

# Using Intelligence in Zeek

---



**Joe Abraham**

Cybersecurity Consultant

@joeabrah [www.defendthenet.com](http://www.defendthenet.com)



**Consumes data**

**Zeek matches on  
intelligence**

**Static intel files contain:**

**Domains**

**IP addresses**

**Email addresses**

**Hashes**

**Etc.**

# Zeek Intelligence Framework



# Using Intelligence

## **Add GeoIP information**

- Adds location information based on IP addresses**

## **Track and alert on malicious domains**

**Contextual data helps match on potentially malicious traffic**

**Can use intelligence to create alerts on specific criteria**



# Intelligence and Hashing

## File Analysis

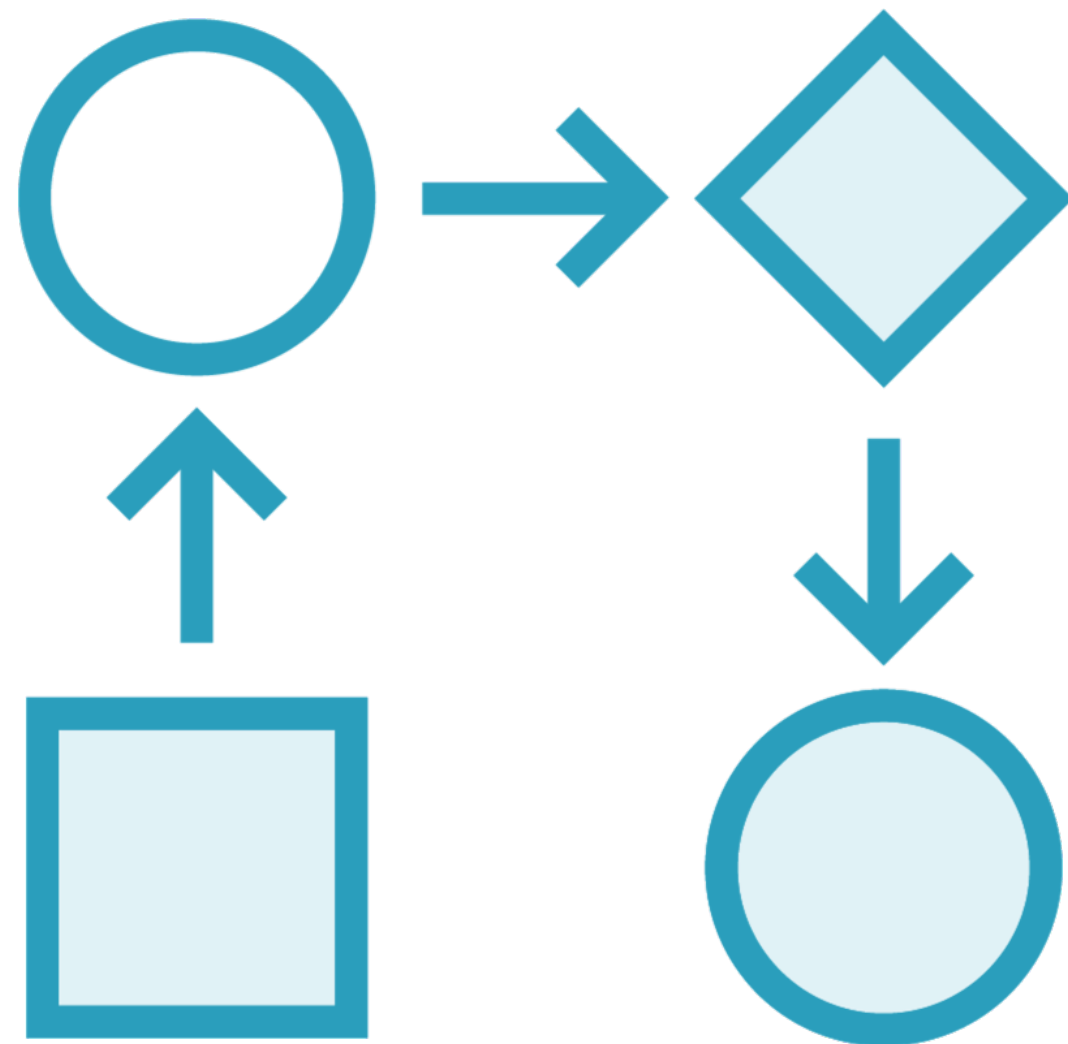
File hashes can be used to match on known malicious files

## JA3/JA3S

Provides hashes on SSL/TLS conversations for fingerprinting and matching known malicious fingerprints



# Intelligence Framework Workflow



**Intelligence file or downloaded intelligence**

**Format it using Zeek intel framework**

**Load intel framework and match traffic**

**Uses `intel.log`, `notice.log`, webhooks, or other actions**



# Intel Types

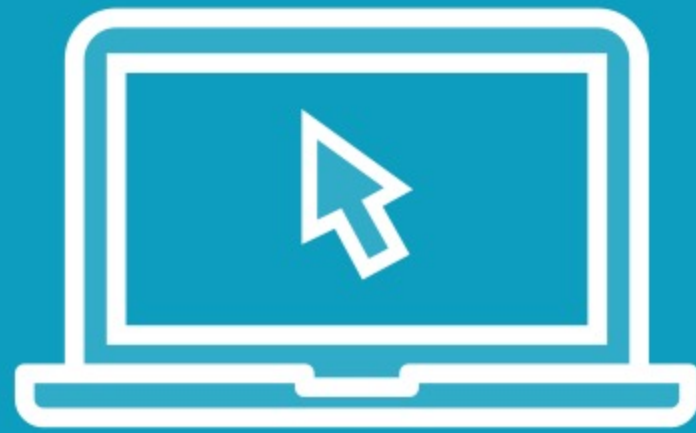
```
Intel::ADDR  
Intel::SUBNET  
Intel::URL  
Intel::SOFTWARE  
Intel::EMAIL  
Intel::DOMAIN  
Intel::USER_NAME  
Intel::CERT_HASH  
Intel::PUBKEY_HASH  
Intel::FILE_HASH  
Intel::FILE_NAME
```

# Intelligence Functions and Options

Item Name	Functionality
<a href="#"><u>Intel::Info: record</u></a>	Record used for the logging framework representing a positive hit within the intelligence framework.
<a href="#"><u>Intel::Item: record</u></a>	Represents a piece of intelligence.
<a href="#"><u>Intel::MetaData: record</u></a>	Data about an <a href="#"><u>Intel::Item</u></a> .
<a href="#"><u>Intel::Seen: record</u></a>	Information about a piece of “seen” data.
<a href="#"><u>Intel::Type: enum</u></a>	Enum type to represent various types of intelligence data.
<a href="#"><u>Intel::TypeSet: set</u></a>	Set of intelligence data types.
<a href="#"><u>Intel::log_intel: event</u></a>	
<a href="#"><u>Intel::match: event</u></a>	Event to represent a match in the intelligence data from data that was seen.
<a href="#"><u>Intel::insert: function</u></a>	Function to insert intelligence data.
<a href="#"><u>Intel::remove: function</u></a>	Function to remove intelligence data.
<a href="#"><u>Intel::seen: function</u></a>	Function to declare discovery of a piece of data in order to check it against known intelligence for matches.



# Demo



**Configure and validate URL matching with intelligence framework**





# What is JA3 and How Can We Use It?

---

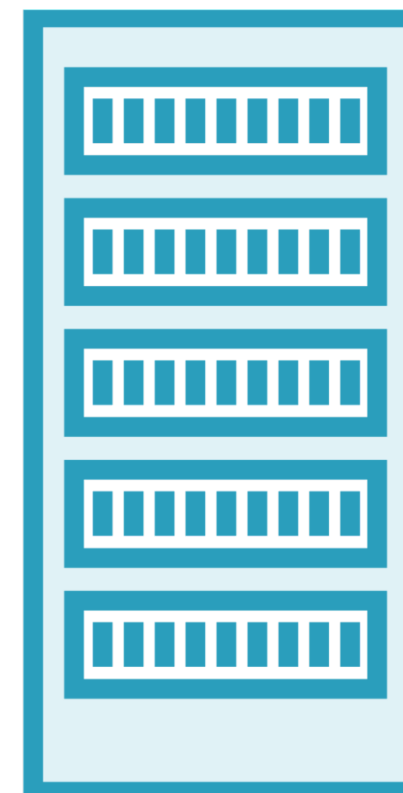


# JA3 and JA3S



**JA3**

Provides **client** side fingerprinting  
based on client hello packets



**JA3S**

Provides **server** side fingerprinting  
based on client hello packets



# JA3/JA3S Hashing

Our Client Hash: e7d705a3286e19ea42f587b344ee6865

Emotet JA3 Hash: 4d7a28d6f2263ed61de88ca66eb011e3

No Match!

# Zeek JA3 Package

**ja3**

<https://github.com/salesforce/ja3>

👁 95

★ 1288

🍴 179

🔔 16

Last Push 5/27/21, 3:31 PM

## JA3 - A method for profiling SSL/TLS Clients

JA3 is a method for creating SSL/TLS client fingerprints that should be easy to produce on any platform and can be easily shared for threat intelligence.

Before using, please read this blog post: [TLS Fingerprinting with JA3 and JA3S](#)

This repo includes JA3 and JA3S scripts for [Zeek](#) and [Python](#). You can find a nice Rust implementation of the JA3 algorithm [here](#)

JA3 support has also been added to:

- Moloch
- Trisul NSM
- NGiNX
- MISP
- Darktrace
- Suricata
- Elastic.co Packetbeat
- Splunk
- MantisNet
- ICEBRG
- Redsocks
- NetWitness
- ExtraHop
- Vectra Cognito Platform
- Corvil
- Java
- Go
- Security Onion
- AIEngine
- RockNSM
- Corelight
- VirusTotal
- SELKS
- Stamus Networks
- and more...

To download to your Zeek instance: **zkg install salesforce/ja3**



# Demo



**Use Zeek package to gather JA3/JA3S hashes**



# Additional Zeek Integration Information

---



# Course Review

**Identifying Zeek  
Integrations**

**Deploying Zeek with  
Security Onion**

**Ingesting and  
Enriching Zeek Logs**

**Integrating Zeek  
with RockNSM**

**Using Intelligence in  
Zeek**



# Additional Resources



<https://docs.zeek.org/en/master/index.html>



<https://packages.zeek.org/packages>



<https://arkime.com>



<https://www.elastic.co>





# Placeholder



Thank You!

