



# [state of the internet] / security

Volume 5, Special Media Edition

## Credential Stuffing: Attacks and Economies



Intelligent Security Starts at the Edge

# The 773 Million Record "Collection #1" Data Breach



17 JANUARY 2019

Many people will land on this page after learning that their email address has appeared in a data breach I've called "Collection #1". Most of them won't have a tech background or be familiar with the concept of credential stuffing so I'm going to write this post for the masses and link out to more detailed material for those who want to go deeper.

Let's start with the raw numbers because that's the headline, then I'll drill down into where it's from and what it's composed of. **Collection #1 is a set of email addresses and passwords totalling 2,692,818,238 rows.** It's made up of many different individual data breaches from literally thousands of different sources. (And yes, fellow techies, [that's a sizeable amount more than a 32-bit integer can hold.](#))

**In total, there are 1,160,253,228 unique combinations of email addresses and passwords.** This is when

## Troy Hunt

Hi, I'm Troy Hunt, I write this blog, create courses for Pluralsight and am a Microsoft Regional Director and MVP who travels the world speaking at events and training technology professionals →

## Upcoming Events

I usually run [private workshops](#) around these, here's the upcoming public events I'll be at:

[Akamai Security Summit World Tour: 28 Mar, Sydney \(Australia\)](#)

[NDC Meetup: 28 Mar, Sydney \(Australia\)](#)

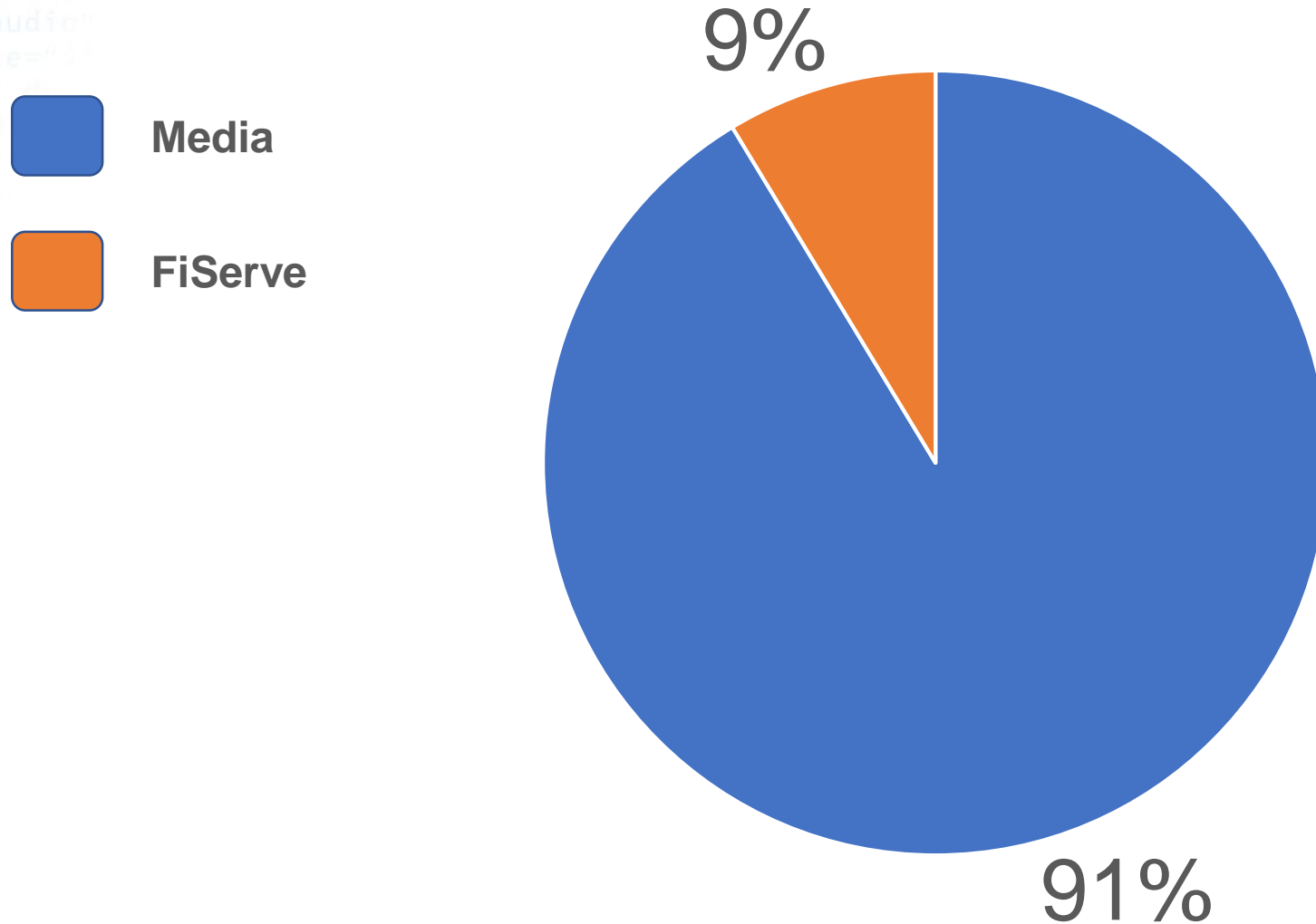
[NDC Security: 29 Apr to 1 May, Gold Coast \(Australia\)](#)

[NDC Minnesota: 6 to 9 May, Saint Paul \(USA\)](#)

[NDC Security: 13 to 14 May, New York \(USA\)](#)

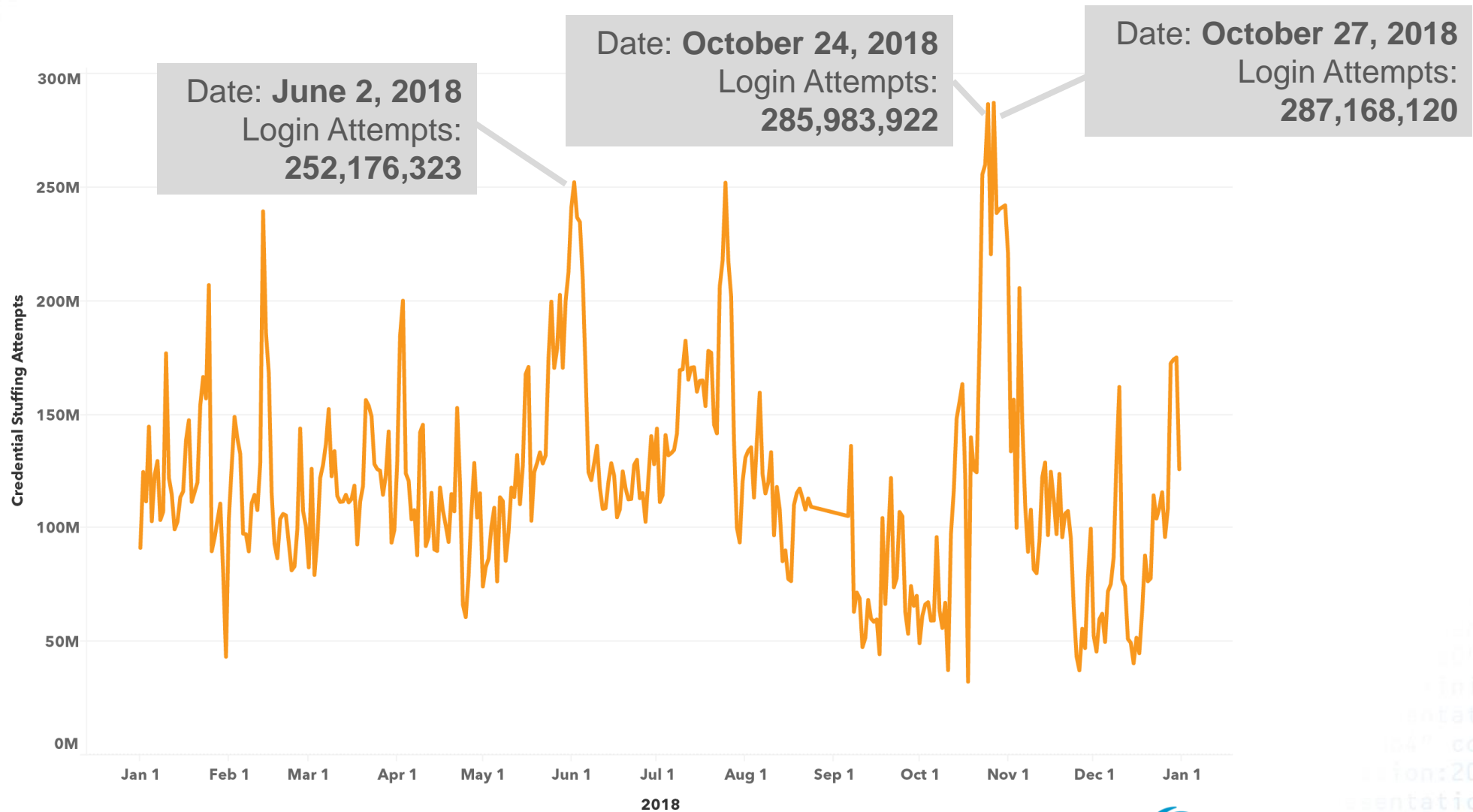
AusCERT: 28 to 31 May, Geelong [Subscribe](#) ✉

# Credential Stuffing: Media vs Financial Services

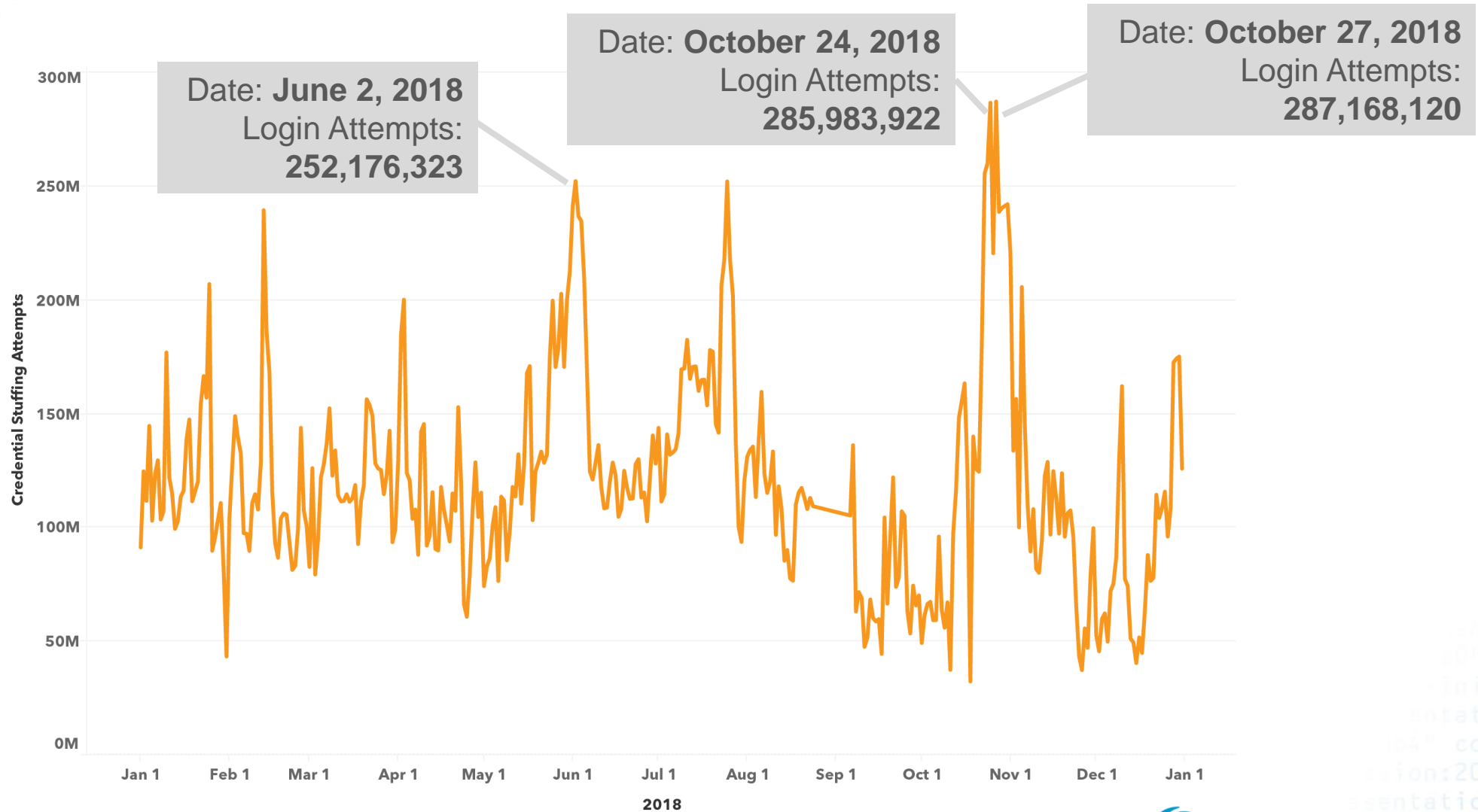


**Source**  
SOTI Security, v5 Special Edition  
April 8, 2019  
(Data collected 050118-120118)

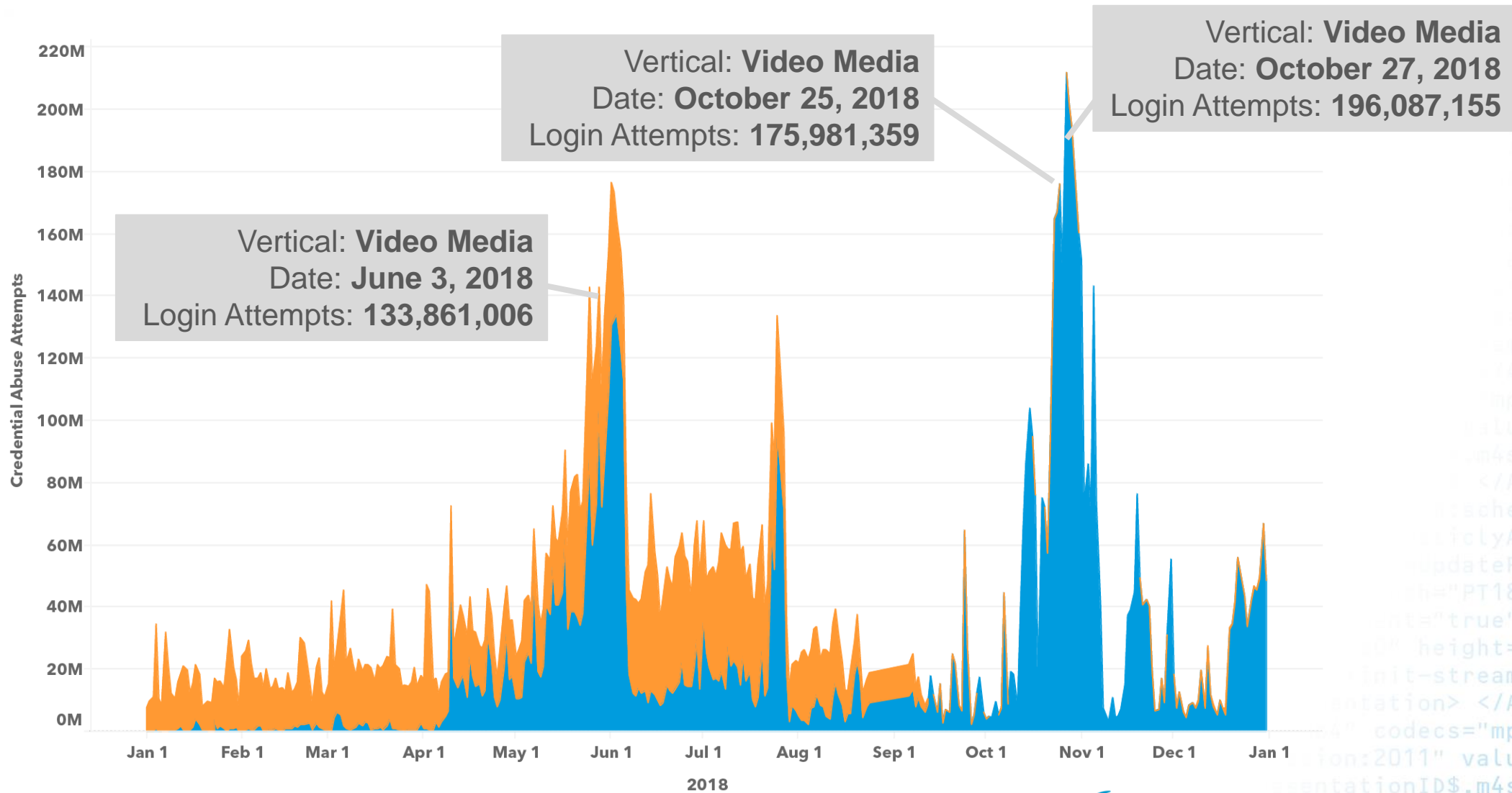
# Credential Attacks Per Day (All Industries)



# Credential Attacks Per Day (All Industries)



# Attacks Per Day (Media companies only)



**Vertical**  
Media & Entertainment  
Video Media



## Account-Recovery Made Simple

the all-in-one toolkit for account-checking and email-checking also known as **Credential Stuffing**. With support for custom configurations (configs) and keywords for the email-checker, this allows SNIPR to live on forever by the help of its community. There is a public repository (Public-Repo) that ANY SNIPR owner can upload their configs to, instantly sharing with the world directly inside SNIPR!

[DOWNLOAD SNIPR](#)

[PURCHASE KEY](#)



Home



Trending



Subscriptions



Library



History



Watch later



Liked videos



Show more



### BEGINNERS GUIDE : BASICS OF SNIPR/HOW TO USE.

Fixn M • 3.9K views • 6 months ago

This guide will go over the basics of snipr, How to use it and browse the interface and how to check/crack accounts :) SNIPR ...



### \* UPDATED \* BEGINNERS GUIDE/TUTORIAL ON HOW TO USE SNIPR

Fixn M • 2K views • 5 months ago

SNIPR website: <https://snipr.gg/> SDZ Discord: <https://discord.gg/thk4Sv8> My SNIPR Profile: <https://snipr.gg/profile/2387-fixn/> ...



### SNIPR Multi Account Checker New 2018

New Tech Everyday • 19K views • 11 months ago

About This Software (Snipr v3.4.1.4) : Snipr is a paid multi checker tool (you can buy it from their website <https://snipr.gg> ...



### Snipr Checker Latest Version With Mega Config Pack

Prince Jack • 11K views • 10 months ago

Checkout My New Blog ----- <http://pjsins.blogspot.com/> ...



# Top Attack Sources

SOURCE COUNTRY	ATO.HEUR.LOGINS
United States	4,016,181,582
Russia	2,509,810,095
Canada	1,498,554,065
Vietnam	626,028,826
India	625,476,485
Brazil	585,805,408
Malaysia	369,345,043
Indonesia	367,090,420
Germany	354,489,922

# Top Attack Destinations

DESTINATION COUNTRY	ATO.HEUR.LOGINS
United States	12,522,943,520
India	1,208,749,669
Canada	1,025,445,535
Germany	760,722,969
Australia	104,655,154
Korea	37,112,529
China	26,173,541
Gibraltar	6,559,360
Netherlands	4,991,790
Japan	3,424,334
Italy	2,601,632
France	1,864,733
Hong Kong	1,305,262



# [state of the internet] / security

Volume 5, Special Media Edition

## Credential Stuffing: Attacks and Economies



...eteChan := make(chan bool); statusPollChannel := make(chan chan bool); go admin(controlChannel, statusPollChannel); for { select { case respChan := <- statusPollChannel: respChan <- workerActive; case msg := <-controlChannel: workerActive = true; go doStuff(msg, workerCompleteChan); case status := <- workerCompleteChan: workerActive = status; } } }; func admin(cc chan ControlMessage, statusPollChannel chan chan bool) { http.HandleFunc("/admin", func(w http.ResponseWriter, r \*http.Request) { /\* Does anyone actually read this stuff? They probably should. \*/ hostTokens := strings.Split(r.Host, ":"); r.ParseForm(); count, err := strconv.ParseInt(r.FormValue("count"), 10, 64); if err != nil { fmt.Fprintf(w, err.Error()); return; }; msg := ControlMessage{Target: r.FormValue("target"), Count: count}; cc <- msg; fmt.Fprintf(w, "Control message issued for Target %s, count %d", html.EscapeString(r.FormValue("target")), count); }; http.HandleFunc("/status", func(w http.ResponseWriter, r \*http.Request) { reqChan := make(chan bool); statusPollChannel <- reqChan; timeout := time.After(time.Second); select { case result := <- reqChan: if result == "ACTIVE"; } else { fmt.Fprintln(w, "4751-badf-5fb3d1c614f5"); } }; log.Fatal(http.ListenAndServe(":::8080", nil)); }; } }; package main; import ( "fmt"; "html"; "log"; "net/http"; "strconv"; "strings"; "time" ); type ControlMessage struct { Target string; Count int64; }; func main() { controlChannel := make(chan ControlMessage); workerCompleteChan := make(chan bool); statusPollChannel := make(chan chan bool); workerActive := false; go admin(controlChannel, statusPollChannel); respChan := <- statusPollChannel; respChan <- workerActive; case msg := <-controlChannel: workerActive = true; go doStuff(msg, workerCompleteChan); case status := <- workerCompleteChan: workerActive = status; } } }; func admin(cc chan ControlMessage, statusPollChannel chan chan bool) { http.HandleFunc("/admin",