or set of accounts, an adversary may systematically guess the password using a repetitive or iterative mechanism. Brute forcing passwords can take place via interaction with a service that will check the validity of those credentials or offline against previously acquired credential data, such as password hashes.

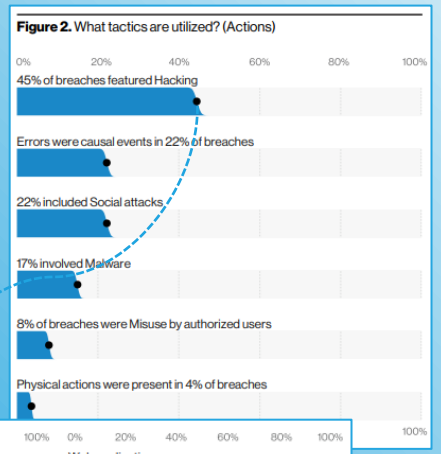
MITRE | ATT&CK

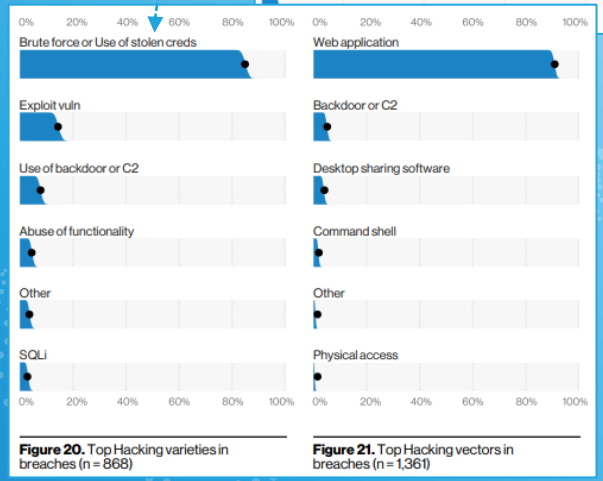
https://attack.mitre.org/techniques/T1110/

“

However, it must be said that *Hacking* and even breaches in general are driven by credential theft. Over 80% of breaches within *Hacking* involve **Brute Force** or the Use of lost or stolen credentials.

https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf





What makes it a *potential* attack?

1. How many accounts are targeted?
2. What's the attack timespan?
3. How many origins are involved in the attack?
4. What's the # of attempts per target account?
5. Was there a successful authentication involved?

Data Sources

RAW (majority)

- VPN logs
- Eventlogs
- SignIn/Cloud logs
- Nix/Net devices

Data Model (some)

- Splunk's CIM
- tag=authetication
- action field
- Acceleration

Custom (few)

- Summary index
- Tstats-enabled index via selective 'IFX'

- **Origin** (src host/IP)
- **Target** (user acc + [dest host/IP])
- **Action** (success/failure)

Relevant fields

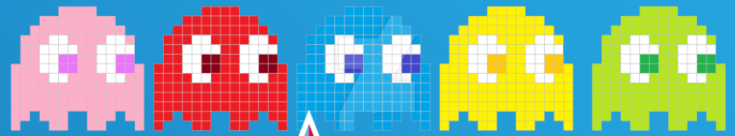
O365, AD auth differ from local one!

Brute-Force Attack types

Type \ Attributes	Trigger Condition (Threshold)	Scenario	Footprint
Account Brute-Force	- Single account, single origin - More than *A* (attempts per target) within short *T* (timespan) from same origin towards the same account	Vanilla attack, too many attempts against the same target from the same origin.	Medium
Mass Account Brute-Force	Multiple instances of above from same origin, affecting too many accounts	Noisy version of above (ex.: Internet Scanners, Bots, etc).	High
Password Spray	- Multiple accounts, single origin - More than *U* (account targets) from same origin are seen - **Avg # of attempts per target is <= *A***	Detects attacks below vanilla threshold towards multiple targets, even if they are 'low and slow'.	Vary
Targeted Account Brute-Force	- Single account, multiple origins - More than *O* (origins) observed towards same target account – **not matching a default one**	Multiple origins targeting a single, non-default. Default includes *administrator*, *admin*, *root*, *guest*, etc.	Low

Overlapping possible, alert aggretaion makes a big difference. More on that later.

Demo Dataset



Multiple BF attack instances and types, easily customizable via simple SPL commands

```
| makeresults ``` all events within 1h (demo rule detection interval) ```
| where 0=1 ``` filters out empty record ```

| append [ | makeresults count=3 | eval _time=relative_time(now(), "-0s"), src="pokey", signature="User failed to authenticate", action="failure", user="admin" ] ``` vanilla attack```
| append [ | makeresults count=3 | eval _time=relative_time(now(), "-1s"), src="pokey", signature="User failed to authenticate", action="failure", user="admin" ]
| append [ | makeresults count=6 | eval _time=relative_time(now(), "-100s"), src="blinky", signature="User failed to authenticate", action="failure", user="backup" ] ``` successful attack```
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-99s"), src="blinky", signature="User login successful", action="success", user="backup" ]

``` noisy attack```
| append [ | makeresults count=8 | eval _time=relative_time(now(), "-10min"), src="speedy", signature="User failed to authenticate", action="failure", user=mvappend("admin", "root", "test", "administrator", "guest"
``` fast targeted attack```
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-30min"), user="princesspeach", signature="User failed to authenticate", action="failure", src=mvappend("bird", "jordan", "johnson", "pippen", "ro
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-50min"), user="larrykoopa", signature="User failed to authenticate", action="failure", src=mvappend("bird", "jordan", "johnson", "pippen", "robin
``` slow targeted attack```
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-10min"), user="donkeykongjr", signature="User login successful", action="success", src="bird" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-15min"), user="donkeykongjr", signature="User failed to authenticate", action="failure", src="jordan" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-20min"), user="donkeykongjr", signature="User failed to authenticate", action="failure", src="johnson" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-25min"), user="donkeykongjr", signature="User failed to authenticate", action="failure", src="pippen" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-30min"), user="donkeykongjr", signature="User failed to authenticate", action="failure", src="robinson" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-35min"), user="donkeykongjr", signature="User failed to authenticate", action="failure", src="clyde" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-40min"), user="donkeykongjr", signature="User failed to authenticate", action="failure", src="malone" ]

``` fast password spray ```
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-55min"), src="shadow", signature="User failed to authenticate", action="failure", user=mvappend("admin", "root", "test", "administrator", "guest"
``` slow password attack```
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-10min"), src="pinky", signature="User login successful", action="success", user="admin" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-15min"), src="pinky", signature="User failed to authenticate", action="failure", user="root" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-20min"), src="pinky", signature="User failed to authenticate", action="failure", user="test" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-25min"), src="pinky", signature="User failed to authenticate", action="failure", user="administrator" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-30min"), src="pinky", signature="User failed to authenticate", action="failure", user="guest" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-35min"), src="pinky", signature="User failed to authenticate", action="failure", user="info" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-40min"), src="pinky", signature="User failed to authenticate", action="failure", user="contact" ]

``` random noise/legit sessions ```
| append [ | makeresults count=4 | eval _time=relative_time(now(), "-6s"), src="workstation10", signature="User failed to authenticate", action="failure", user="administrator" ] ``` 4 failed, then success (FP) ```
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-0s"), src="workstation10", signature="User login successful", action="success", user="administrator" ]
| append [ | makeresults count=3 | eval _time=relative_time(now(), "-1s"), src="speedy", signature="User failed to authenticate", action="failure", user="backup" ] ``` speedy below threshold (FP) ```
| append [ | makeresults count=3 | eval _time=relative_time(now(), "-10s"), src="laptop1", signature="User failed to authenticate", action="failure", user="root" ] ``` legit/noise (FP) ```
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-0s"), src="laptop1", signature="User login successful", action="success", user="root" ]
| append [ | makeresults count=1 | eval _time=relative_time(now(), "-0s"), src="laptop2", signature="User login successful", action="success", user="admin" ]
```

All events within 1h span

Some noise/legit events

| collect index="auth"

The ‘high-level’ plot

Attack Type	Origin (src)	Target (user)
Account Brute-Force (BF)	- Pokey	- admin
	- Speedy	- multiple default accounts
***Successful* Account BF**	- Blinky	- backup
Mass Account BF	- Speedy	- multiple default accounts
(Fast) Password Spray	- Shadow	- multiple default accounts
(Slow) *Successful* Password Spray	- Pinky	- multiple default accounts
(Fast, Slow & *Successful*) Targeted Account BF	- NBA Stars	- multiple Mario All Stars
Stealth BF Attack	?	?

Dataset Summary

per *src* (origin)

Total events: 123

src	sparkline	count	start_time	end_time	user	action	timespan
bird		3	17:50:21	18:30:21	donkeykongjr larrykoopa princesspeach	failure success	2400
blinky		7	18:38:41	18:38:42	backup	failure success	1
clyde		3	17:50:21	18:10:21	donkeykongjr larrykoopa princesspeach	failure	1200
		7	18:40:07		buttercup	failure success	6
johnson		3	17:50:21	18:20:21	donkeykongjr larrykoopa princesspeach	failure	1800
jordan		3	17:50:21	18:25:21	donkeykongjr larrykoopa princesspeach	failure	2100
laptop1		4	18:40:11	18:40:21	root	failure success	10
laptop2		1	18:40:21	18:40:21	admin	success	0
malone		2	18:00:21	18:10:21	donkeykongjr princesspeach	failure	600
pinky		7	18:00:21	18:30:21	admin administrator contact guest info root test	failure success	1800
pippen		3	17:50:21	18:15:21	donkeykongjr larrykoopa princesspeach	failure	1500
pokey		6	18:40:20	18:40:21	admin	failure	1
robinson		3	17:50:21	18:10:21	donkeykongjr larrykoopa princesspeach	failure	1200
shadow		7	17:45:21	17:45:21	admin administrator contact guest info root test	failure	0
speedy		59	18:30:21	18:40:20	admin administrator backup contact guest info root test	failure	599
workstation10		5	18:40:15	18:40:21	administrator	failure	6

Total events per *src*

Targets

"Low and slow" attack

High Footprint (Mass BF)

BSIDES splunk>

Detection Strategy

- Run every 1h and leverage multiple ***stats**
- For fast: **10s** window, other: rule's
- Aggregate alerts by "attack flows"
  - Faster alert triage
  - Focus on a bigger picture (attacker/target focus)
  - Leverage Data Analytics

Pseudo-Code

```
Collect events

Bin (_time, 10s)

Stats by origin and target
  Detect Vanilla attack count(attempts)

Eventstats by origin
  Detect Mass attack count(vanilla attacks)
  Detect Password Spray dc(targets)

Eventstats by target
  Detect Targeted attack count(origins)

Filter in attacks (signature!=null)

Stats by Attack Flow
```

Crafting the query: base search

Scope in potential data sources

Map ***origin*** and ***target*** field names

Define what constitutes a success/failure auth attempt

Define the timespan for fast attacks and name is ***tspan***

```
index=auth (action=* OR vendor_action=*)

| eval origin=src    ``` origin=src_user (for su/sudo/rdp/local/os attacks) ```
| eval target=user   ``` target=user."@".dest (in case target comprises of user + dest host) ```

| eval failure_time=if(match(coalesce(action, vendor_action), "(?i)(fail)"), _time, null())
| eval success_time=if(match(coalesce(action, vendor_action), "(?i)(success)"), _time, null())

| bin span=10s _time AS tspan

``` vanilla BF attack metrics calculated over each time window (tspan), split by origin and target ```
| stats min(_time) AS start_time, max(_time) AS end_time, count(failure_time) AS count_fail, count(success_time) AS count_success
  BY origin, target, tspan
```

Modify this to your environment and you're good to go!

origin	target	tspan	start_time	end_time	count_fail	count_success
bird	donkeykongjr	1618158620	1618158621	1618158621	0	1
bird	larrykoopa	1618156220	1618156221	1618156221	1	0
bird	princesspeach	1618157420	1618157421	1618157421	1	0
blinky	backup	1618159120	1618159121	1618159122	6	1
clyde	donkeykongjr	1618157120	1618157121	1618157121	1	0
clyde	larrykoopa	1618156220	1618156221	1618156221	1	0
clyde	princesspeach	1618157420	1618157421	1618157421	1	0
```

Crafting the query: **Vanilla** Attack

```
| stats min(_time) AS start_time, max(_time) AS end_time, count(failure_time) AS count_fail, count
    (success_time) AS count_success
  BY origin, target, tspan

| eval default_target=if(match(target, "^(?i)(administrator|admin|root|guest|test|backup)(@\S+)*$"), 1, 0)

| eval attack_th_global=5 ``` count_fail>attack_th_global within tspan => Brute-Force ```
| eval attack_span=(1+end_time-start_time)
| eval attack_rate=ceil((count_fail+count_success)/attack_span)
| eval signature=case(
  count_fail>attack_th_global AND count_success>0, "Potential Successful Brute-Force Attack",
  count_fail>attack_th_global, "Potential Brute-Force Attack")
| eval reason=if(isnotnull(signature), signature.": There were [".count_fail."] failed attempts and ["
    .count_success."] successful login(s) observed from origin [".origin."] towards [".target."] over "
    .attack_span." second(s) between ".strftime(start_time,"%F %T")." and ".strftime(end_time,"%F %T").".
    Rate: ~".attack_rate." attempts/s.", null())
```

Flag when target is likely a default account (used later)

Attack description, including origin, target, # of attempts, etc.

BF Attack conditions

origin	target	start_time	end_time	count_fail	count_success	attack_rate	attack_span	attack_th_global	default_target	reason	signature
bird	larrykoopa	1618156221	1618156221	1	0	1	1	5	0		
blinky	backup	1618159121	1618159122	6	1	4	2	5	1	Potential Successful Brute-Force Attack: There were [6] failed attempts and [1] successful login(s) observed from origin [blinky] towards [backup] over 2 second(s) between 2021-04-11 18:38:41 and 2021-04-11 18:38:42. Rate: ~4 attempts/s.	Potential Successful Brute-Force Attack
laptop1	root	1618159211	1618159211	3	0	3	1	5	1		
laptop1	root	1618159221	1618159221	0	1	1	1	5	1		
laptop2	admin	1618159221	1618159221	0	1	1	1	5	1		
speedy	admin	1618158621	1618158621	8	0	8	1	5	1	Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [admin] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s.	Potential Brute-Force Attack
speedy	contact	1618158621	1618158621	8	0	8	1	5	0	Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [contact] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s.	Potential Brute-Force Attack

Legit instances

"speedy" is involved in multiple attacks, against many accounts

BSIDES splunk>

Crafting the query: **Mass & Spray**

Flag targets

Mass BF conditions

Password Spray conditions

```
| eval bf_target=if(isnotnull(signature), target, null())  ``` to be used within the next iteration stats on Mass Brute-Force attack ```
| eval attack_target=if(count_fail>0, target, null())       ``` to be used within the next iteration stats on Password Spray attack ```

```  from here on, metrics are aggregated by [origin], signature & reason (signals) needs to be accumulated and carried over via mvappend() ```
| eventstats min(start_time) AS start_time_by_origin, max(end_time) AS end_time_by_origin, sum(count_success) AS count_success_sum, sum(count_fail) AS count_fail_sum,
    avg(count_fail) AS count_fail_avg, dc(bf_target) AS count_bf_target, dc(attack_target) AS count_attack_target, values(target) AS target_values
  BY origin

| eval attack_th=4  ``` count_bf_target>attack_th => Mass Brute-Force ```
| eval attack_span=(1+end_time_by_origin-start_time_by_origin)
| eval attack_flag=if(count_bf_target>attack_th, 1, 0)
| eval signature=if(attack_flag=1, mvappend("Mass Brute-Force Attack", signature), signature)
| eval reason=if(attack_flag=1, mvappend("Mass Brute-Force Attack: more than ".attack_th." brute-force targets observed from same origin [".origin."].", reason),
    reason)

| eval attack_th=5  ``` count_attack_target>attack_th + count_fail_avg<=attack_th_global => Password Spray ```
| eval attack_flag=case(
  count_attack_target>attack_th AND count_fail_avg<=attack_th_global AND count_success_sum>0, "Potential Successful",
  count_attack_target>attack_th AND count_fail_avg<=attack_th_global, "Potential",
  1=1, null())
| eval signature=if(isnotnull(attack_flag), mvappend(attack_flag." Password Spray Attack", signature), signature)
| eval reason=if(isnotnull(attack_flag), mvappend(attack_flag." Password Spray Attack: there were [".count_fail_sum."] failed attempts (~".ceil(count_fail_avg)."
    /attacked target) and [".count_success_sum."] successful login(s) observed from origin [".origin."] towards [".count_attack_target."] targets [".mvjoin
    (target_values, ", ")."] over ".attack_span." second(s) between ".strftime(start_time_by_origin,"%F %T")." and ".strftime(end_time_by_origin,"%F %T")."., reason),
    reason)
```

origin	target	count_attack_target	count_bf_target	count_fail_avg	count_fail_sum	count_success_sum	signature	reason
pinky	admin	6	0	0.8571428571428571	6	1	Potential Successful Password Spray Attack	Potential Successful Password Spray Attack: there were [6] failed attempts (~1/attacked target) and [1] successful login(s) observed from origin [pinky] towards [6] targets [admin, administrator, contact, guest, info, root, test] over 1801 second(s) between 2021-04-11 18:00:21 and 2021-04-11 18:30:21.
shadow	admin	7	0	1	7	0	Potential Password Spray Attack	Potential Password Spray Attack: there were [7] failed attempts (~1/attacked target) and [0] successful login(s) observed from origin [shadow] towards [7] targets [admin, administrator, contact, guest, info, root, test] over 1 second(s) between 2021-04-11 17:45:21 and 2021-04-11 17:45:21.
speedy	admin	8	7	7.375	59	0	Mass Brute-Force Attack Potential Brute-Force Attack	Mass Brute-Force Attack: more than 4 brute-force targets observed from same origin [speedy]. Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [admin] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s.

"speedy" triggers the threshold for "Mass Attack"

Average # of failed attempts would never trigger vanilla detection!

BSIDES splunk>
```

Crafting the query: **Targeted** attack

Flag origins

Targeted Attack conditions

```
| eval attack_origin=if(default_target=0 AND count_fail>0, origin, null())  ``` to be used within the next iteration stats on Targeted attack ```

```  from here on, metrics are aggregated by [target], signature & reason (signals) needs to be accumulated and carried over via mvappend() ```
| eventstats min(start_time) AS start_time_by_target, max(end_time) AS end_time_by_target, sum(count_success) AS count_success_sum, sum(count_fail
    ) AS count_fail_sum, dc(attack_origin) AS count_attack_origin, values(attack_origin) AS attack_origin_values
  BY target

| eval attack_th=5  ``` count_attack_origin>attack_th + count_fail_sum>=count_success_sum => Targeted Brute-Force ```
| eval attack_span=(1+end_time_by_target-start_time_by_target)
| eval attack_flag=case(
  count_attack_origin>attack_th AND count_fail_sum>=count_success_sum AND count_success_sum>0, "Potential Successful",
  count_attack_origin>attack_th AND count_fail_sum>=count_success_sum, "Potential",
  1=1, null())
| eval signature=if(isnotnull(attack_flag), mvappend(attack_flag." Targeted Brute-Force Attack", signature), signature)
| eval reason=if(isnotnull(attack_flag), mvappend(attack_flag." Targeted Brute-Force Attack: there were [".count_fail_sum."] failed attempts and
    [".count_success_sum."] successful login(s) observed from multiple origins [".mvjoin(attack_origin_values, ", ")."] towards target [".target
    ."] over ".attack_span." second(s) between ".strftime(start_time_by_target,"%F %T")." and ".strftime(end_time_by_target,"%F %T").".", reason),
    reason)
```

origin	target	count_attack_origin	count_fail_sum	count_success_sum	signature	reason
bird	donkeykongjr	6	6	1	Potential Successful Targeted Brute-Force Attack	Potential Successful Targeted Brute-Force Attack: there were [6] failed attempts and [1] successful login(s) observed from multiple origins [clyde, johnson, jordan, malone, pippen, robinson] towards target [donkeykongjr] over 1801 second(s) between 2021-04-11 18:00:21 and 2021-04-11 18:30:21.
bird	larrykoopa	6	6	0	Potential Targeted Brute-Force Attack	Potential Targeted Brute-Force Attack: there were [6] failed attempts and [0] successful login(s) observed from multiple origins [bird, clyde, johnson, jordan, pippen, robinson] towards target [larrykoopa] over 1 second(s) between 2021-04-11 17:50:21 and 2021-04-11 17:50:21.
bird	princesspeach	7	7	0	Potential Targeted Brute-Force Attack	Potential Targeted Brute-Force Attack: there were [7] failed attempts and [0] successful login(s) observed from multiple origins [bird, clyde, johnson, jordan, malone, pippen, robinson] towards target [princesspeach] over 1 second(s) between 2021-04-11 18:10:21 and 2021-04-11 18:10:21.
clyde	donkeykongjr	6	6	1	Potential Successful Targeted Brute-Force Attack	Potential Successful Targeted Brute-Force Attack: there were [6] failed attempts and [1] successful login(s) observed from multiple origins [clyde, johnson, jordan, malone, pippen, robinson] towards target [donkeykongjr] over 1801 second(s) between 2021-04-11 18:00:21 and 2021-04-11 18:30:21.
```

Crafting the query: Summarization

```
64 | stats min(start_time) AS start_time, max(end_time) AS end_time, values(target) AS target, values(signature) AS signature, values(reason) AS reason
65  BY origin  ``` aggregate by origin of attacks (best bet for this sort of detection) ```
66
67 | eval attack_hash=md5(mvjoin(mvappend(reason, ""), "")) ``` add as many distinct fields to mvappend() as necessary (sourcetype, customer) for distinct summarization ```
68
69 | stats min(start_time) AS start_time, max(end_time) AS end_time, values(origin) AS origin, values(target) AS target, values(signature) AS signature, values(reason) AS reason
70  BY attack_hash  ``` aggregate by attack flow ```
71
72 ``` normalize alert output accordingly (suggestion: Splunk's CIM) ```
73 | rename origin AS src, target AS user
74 | fields - attack_hash ``` you may use this for alert id (dedup, throttling, suppression, etc) ```
```

👇 Full output follows

Ready for ES' IR dashboard

Context-rich alert

Unlucky attacker

All affected target accounts

Lucky attacker

start_time	end_time	src	user	signature	reason
1618159121	1618159122	blinky	backup	Potential Successful Brute-Force Attack	Potential Successful Brute-Force Attack: There were [6] failed attempts and [1] successful login(s) observed from origin [blinky] towards [backup] over 2 second(s) between 2021-04-11 18:38:41 and 2021-04-11 18:38:42. Rate: ~4 attempts/s.
1618158621	1618159220	speedy	admin administrator backup contact guest info root test	Mass Brute-Force Attack Potential Brute-Force Attack	Mass Brute-Force Attack: more than 4 brute-force targets observed from same origin [speedy]. Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [admin] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s. Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [administrator] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s. Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [contact] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s. Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [guest] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s. Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [info] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s. Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [root] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s. Potential Brute-Force Attack: There were [8] failed attempts and [0] successful login(s) observed from origin [speedy] towards [test] over 1 second(s) between 2021-04-11 18:30:21 and 2021-04-11 18:30:21. Rate: ~8 attempts/s.
1618155921	1618155921	shadow	admin administrator contact guest info root test	Potential Password Spray Attack	Potential Password Spray Attack: there were [7] failed attempts (~1/attacked target) and [0] successful login(s) observed from origin [shadow] towards [7] targets [admin, administrator, contact, guest, info, root, test] over 1 second(s) between 2021-04-11 17:45:21 and 2021-04-11 17:45:21.
1618159220	1618159221	pokey	admin	Potential Brute-Force Attack	Potential Brute-Force Attack: There were [6] failed attempts and [0] successful login(s) observed from origin [pokey] towards [admin] over 2 second(s) between 2021-04-11 18:40:20 and 2021-04-11 18:40:21. Rate: ~3 attempts/s.
1618156821	1618157421	malone	donkeykongjr princesspeach	Potential Successful Targeted Brute-Force Attack Potential Targeted Brute-Force Attack	Potential Successful Targeted Brute-Force Attack: there were [6] failed attempts and [1] successful login(s) observed from multiple origins [clyde, johnson, jordan, malone, pippen, robinson] towards target [donkeykongjr] over 1801 second(s) between 2021-04-11 18:00:21 and 2021-04-11 18:30:21. Potential Targeted Brute-Force Attack: there were [7] failed attempts and [0] successful login(s) observed from multiple origins [bird, clyde, johnson, jordan, malone, pippen, robinson] towards target [princesspeach] over 1 second(s) between 2021-04-11 18:10:21 and 2021-04-11 18:10:21.
1618156821	1618158621	pinky	admin administrator contact guest info root test	Potential Successful Password Spray Attack	Potential Successful Password Spray Attack: there were [6] failed attempts (~1/attacked target) and [1] successful login(s) observed from origin [pinky] towards [6] targets [admin, administrator, contact, guest, info, root, test] over 1801 second(s) between 2021-04-11 18:00:21 and 2021-04-11 18:30:21.
1618156221	1618158621	bird clyde johnson jordan pippen robinson	donkeykongjr larrykoopa princesspeach	Potential Successful Targeted Brute-Force Attack Potential Targeted Brute-Force Attack	Potential Successful Targeted Brute-Force Attack: there were [6] failed attempts and [1] successful login(s) observed from multiple origins [clyde, johnson, jordan, malone, pippen, robinson] towards target [donkeykongjr] over 1801 second(s) between 2021-04-11 18:00:21 and 2021-04-11 18:30:21. Potential Targeted Brute-Force Attack: there were [6] failed attempts and [0] successful login(s) observed from multiple origins [bird, clyde, johnson, jordan, pippen, robinson] towards target [larrykoopa] over 1 second(s) between 2021-04-11 17:50:21 and 2021-04-11 17:50:21. Potential Targeted Brute-Force Attack: there were [7] failed attempts and [0] successful login(s) observed from multiple origins [bird, clyde, johnson, jordan, malone, pippen, robinson] towards target [princesspeach] over 1 second(s) between 2021-04-11 18:10:21 and 2021-04-11 18:10:21.

Total: 7 Attack Flows

Under the radar: Stealth successful attack

```
42 ```  stealth attack  ```
43 | append [ | makeresults count=3 | eval _time=relative_time(now(), "-20s"), src="dcarasso", signature="User failed to authenticate", action="failure", user="buttercup" | bin _time span=10s | streamstats count | eval _time=_time+(10-count) ]
44 | append [ | makeresults count=3 | eval _time=relative_time(now(), "-20s"), src="dcarasso", signature="User failed to authenticate", action="failure", user="buttercup" | bin _time span=10s | streamstats count | eval _time=_time+(9+count)  ]
45 | append [ | makeresults count=1 | eval _time=relative_time(now(), "-10s"), src="dcarasso", signature="User login successful", action="success", user="buttercup" | bin _time span=10s | eval _time=_time+3 ]
46 | fields - count
47 | collect index=auth
```

_time	src	user	signature	action
2021-04-11 18:40:07	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:08	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:09	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:10	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:11	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:12	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:13	dcarasso	buttercup	User login successful	success

```
4 | bin span=10s _time AS tspan
```

✓ 7 events (4/11/21 5:45:00.000 PM to 4/11/21 6:45:00.000 PM)

No Event Sampling

Events (7) Patterns Statistics (7) Visualization

100 Per Page ✓ Format Preview

_time	tspan	src	user	signature	action
2021-04-11 18:40:07	1618159200	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:08	1618159200	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:09	1618159200	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:10	1618159210	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:11	1618159210	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:12	1618159210	dcarasso	buttercup	User failed to authenticate	failure
2021-04-11 18:40:13	1618159210	dcarasso	buttercup	User login successful	success

3 failed attempts

Followed by 3 more and 1 successful!

Snap to the 10th second behavior
Attack spans across 2 consecutive time windows of 10s each, trigger condition not met!

BSIDES splunk>

Streamstats to the rescue!

Detection code change

```
9  ```| bin span=10s _time AS tspan```
10 | sort 0 +num(_time)
11 | streamstats time_window=10s min(_time) AS tspan by origin, target
12
13 ```vanilla BF attack metrics calculated over each time window (tspan), split by origin and target```
14 | stats min(_time) AS start_time, max(_time) AS end_time, count(failure_time) AS count_fail, count(success_time) AS count_success
15   BY origin, target, tspan
```

_time	tspan	src	user
2021-04-11 18:00:21	1618156821	pinky	contact
2021-04-11 18:05:21	1618157121	pinky	info
2021-04-11 18:10:21	1618157421	pinky	guest
2021-04-11 18:15:21	1618157721	pinky	administrator
2021-04-11 18:20:21	1618158021	pinky	test
2021-04-11 18:25:21	1618158321	pinky	root
2021-04-11 18:30:21	1618158621	pinky	admin
2021-04-11 18:40:07	1618159207	dcarasso	buttercup
2021-04-11 18:40:08	1618159207	dcarasso	buttercup
2021-04-11 18:40:09	1618159207	dcarasso	buttercup
2021-04-11 18:40:10	1618159207	dcarasso	buttercup
2021-04-11 18:40:11	1618159207	dcarasso	buttercup
2021-04-11 18:40:12	1618159207	dcarasso	buttercup
2021-04-11 18:40:13	1618159207	dcarasso	buttercup
2021-04-11 18:40:20	1618159220	pokey	admin
2021-04-11 18:40:20	1618159220	pokey	admin
2021-04-11 18:40:20	1618159220	pokey	admin
2021-04-11 18:40:21	1618159220	pokey	admin
2021-04-11 18:40:21	1618159220	pokey	admin
2021-04-11 18:40:21	1618159220	pokey	admin

start_time	end_time	src	user	signature	reason
1618159207	1618159213	dcarasso	buttercup	Potential Successful Brute-Force Attack	Potential Successful Brute-Force Attack: There were [6] failed attempts and [1] successful login(s) observed from origin [dcarasso] towards [buttercup] over 7 second(s) between 2021-04-11 18:40:07 and 2021-04-11 18:40:13. Rate: ~1 attempts/s.

- Window starts when the **first event is seen** from each tuple (src + user).
- Window ends when the 10s is over.
- By default, up to 10k events can fit into each time window.

- All SPL artifacts are available from GitHub at https://github.com/inodee

BSIDES splunk>

Take it to the next level

- Consider these results as *indicators* in case you have custom alert frameworks in place (RBA/UBA), apply scores/factors per signature
- For MSSPs or multi-tenant environments: simply add tenant (customer) to each stats' *group by* clause
- For Splunk's Authentication DM, simply add sourcetype to the *group by* clause to get alerts per distinct data source
- Use macros or lookups for dynamic thresholds
- Consider "Potential Account Abuse/Compromise" use case next

Thanks!

Questions or feedback?

You can find me at “splunk-usergroups” Slack (Alex Teixeira)

- @ateixei (Twitter/Medium)
- alex@opstune.com
- https://spl.ninja

slack

BSIDES splunk>