RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

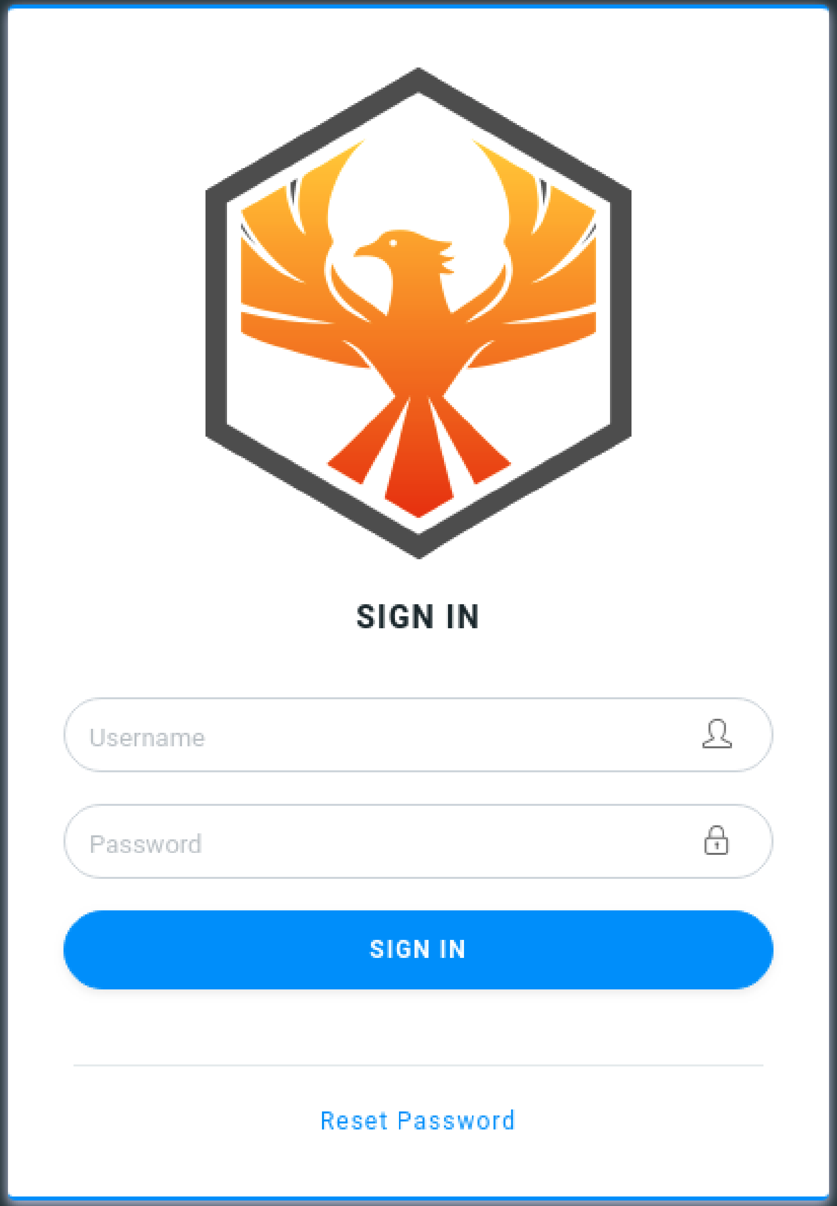
HUMAN ELEMENT

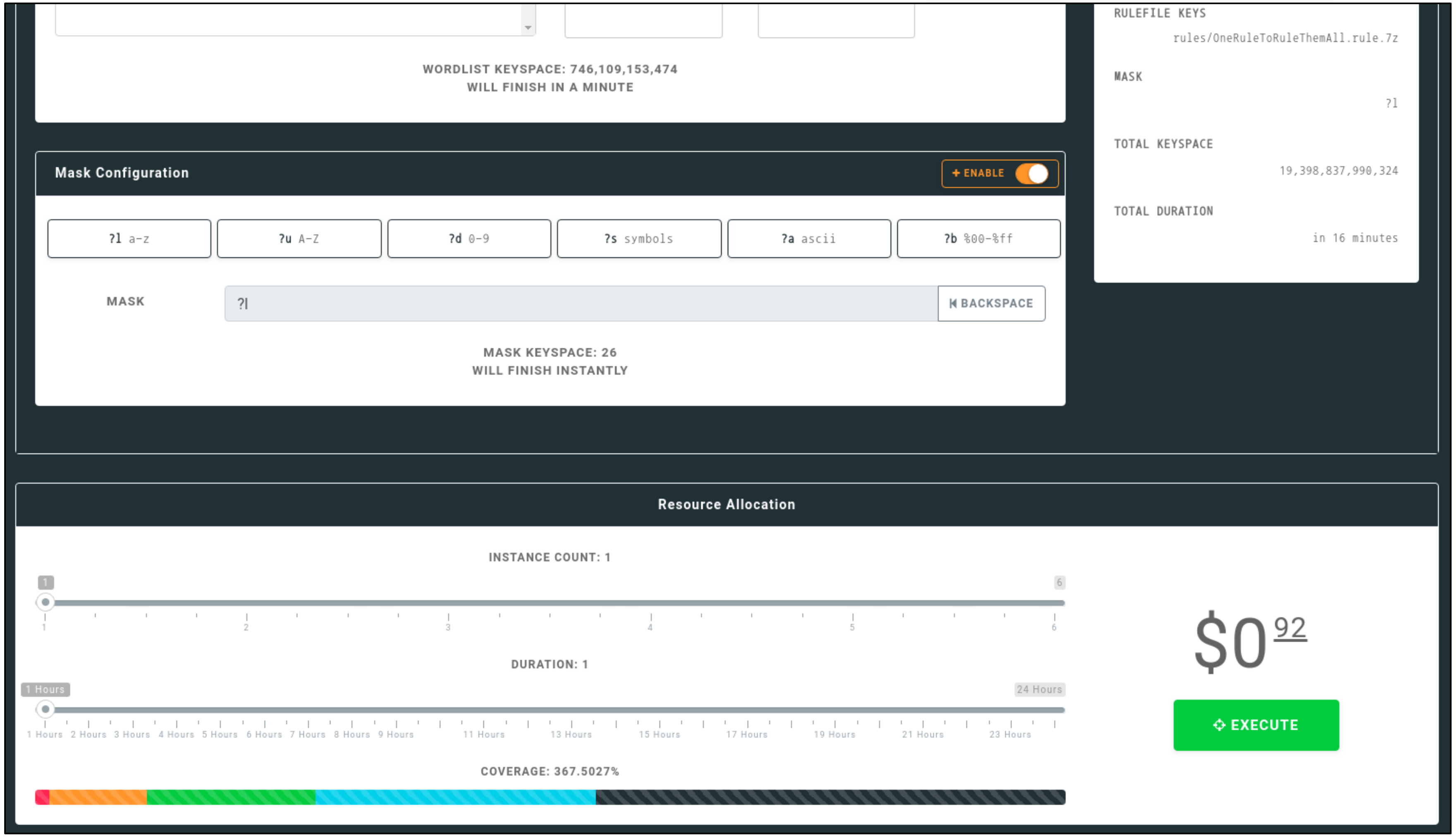
SESSION ID: OST-R01

NPK: High-Powered, Open Source Hash Cracking Tool

Brad Woodward

Director; Labs
Coalfire Systems


SIGN IN
Username
Password
SIGN IN
Reset Password



WORDLIST KEYSPACE: 746,109,153,474
WILL FINISH IN A MINUTE

Mask Configuration

+ ENABLE

MASK: ?l

MASK KEYSPACE: 26
WILL FINISH INSTANTLY

RULEFILE KEYS
rules/OneRuleToRuleThemAll.rule.7z

MASK
?l

TOTAL KEYSPACE
19,398,837,990,324

TOTAL DURATION
in 16 minutes

Resource Allocation

INSTANCE COUNT: 1

DURATION: 1

COVERAGE: 367.5027%

$0.92

Hashing Basics

What is Hashing?

- A hashing algorithm deterministically converts an arbitrary length input into a fixed length, unique* output.

Input	SHA1 Hash
Hello World!	2ef7bde608ce5404e97d5f042f95f89f1c232871
RSA 2019	cba24a235b934128482fca76086162c3d2405fe0
RSA 2020	4158141aae1507811df4a25ba5223dff14fcb4e7
RSA 2021	9041b7466c0bd9360818b5b117b0e1de7f324a9b

Hash Speed as a Work Factor

Hash Rates for Nvidia Tesla M60 GPU

Algorithm	Hash Rate
NTLM	18,300,000,000 H/s
NetNTLMv2	770,000,000 H/s
WPA2	189,200 H/s
bcrypt	7,010 H/s
PBKDF2-HMAC-Whirlpool + XTS 512	31 H/s

Useful Behaviors

Determinism

- The same input will always result in the same output

Deviance

- Similar inputs yield entirely dissimilar outputs

Uniform Distribution

- No statistically relevant bias exists for output across the keyspace

Where Hashing is Used

As a Zero-Knowledge Proof

- Determinism allows one party can verify that another knows a secret, without knowing the secret themselves.

As an Index-less Distributed Storage Lookup Method

- Cassandra
- DynamoDB

For Data Integrity Verification

As a Blockchain Lottery

Salting a Hash

A 'salt' is a unique value appended to an input value before performing the hash function, with the objective of preventing identical inputs from having identical outputs. The salt does not need to be a secret value.

Salting is Effective Against

- Rainbow Table Attacks
- Mass Cracking Campaigns with Large Hash Sets

Work Factors vs. Campaign Duration

Hash Algorithm	Campaign Duration
NTLM	58 Minutes
NetNTLMv2	23 Hours 27 Minutes
IKE-PSK SHA1	55 Hours 44 Minutes
WPA2	10 Years 321 Days
bcrypt	291 Years 246 Days

Hashcat for Cracking Campaigns

Hashcat

Password Candidates -> Hashing Algorithm -> Hardware Optimizations

Generating Candidates: Hashcat Masks

- Masks replace Bruteforce
  - ?s = symbol
  - ?l = lowercase
  - ?u = uppercase
  - ?d = digit

?s?u?l?l?l?l?d?d?d?d will crack ‘$Tiger2020’

Generating Candidates: Dictionaries and Rules

‘Dictionaries’ are simple wordlists

- Extremely fast, but very minimal success rate
- If the password doesn’t match an entry exactly, it won’t be recovered

‘Rules’ modify candidates in deterministic ways:

- Add a symbol to the front
- Capitalize the first letter
- Add four digits to the end

How to Build Campaigns Wrong

“I got a hash! I should throw every wordlist and every rule file at it at the same time!”

“I got a hash, let’s run it through RockYou really quick”

“I have one set of dictionaries and rule files I always use”

RSAConference2020

Using NPK for Distributed, Cloud-based Campaign Management

Using NPK

Demo

Using NPK

p3.16xlarge

- 8x Tesla V100 GPUs
- NTLM @ 633 GH/s
- $0 Up-Front
- ~$176/day

Professional / Commercial

- 8x Nvidia GTX 1080Ti
- NTLM @ 416 GH/s
- $26,000 Up-Front
- ~$3/day

Price Crossover: 3,468 Hours = 144 Days
Performance Crossover: 5,357 Hours = 223 Days

Challenges/Risks of Cracking in AWS

- Some EC2 knowledge required
  - Internet Access, VPCs, Keys, Security Groups
- Instance set-up has costs, too
  - Installing drivers & Hashcat
  - Uploading Wordlists, Rules, and Hashes
- Securing Hashes/Recovered Plaintexts
  - Risk Tolerance of Sensitive Data on Third-Party Platforms
  - Persistence of Data After Campaign
- Runaway Instances
  - Single p3.16xlarge instance = $24.48/hr = **$17,625**/mo.

Using NPK

- Campaign Limits
- Understanding Hash Benchmarks
- Coverage Estimates
- Viewing Campaign Progress & Status
- Showing Recovered Hashes

Apply what you've Learned

- Do the math!
  - Algorithm
  - Candidate Pool
  - Hardware Capabilities
- Consider effectiveness of NPK vs. flexibility of Hashcat alone
- Embrace NPK for safe, low risk cracking in the Cloud

Try it out!

- Deploy NPK, then:
  - Run your first campaign
    - Active Directory password strength assessment?
    - Wireless password resilience?
  - Compare strength of algorithms to inform password policies
    - How do 12-character Windows passwords compare to 8 characters on Linux?