Enhance Encrypted Network Telemetry

https://github.com/salesforce/ja3

bro-pkg install ja3
Jeff Atkinson is a security researcher with almost two decades focused in Information Security. He brings a unique perspective on defense strategies with a strong background in Incident Response, Threat Intelligence, and Malware Analysis. In both public and private sectors, including Fortune 50 companies, he deployed scalable custom network monitoring solutions, always including his favorite tool, Bro. He is one of the original creators of the JA3 fingerprint technique.

$whoami

First Version I used: Bro-0.7
How SSL works
How SSL works

corp net

NIDS
or something

AWS

Inernet
How SSL works
How SSL works

corp net → Internet

NIDS or something

AWS
How SSL works

CorpNet

Internet

SYNack

AWS

NIDS

or something
How SSL works

corp net

Internet

ACK

AWS

NIDS or something
How SSL works

corp net

Internet

Hello

AWS

NIDS

or something
How SSL works

CorP Net \(\rightarrow\) Internet

Hello

NIDS or something

AWS
How SSL works

corp.net | Internet

x509

NIDS or something
Tools

TTPs

Artifacts

Domain Names

IP Addresses

Hash Values

Harder for threat actors to change

X509 Certificates
event x509_certificate(f: fa_file, cert_ref: opaque of x509, cert: X509::Certificate)

type Certificate: record {
    version: count &log;  ##< Version number.
    serial: string &log;   ##< Serial number.
    subject: string &log;  ##< Subject.
    issuer: string &log;   ##< Issuer.
    cn: string &optional; ##< Last (most specific) common name.
    not_valid_before: time &log;  ##< Timestamp before when certificate is not valid.
    not_valid_after: time &log;   ##< Timestamp after when certificate is not valid.
    key_alg: string &log;  ##< Name of the key algorithm
    sig_alg: string &log;   ##< Name of the signature algorithm
    key_type: string &optional &log;  ##< Key type, if key parseable by openssl (either rsa, dsa or ec)
    key_length: count &optional &log;  ##< Key length in bits
    exponent: string &optional &log;  ##< Exponent, if RSA-certificate
    curve: string &optional &log;  ##< Curve, if EC-certificate
};
Metasploit SSL X509

Story Time
Default Metasploit SSL Cert in Bro

x509.log

certificate.issuer:
CN=hrzvox.gov,O=bdlOFqMXlUfgogNQljMuRWgiJ,L=ZTIhjQVsJEuQI1S
gScdegcLSLJVRE,ST=W1,C=US

certificate.subject:
CN=vl3qykkr.com,O=UPdkxNEasODSAlkvuaEMm,L=SZewokfDFSkaAsf
KyeJMNtFlfleGT,ST=NV,C=US
def makessl(params)
    ssl_cert = params.ssl_cert
    if ssl_cert
        issuer = OpenSSL::X509::Name.new(["C","US"],
            ['ST', Rex::Text.rand_state()],
            ['L', Rex::Text.rand_text_alpha(rand(20) + 10)],
            ['O', Rex::Text.rand_text_alpha(rand(20) + 10)],
            ['CN', Rex::Text.rand_hostname])
    end
end
Default Metasploit SSL Cert in Bro

x509.log

certificate.issuer:
CN=hrzvox.gov,
O=bd1OFqMX1Ufg0NQljMuRWgiJ,
L=ZTIhjQVsJEuQIlSgScdegcLSLJVRE,
ST=WI,
C=US
Regex match on rand mixed alpha?

bdlOFqMXlUfgONQljMuRWgiJ
ZTIhjQVsJEuQIlSgScdegcLSLJvre
alDSFlkasfQWAFlksSA
aAfkVCIQmdSDlEkfASgKJZEk
KfaNmtFxGptqeK
jQVsJEuQIlSgoNQljMuR
CtQmddlOFqMXlUldSFSgQljM
SgoNQljasfOFqMXl
KfIKwlMCZoetFFaLKXZ
[a-z][A-Z]{2}
if ( !(cert?$issuer) || (/C=US/ !in cert$issuer) )
    return;

local conn: connection;
for ( c in f$conns )
    conn=f$conns[c];

local metasploit = /[a-z][A-Z]{2}/;
local x509_data: table[string] of string = table();
local parts = split(cert$issuer, /,);  
for ( part_index in parts )
{
    local key_val = split1(parts[part_index], /=/);
    if ( 2 in key_val)
        x509_data[key_val[1]] = key_val[2];
}

if ( "C" in x509_data "C" == "US" && "L" in x509_data & metasploit in x509_data["L"] )
    NOTICE([$note=Metasploit_SSL_Cert, $conn=conn, $msg=fmt("Metasploit SSL, random issuer US city '%s', x509_data["L"]", x509_data["L"])), $sub=cert$issuer, $identifier=cert$issuer]);
Metasploit SSL Round 2
This change emulates the auto-generated snakeoil certificate from Ubuntu 14.04. The main changes include moving to 2048-bit RSA, SHA256, a single name CN for subject/issuer, and the removal of most certificate extensions.
def self.ssl_generate_certificate
    yr = 24*3600*365
    vf = Time.at(Time.now.to_i - rand(yr * 3) - yr)
    vt = Time.at(vf.to_i + (10 * yr))
    cn = Rex::Text.rand_text_alpha_lower(rand(8)+2)
    key = OpenSSL::PKey::RSA.new(2048)
    cert = OpenSSL::X509::Certificate.new
    cert.version = 2
    cert.serial = (rand(0xFFFFFFFF) << 32) + rand(0xFFFFFFFF)
    cert.subject = OpenSSL::X509::Name.new(["CN", cn])
    cert.issuer = OpenSSL::X509::Name.new(["CN", cn])
    cert.not_before = vf
    cert.not_after = vt
    cert.public_key = key.public_key
    ef = OpenSSL::X509::ExtensionFactory.new(nil, cert)
    cert.extensions = [ef.create_extension("basicConstraints","CA:FALSE")]
    ef.issuer_certificate = cert
    cert.sign(key, OpenSSL::Digest::SHA256.new)
Metasploit SSL Round 2

Bro logs

ip.orig_h: 10.1.2.3
ip.orig_P: 1984
ip.resp_h: 192.0.2.1
ip.resp_p: 443
subject: CN=qjpozixk
issuer: CN=qjpozixk
version: TLSv12
certificate.sig_alg: sha256WithRSAEncryption
validation_status: self signed certificate
**Snakeoil Cert**

- Issuer contains CN only
- Issuer and Subject are the same
- 2048bit Key
- Version 3
- Valid for 10 years
  - Starting now
- Usually SHA1 (for now)
- CN = Hostname.Domain

**Metasploit Cert**

- Issuer contains CN only
- Issuer and Subject are the same
- 2048bit Key
- Version 3
- Valid for 10 years
  - Starting now - rand(yr * 3) - yr
- Always SHA256
- CN = rand_text_alpha_lower(rand(8)+2)
Snakeoil Cert

- Issuer contains CN only
- Issuer and Subject are the same
- 2048bit Key
- Version 3
- Valid for 10 years
  - Starting now
- Usually SHA1 (for now)
- CN = Hostname.Domain

Metasploit Cert

- Issuer contains CN only
- Issuer and Subject are the same
- 2048bit Key
- Version 3
- Valid for 10 years
  - Starting now - rand(yr * 3) - yr
- Always SHA256
- CN = rand_text_alpha_lower(rand(8)+2)
event x509_certificate(f: fa_file, cert_ref: opaque of x509, cert: X509::Certificate)
{
  for ( cid in f$conns )
    { if ( cid$resp_h in 10.0.0.0/8 ) { return; } }
if ( ! cert?$subject ) { return; }
if ( ! cert?$issuer ) { return; }
if ( cert$subject in falselist ) { return; }
if ( cert$subject != cert$issuer ) { return; }
if ( /^CN=[a-z]{2,10}$/ == cert$subject )
if ( "sha256WithRSAEncryption" == cert$sig_alg )

NOTICE([$note=Metasploit_SSL_Cert, $conn=f$conns[cid],
  $msg=fmt("Metasploit Randomly Generated SSL Cert, '%s',
cert$subject),
  $sub=cert$issuer]);
}
{ if ( cid$resp_h in 10.0.0.0/8 ) { return; } }
if ( ! cert?$subject ) { return; }
if ( ! cert?$issuer ) { return; }
if ( cert$subject in falselist ) { return; }
if ( cert$subject != cert$issuer ) { return; }
if ( /^CN=[a-z]{2,10}$/ == cert$subject )
if ( "sha256WithRSAEncryption" == cert$sig_alg )

NOTICE
if ( cert$subject != cert$issuer ) { return; }
if ( /^CN=[a-z]{2,10}$/ == cert$subject )
if ( "sha256WithRSAEncryption" == cert$sig_alg )
Metasploit SSL Round 3
Apparently auto-generated certs are getting snagged by some AV. The format of the generated certs is easy to regex against, and it doesn't help that the subject and issuer are identical.

Currently, the cert looks like this:

```bash
$ openssl s_client -connect 10.1.1.12:4444 2>/dev/null | egrep "^subject|^issuer"
subject=/CN=ucb1gdz
issuer=/CN=ucb1gdz
```

which will change slightly in length each time it's generated, but not much. Super easy to detect.

This PR makes them look a little better:

```bash
$ openssl s_client -connect 10.1.1.12:4444 2>/dev/null | egrep "^subject|^issuer"
subject=/C=US/S=WI/L=Denise/O=Gary/CN=41tw6z.y1.biz
issuer=/C=US/O=Timothy/CN=Alan Jessica

$ openssl s_client -connect 10.1.1.12:4444 2>/dev/null | egrep "^subject|^issuer"
subject=/C=US/S=WI/L=Steven/O=George/CN=p.h1qtuz.org
issuer=/C=US/O=Andrea/CN=Bobby Jeremy

$ openssl s_client -connect 10.1.1.12:4444 2>/dev/null | egrep "^subject|^issuer"
subject=/C=US/S=WI/L=Antonio/O=Roy/CN=R5w13jedh.a.edu
issuer=/C=US/O=Nicholas/CN=Gregory Harry
```
Certificate Style

subject=/C=US/ST=WI/L=Denise/O=Gary/CN=41tw6z.yl.biz
issuer=/C=US/O=Timothy/CN=Alan Jessica

subject=/C=US/ST=WI/L=Steven/O=George/CN=p.h1qtuz.org
issuer=/C=US/O=Andrea/CN=Bobby Jeremy

subject=/C=US/ST=VT/L=Antonio/O=Roy/CN=td.0swgljfedb.a.edu
issuer=/C=US/O=Nicholas/CN=Gregory Harry
Certificate Style

subject=/C=US/ST=WI/L=Denise/O=Gary/CN=41tw6z.yl.biz
issuer=/C=US/O=Timothy/CN=Alan Jessica

subject=/C=US/ST=WI/L=Steven/O=George/CN=p.h1qtuz.org
issuer=/C=US/O=Andrea/CN=Bobby Jeremy

subject=/C=US/ST=VT/L=Antonio/O=Roy/CN=td.0swgljfedb.a.edu
issuer=/C=US/O=Nicholas/CN=Gregory Harry

subject=^C=US\ST=[A-Z]{2}\VL=[A-Z][a-z]+\VO=[A-Z][a-z]+\VCN=(\w\.)+$
issuer=^C=US\VO=[A-Z][a-z]+\VCN=[A-Z][a-z]+\s[A-Z][a-z]+$
Extract and mixin cert ops from server module

Generic SSL routines can be in their own module, for import by consumers without having to drag the entire server infrastructure in with it.

This pulls the certificate methods into Rex::Socket::Ssl for use by consumers, and includes the module in Rex::Socket::SslTcpServer as the initial consumer.

GitHub

master (#8)

RageLtMan committed 20 days ago

Showing 2 changed files with 150 additions and 136 deletions.
Certificate Style

Subject: C=/C=US/ST=VA/O=Nienow LLC/OU=connect/CN=nienow.llc.org/emailAddress=connect@nienow.llc.org
Issuer: C=/C=US/ST=VA/O=Nienow LLC/OU=connect/CN=nienow.llc.org/emailAddress=connect@nienow.llc.org

Subject: C=/C=US/ST=TN/O=Toy-Rippin/OU=interface/CN=toy.rippin.org/emailAddress=interface@toy.rippin.org
Issuer: C=/C=US/ST=TN/O=Toy-Rippin/OU=interface/CN=toy.rippin.org/emailAddress=interface@toy.rippin.org

CertSubject: C=/C=US/ST=*  
CertIssuer: C=/C=US/ST=*

```
69 + cert.serial = (rand(@xFFFFFFFF) << 32) + rand(@xFFFFFFFF)
70 + cert.subject = OpenSSL::X509::Name.new(["C", subject])
71 + cert.issuer = OpenSSL::X509::Name.new(["C", issuer])
```
Tools

TTPs

Artifacts

Domain Names

IP Addresses

Hash Values

X509 Certificates

Harder for threat actors to change
Analysis of Googlebot's frugal cipher suite list

July 02, 2009

Two weeks ago, I announced SSL Labs and my technique for passive SSL cipher suite analysis. It won't surprise you to learn that I've been carefully observing the cipher suites used in the requests that came to the web site since. (In fact, I announced the site slightly earlier than I had planned because I wanted to get my hands on some real-life data.) One client's SSL fingerprint immediately caught my attention, because it supported only 4 cipher suites. It was Googlebot.

There were 115 visits from Googlebot in the two-week period, using 5 different user agent strings (although Googlebot will sometimes send a request without User-Agent set):

- Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
- SAMSUNG-SGH-E250/1.0 Profile/MIDP-2.0 Configuration/CLDC-1.1
...then there was Lee Brotherston

TLS fingerprinting

Smarter Defending & Stealthier Attacking

Posted on September 25, 2015

Background

Transport Layer Security (TLS) provides security in the form of encryption to all manner of network connections from legitimate financial transactions, to private conversations, and malware calling home. The inability for an eavesdropper to analyze this encrypted traffic protects its users, whether they are legitimate or malicious. Those using TLS operate under the assumption that although an eavesdropper can easily observe the existence of their session, its source and destination IP addresses, that the content itself is secure and unreadable without access to cryptographic keying material at one or both ends of the connection. On the surface this holds true, barring any configuration flaws or exploitable vulnerabilities. However, using TLS Fingerprinting, it is easy to quickly and passively determine which client is being used, and then to apply this information from both the attacker and the defender perspectives.

Previously, I have been able to demonstrate that certain clients could be differentiated from other network traffic. Specifically, that meant discriminating SuperFish, PrivDog, and GeniusBox from mainstream browsers when making HTTPS connections, and generating IDS signatures based on these findings to assist network administrators in being able to identify problematic hosts without requiring access to either endpoint. I have now expanded this technique to improve the accuracy of the fingerprints; provide tools to enable others to create fingerprints; and tools that will enable use by others in their own environments.

TLS
Our Requirements

- Needs to work on existing tools
- Destination agnostic, focused on client
- Unique to client application
- Easy to create
- Easy to share
- Easy to consume by any tool
How SSL works

- TLSv1.2 Record Layer: Handshake Protocol: Client Hello
  - Content Type: Handshake (22)
  - Version: TLS 1.2 (0x0303)
  - Length: 227
- Handshake Protocol: Client Hello
  - Handshake Type: Client Hello (1)
  - Length: 223
  - Version: TLS 1.2 (0x0303)
  - Random
    - Session ID Length: 32
    - Session ID: 575ee6393e5f5a73b8ae368cf6e5826b
  - Cipher Suites Length: 26
  - Cipher Suites (13 suites)
    - Compression Methods Length: 1
    - Compression Methods (1 method)
    - Extensions Length: 124
    - Extension: server_name
    - Extension: elliptic_curves
    - Extension: ec_point_formats
    - Extension: signature_algorithms
    - Extension: next_protocol_negotiation
    - Extension: Application Layer Protocol Negotiation
    - Extension: status_request
    - Extension: signed_certificate_timestamp
    - Extension: Extended Master Secret
Microsoft Edge (Browser)

TLS Version: 1.2
Cipher Suites: 19
Extensions: 107
Dridex Malware (Banking Trojan)

TLS Version 1.2
Cipher Suites: 21
Extensions: 41
Trickbot Malware (Banking Trojan)

TLS Version: 1
Cipher Suites: 12
Extensions: 21
Microsoft Edge (Browser)
Trickbot Malware (Banking Trojan)
Fingerprinting TLS - The JA3 Method
# Fingerprinting TLS - The JA3 Method

<table>
<thead>
<tr>
<th>Version</th>
<th>Ciphers</th>
</tr>
</thead>
<tbody>
<tr>
<td>771</td>
<td>49172-157-156-61-53-47-10</td>
</tr>
</tbody>
</table>
Fingerprinting TLS - The JA3 Method

<table>
<thead>
<tr>
<th>Version</th>
<th>Ciphers</th>
<th>Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>771</td>
<td>49172-157-156-61-53-47-10</td>
<td>0-5-10-11-13</td>
</tr>
</tbody>
</table>
## Fingerprinting TLS - The JA3 Method

Version, Ciphers, Extensions, EllipticCurves

Fingerprinting TLS - The JA3 Method

Version,Ciphers,Extensions,EllipticCurves,ECPointFormats

771,49172-157-156-61-53-47-10,0-5-10-11-13,29-23-24,0
# Fingerprinting TLS - The JA3 Method

<table>
<thead>
<tr>
<th>Version</th>
<th>Ciphers</th>
<th>Extensions</th>
<th>EllipticCurves</th>
<th>ECPointFormats</th>
</tr>
</thead>
<tbody>
<tr>
<td>771</td>
<td>49172-157-156-61-53-47-10</td>
<td>0-5-10-11-13</td>
<td>29-23-24</td>
<td>0</td>
</tr>
</tbody>
</table>

MD5 hash
Fingerprinting TLS - The JA3 Method

Version, Ciphers, Extensions, Elliptic Curves, EC Point Formats
771, 49172-157-156-61-53-47-10, 0-5-10-11-13, 29-23-24, 0

MD5 hash

JA3 = f4c4f050188e15839a6cd3af798b6c77
Fingerprinting TLS - The JA3 Method

Version,Ciphers,Extensions,EllipticCurves,ECPointFormats

771,49172-157-156-61-53-47-10,,,

MD5 hash

JA3 = 4dd4fca5534245b13b641d54a7035851
Fingerprinting TLS - The JA3 Method

$\text{MD5 sum} \begin{pmatrix} 771,49196-49195-49200-49199-159-158 \\ -49188-49187-49192-49191-49162-4916 \\ 1-49172-49171-157-156-61-60-53-47-10 \\ ,0-5-10-11-13-35-23-65281,29-23-24,0 \end{pmatrix}$
Google Chrome

JA3 = 94c485bca29d5392be53f2b8cf7f4304
Microsoft Edge

JA3 = 10ee8d30a5d01c042afd7b2b205facc4
Tor Client

JA3 = e7d705a3286e19ea42f587b344ee6865
Dridex Malware

JA3 = 74927e242d6c3febfb8cb9cab10a7f889
Trickbot Malware

JA3 = 6734f37431670b3ab4292b8f60f29984
Mapping JA3 to Client Application
"Copyright (c) 2017 salesforce.com inc.
All rights reserved.
Licensed under the BSD 3-Clause license.
For full license text see LICENSE.txt file in the repo root or https://opensource.org/licenses/bsd-3-cla
## Mapping JA3 to Client Application

<table>
<thead>
<tr>
<th>JA3</th>
<th>Client Application</th>
<th>count</th>
<th>percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>f58966d34ff9488a83797b55c804724d</td>
<td>Google Chrome</td>
<td>451</td>
<td>40.952761</td>
</tr>
<tr>
<td>94c485bca295d392be53f2b8cf7f4304</td>
<td>Google Chrome</td>
<td>339</td>
<td>30.790191</td>
</tr>
<tr>
<td>bc6c386f480ee979d9e52d472b772d8</td>
<td>Google Chrome</td>
<td>172</td>
<td>15.622162</td>
</tr>
<tr>
<td>be1a7de97ea176604a3c70622189d78d</td>
<td>Used by many programs on OSX, apple, WebKit, Networking</td>
<td>44</td>
<td>3.996367</td>
</tr>
<tr>
<td>c07cb55f88702033a8f52c46d23e0b2</td>
<td>Used by many programs on OSX, apple, WebKit, Networking</td>
<td>27</td>
<td>2.452316</td>
</tr>
<tr>
<td>83e04bc58d402f633983cbf22724b02</td>
<td>Charles, Google Play Music, Desktop Player, Postman, Slack, and other desktop programs</td>
<td>19</td>
<td>1.725704</td>
</tr>
<tr>
<td>a312f9162a08eefdf7feb7a13cd7e9bb</td>
<td>apple, WebKit, Networking, Spotify, WhatsApp, Skype, iTunes</td>
<td>13</td>
<td>1.180745</td>
</tr>
<tr>
<td>6cd1b944f5885e2cbfe98a840b76e8b8</td>
<td>Google Chrome</td>
<td>13</td>
<td>1.180745</td>
</tr>
<tr>
<td>da949af9db66dfb20730f8f171584a71</td>
<td>Google Chrome</td>
<td>6</td>
<td>0.544959</td>
</tr>
<tr>
<td>0b61c673ee71fe9ee725bd687c458b09</td>
<td>Google Chrome</td>
<td>5</td>
<td>0.454133</td>
</tr>
<tr>
<td>ba99c9b667f25ad098115c71c59d29e51</td>
<td>Google Chrome</td>
<td>4</td>
<td>0.363306</td>
</tr>
<tr>
<td>62448833d82302412270c03b7d441e31b</td>
<td>parsecd, apple, geod, apple, photomoments, photoanalysisd, FreedomProxy</td>
<td>4</td>
<td>0.363306</td>
</tr>
<tr>
<td>b4f4e5164f938870485678536f01f9ce</td>
<td>Google Chrome</td>
<td>2</td>
<td>0.181653</td>
</tr>
<tr>
<td>f28d34ce9e732f544de2350027d74c3f</td>
<td>Used by many programs, Quip, Aura, Spotify, Chatty</td>
<td>1</td>
<td>0.090827</td>
</tr>
<tr>
<td>f1c5cf087b959ce3c1bd6285407f89a</td>
<td>Used by many programs on OSX, apple, WebKit, Networking</td>
<td>1</td>
<td>0.090827</td>
</tr>
</tbody>
</table>
## Large Network Example

<table>
<thead>
<tr>
<th>JA3</th>
<th>ClientApplication</th>
<th>count</th>
<th>percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>94c485bca29d5392be53f2b8cf7f4304</td>
<td>Google Chrome</td>
<td>25091171</td>
<td>39.108785</td>
</tr>
<tr>
<td>f58966d34ff9488a83797b55c804724d</td>
<td>Google Chrome</td>
<td>9917687</td>
<td>15.458373</td>
</tr>
<tr>
<td>3e4e87dda5a3162306609b7e330441d2</td>
<td>apple.WebKit.Networking,itunesstored</td>
<td>4099567</td>
<td>6.390016</td>
</tr>
<tr>
<td>bc6c386f480ee97b9d9e52d472b777d8</td>
<td>Google Chrome</td>
<td>2960252</td>
<td>4.614048</td>
</tr>
<tr>
<td>1885aa9927f99ed538ed895d9335995c</td>
<td></td>
<td>2862309</td>
<td>4.461387</td>
</tr>
<tr>
<td>0ffee3ba8e615ad22535e7f771690a28</td>
<td>firefox</td>
<td>1836254</td>
<td>2.862109</td>
</tr>
<tr>
<td>c07cb55f88702033a8f52c046d23e0b2</td>
<td>Used by many programs on OSX,apple.WebKit.Networking</td>
<td>1707345</td>
<td>2.661183</td>
</tr>
<tr>
<td>187dfde7edc8cedcc3deeccc21daeb</td>
<td>eclipse,java,studio,STS</td>
<td>1494424</td>
<td>2.329310</td>
</tr>
<tr>
<td>6cd1b944f5885e2cfbe98a840b75ebeb</td>
<td>Google Chrome</td>
<td>1193719</td>
<td>1.860611</td>
</tr>
<tr>
<td>37f691b063c10372135db21579643bf1</td>
<td></td>
<td>843801</td>
<td>1.315205</td>
</tr>
<tr>
<td>1fbe5382f9d8430fe921df747c46d95f</td>
<td>FieldServiceApp,socialstudio</td>
<td>783763</td>
<td>1.221626</td>
</tr>
<tr>
<td>e4d448cde0dc1243c1eb026c74ac9a</td>
<td>Used by many programs on OSX,apple.WebKit.Networking</td>
<td>717555</td>
<td>1.118429</td>
</tr>
<tr>
<td>c05de18b01a05f2f6900ffe96b3da7a</td>
<td>Used by many programs on OSX,apple.WebKit.Networking</td>
<td>715799</td>
<td>1.115692</td>
</tr>
<tr>
<td>7b343af1092863fdd822d6f10645abfb</td>
<td>apple.WebKit.Networking,itunesstored</td>
<td>429911</td>
<td>0.670088</td>
</tr>
<tr>
<td>JA3</td>
<td>ClientApplication</td>
<td>count</td>
<td>percent</td>
</tr>
<tr>
<td>-----</td>
<td>-------------------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>3e4e87dda5a3162306609b7e330441d2</td>
<td>apple.WebKit.Networking, itunesstored</td>
<td>1354514</td>
<td>96.446367</td>
</tr>
<tr>
<td>c07cb5f88702033a8f52c046d23e0b2</td>
<td>Used by many programs on OSX, apple.WebKit.Networking</td>
<td>18605</td>
<td>1.324744</td>
</tr>
<tr>
<td>bc6c386f480ee97b9d9e52d472b772d8</td>
<td>Google Chrome</td>
<td>14959</td>
<td>1.065136</td>
</tr>
<tr>
<td>ab2382d4d8acbf77e96935ba1e287b7</td>
<td>Google Chrome</td>
<td>4358</td>
<td>0.310306</td>
</tr>
<tr>
<td>6cd1b944f5885e2cfbe98a840b75eeb8</td>
<td>Google Chrome</td>
<td>3837</td>
<td>0.273208</td>
</tr>
<tr>
<td>d5e8fe02b8ad14691619ceb0eba94fc6</td>
<td>Google Chrome</td>
<td>2449</td>
<td>0.174378</td>
</tr>
<tr>
<td>1ad848b33ae442ed4c23356f04a7dc8e</td>
<td></td>
<td>2369</td>
<td>0.168681</td>
</tr>
<tr>
<td>fc5cf087b959cec31bd6285407f689a</td>
<td>Used by many programs on OSX, apple.WebKit.Networking</td>
<td>1895</td>
<td>0.134931</td>
</tr>
<tr>
<td>c56cc8486d51dd06dce2bdf0c8ae9dbc</td>
<td></td>
<td>435</td>
<td>0.030974</td>
</tr>
<tr>
<td>b9ffee599a89a2a54c859098a580fe15</td>
<td></td>
<td>314</td>
<td>0.022358</td>
</tr>
</tbody>
</table>
Malware and Sandboxes
Baseline your sandbox

https://github.com/gbarford/testssl

Win10-socket:          c12f54a3f91dc7bafd92cb59fe009a35
Win10-socket-SNI:      3b5074b1b5d032e5620f69f9f700ff0e
Win10-powershell:      fc54e0d16d9764783542f0146a98b300
Win10-powershell-SNI:  54328bd36c14bd82ddaa0c04b25ed9ad
Win10-iexplore:        be6155e945a3e59a1dd0841b86f6c945
Win10-iexplore-SNI:    10ee8d30a5d01c042afd7b2b205facc4

Win2016-socket:        043c543b63b895881d9abfbc320cb863
Win2016-socket-SNI:    7c410ce832e848a3321432c9a82e972b
Win2016-powershell:    17b69de9188f4c205a00fe5ae9c1151f
Win2016-powershell-SNI:235a856727c14dba889ddee0a38dd2f2
Win2016-iexplore:      4f2e9c50db9bd107439136bd24740c0d
Win2016-iexplore-SNI:  f88610704d61a237aa9e5e0849573998
Moloch
Moloch

molo.ch
## Meterpreter on Windows 10

<table>
<thead>
<tr>
<th>Start Time</th>
<th>Stop Time</th>
<th>Src IP / Country</th>
<th>Src Port</th>
<th>Dst IP / Country</th>
<th>Dst Port</th>
<th>Packets</th>
<th>Databytes</th>
<th>JA3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017/12/22</td>
<td>16:13:05</td>
<td>10.0.0.7</td>
<td>55401</td>
<td>10.0.0.4</td>
<td>443</td>
<td>547</td>
<td>407,793</td>
<td>8916410db83077a5460817142dc6c80e</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>16:13:05</td>
<td>10.0.0.7</td>
<td>55400</td>
<td>10.0.0.4</td>
<td>443</td>
<td>165</td>
<td>209,177</td>
<td>72a589da586844d7f0818ce564948eea</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:54:27</td>
<td>10.0.0.7</td>
<td>55321</td>
<td>10.0.0.4</td>
<td>443</td>
<td>458</td>
<td>309,759</td>
<td>72a589da586844d7f0818ce564948eea</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:54:27</td>
<td>10.0.0.7</td>
<td>55320</td>
<td>10.0.0.4</td>
<td>443</td>
<td>169</td>
<td>209,193</td>
<td>218,451</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:49:25</td>
<td>10.0.0.7</td>
<td>55319</td>
<td>10.0.0.4</td>
<td>443</td>
<td>479</td>
<td>295,069</td>
<td>72a589da586844d7f0818ce564948eea</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:49:24</td>
<td>10.0.0.7</td>
<td>55318</td>
<td>10.0.0.4</td>
<td>443</td>
<td>165</td>
<td>183,068</td>
<td>218,164</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:47:52</td>
<td>10.0.0.7</td>
<td>55307</td>
<td>10.0.0.4</td>
<td>443</td>
<td>97</td>
<td>122,796</td>
<td>72a589da586844d7f0818ce564948eea</td>
</tr>
<tr>
<td>Start Time</td>
<td>Stop Time</td>
<td>Src IP / Country</td>
<td>Src Port</td>
<td>Dst IP / Country</td>
<td>Dst Port</td>
<td>Packets</td>
<td>Databytes / Bytes</td>
<td>JA3</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>------------------</td>
<td>---------</td>
<td>-----------------</td>
<td>---------</td>
<td>---------</td>
<td>------------------</td>
<td>-----</td>
</tr>
<tr>
<td>2017/12/22 16:13:05</td>
<td>2017/12/22 16:13:27</td>
<td>10.0.0.7</td>
<td>55401</td>
<td>10.0.0.4</td>
<td>443</td>
<td>547</td>
<td>407,793 437,919</td>
<td>8916410db85077a5460817142dcbc8de</td>
</tr>
<tr>
<td>2017/12/22 16:13:05</td>
<td>2017/12/22 16:13:05</td>
<td>10.0.0.7</td>
<td>55400</td>
<td>10.0.0.4</td>
<td>443</td>
<td>165</td>
<td>209,177 218,201</td>
<td>72a589da5868444d7f0818ce684948eea</td>
</tr>
<tr>
<td>2017/12/22 15:54:27</td>
<td>2017/12/22 15:54:50</td>
<td>10.0.0.7</td>
<td>55321</td>
<td>10.0.0.4</td>
<td>443</td>
<td>458</td>
<td>309,759 335,035</td>
<td>8916410db85077a5460817142dcbc8de</td>
</tr>
<tr>
<td>2017/12/22 15:54:26</td>
<td>2017/12/22 15:54:27</td>
<td>10.0.0.7</td>
<td>55320</td>
<td>10.0.0.4</td>
<td>443</td>
<td>169</td>
<td>209,193 218,451</td>
<td>72a589da5868444d7f0818ce684948eea</td>
</tr>
<tr>
<td>2017/12/22 15:49:25</td>
<td>2017/12/22 15:50:01</td>
<td>10.0.0.7</td>
<td>55319</td>
<td>10.0.0.4</td>
<td>443</td>
<td>479</td>
<td>295,869 322,305</td>
<td>8916410db85077a5460817142dcbc8de</td>
</tr>
<tr>
<td>2017/12/22 15:49:24</td>
<td>2017/12/22 15:49:25</td>
<td>10.0.0.7</td>
<td>55318</td>
<td>10.0.0.4</td>
<td>443</td>
<td>165</td>
<td>183,068 192,164</td>
<td>72a589da5868444d7f0818ce684948eea</td>
</tr>
<tr>
<td>Start Time</td>
<td>Stop Time</td>
<td>Src IP / Country</td>
<td>Src Port</td>
<td>Dst IP / Country</td>
<td>Dst Port</td>
<td>Packets</td>
<td>Databytes / Bytes</td>
<td>JA3</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>------------------</td>
<td>---------</td>
<td>-----------------</td>
<td>---------</td>
<td>---------</td>
<td>------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:49:25</td>
<td>10.0.0.7</td>
<td>55318</td>
<td>10.0.0.4</td>
<td>443</td>
<td>165</td>
<td>183,068</td>
<td>72a589da586844d7f0818ce684948eea</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:49:24</td>
<td>10.0.0.7</td>
<td>55318</td>
<td>10.0.0.4</td>
<td>443</td>
<td>165</td>
<td>192,164</td>
<td>72a589da586844d7f0818ce684948eea</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:54:26</td>
<td>10.0.0.7</td>
<td>55320</td>
<td>10.0.0.4</td>
<td>443</td>
<td>169</td>
<td>209,193</td>
<td>72a589da586844d7f0818ce684948eea</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:54:27</td>
<td>10.0.0.7</td>
<td>55321</td>
<td>10.0.0.4</td>
<td>443</td>
<td>458</td>
<td>309,759</td>
<td>8916410db85077a5460817142dc8cb8de</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>16:13:05</td>
<td>10.0.0.7</td>
<td>55400</td>
<td>10.0.0.4</td>
<td>443</td>
<td>165</td>
<td>209,177</td>
<td>72a589da586844d7f0818ce684948eea</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>16:13:05</td>
<td>10.0.0.7</td>
<td>55401</td>
<td>10.0.0.4</td>
<td>443</td>
<td>547</td>
<td>407,793</td>
<td>8916410db85077a5460817142dc8cb8de</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:54:50</td>
<td>10.0.0.7</td>
<td>55321</td>
<td>10.0.0.4</td>
<td>443</td>
<td>479</td>
<td>295,869</td>
<td>8916410db85077a5460817142dc8cb8de</td>
</tr>
<tr>
<td>2017/12/22</td>
<td>15:54:27</td>
<td>10.0.0.7</td>
<td>55320</td>
<td>10.0.0.4</td>
<td>443</td>
<td>169</td>
<td>209,193</td>
<td>72a589da586844d7f0818ce684948eea</td>
</tr>
</tbody>
</table>
```
/meterpreter/reverse_https

<table>
<thead>
<tr>
<th>Dst Port</th>
<th>Packets</th>
<th>Databytes</th>
<th>JA3</th>
</tr>
</thead>
<tbody>
<tr>
<td>443</td>
<td>547</td>
<td>407,793</td>
<td>8916410db85077a5460817142dcbec8de</td>
</tr>
<tr>
<td></td>
<td></td>
<td>437,919</td>
<td></td>
</tr>
<tr>
<td>443</td>
<td>165</td>
<td>209,177</td>
<td>72a589da586844d7f0818ce684948eea</td>
</tr>
<tr>
<td></td>
<td></td>
<td>218,201</td>
<td></td>
</tr>
</tbody>
</table>
```
Windows 10 Meterpreter HTTPS

Stager (160-170 packets) JA3:
72a589da586844d7f0818ce684948eea

Payload JA3:
8916410db85077a5460817142dc8de

Stager-SNI (160-170 packets) JA3:
a0e9f5d64349fb13191bc781f81f42e1

Payload-SNI JA3:
ce5f3254611a8c095a3d821d44539877
Powershell Exploit Kits
Powershell Tools (Empire)

No results or none that match your search within your time range.
Powershell Tools (Empire)
Custom Targeted Malware
Secure Sockets Layer

TLSv1.2 Record Layer: Handshake Protocol: Client Hello
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 110

Handshake Protocol: Client Hello
  Handshake Type: Client Hello (1)
  Length: 106
  Version: TLS 1.2 (0x0303)
  Random: b68a3e65dd590910a14d5982da045a63c5bae1f8cbd965e7...
  Session ID Length: 0
  Cipher Suits Length: 2

Cipher Suits (1 suite)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
    Compression Methods Length: 1
    Compression Methods (1 method)
    Extensions Length: 63
    Extension: next_protocol_negotiation (len=0)
    Extension: status_request (len=5)
    Extension: supported_groups (len=4)
    Extension: ec_point_formats (len=2)
    Extension: signature_algorithms (len=14)
    Extension: renegotiation_info (len=1)
    Extension: application_layer_protocol_negotiation (len=5)
    Extension: signed_certificate_timestamp (len=0)
JA3:
87bb7d3dcf10752c52ebeb53f0a57700

Fingerprint String:
771,49196,13172-5-10-11-13-65281-16-18,24,0
Hunting | Alerts | Analysis
### Hunting weird Cert Subjects

#### CertificateSubject

<table>
<thead>
<tr>
<th>Values</th>
<th>Count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN=www.dlink.com,OU=D-Link Corporation,O=D-Link Corporation,L=Asia,ST=Asia,C=TW</td>
<td>3</td>
<td>37.5%</td>
</tr>
<tr>
<td>CN=www.4me27k4toaqg.net</td>
<td>1</td>
<td>12.5%</td>
</tr>
<tr>
<td>CN=www.dlink.com\u005c0D,OU=DHPD Dept.,O=D-LINK,L=Taipie,ST=Taiwan,C=TW</td>
<td>1</td>
<td>12.5%</td>
</tr>
<tr>
<td>CN=www.jahia.com,OU=Jahia,O=Jahia,L=Geneva,ST=Unknown,C=CH</td>
<td>1</td>
<td>12.5%</td>
</tr>
<tr>
<td>CN=www.okrgpc6n32clgswq4e.net</td>
<td>1</td>
<td>12.5%</td>
</tr>
<tr>
<td>CN=www.xsighten.com,OU=Client,O=Ensighten\u005c, Inc.,L=San Jose,ST=California,C=US</td>
<td>1</td>
<td>12.5%</td>
</tr>
</tbody>
</table>
CN=www.okrgpc6n32clgswwq4e.net

<table>
<thead>
<tr>
<th>Values</th>
<th>Count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td>1</td>
<td>100%</td>
</tr>
<tr>
<td>JA3</td>
<td>Domain</td>
<td>PortDestination</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>-----------------</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.zjjm5e31toypxy6kedk2.com">www.zjjm5e31toypxy6kedk2.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.w3sitesgrftbogxrg6jr65.com">www.w3sitesgrftbogxrg6jr65.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.tvci4b2adx2j4ygy7kwyrdy5.com">www.tvci4b2adx2j4ygy7kwyrdy5.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.pqqnumjtf7do.com">www.pqqnumjtf7do.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.pmcn7w.com">www.pmcn7w.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.ali3mbicogyvbyj6.com">www.ali3mbicogyvbyj6.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.madhm3yg7boyw33q7xr3.com">www.madhm3yg7boyw33q7xr3.com</a></td>
<td>9001</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.kzsmnp3.com">www.kzsmnp3.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.ahmmikpyufe.com">www.ahmmikpyufe.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.okuqq.com">www.okuqq.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.2gqcmd4a6.com">www.2gqcmd4a6.com</a></td>
<td>443</td>
</tr>
<tr>
<td>2d8794cb7b52b777bee2695e79c15760</td>
<td><a href="http://www.2ftiovu2k2p3h.com">www.2ftiovu2k2p3h.com</a></td>
<td>9001</td>
</tr>
</tbody>
</table>
No Server, No Problem

Corp Net \rightarrow Internet

NIDS

or something

Hello

\rightarrow AWS
Enrich SSL/TLS Alerting

File Exfil Detection
Exfil Detection Bro Script

Exfil logging based off of:
https://github.com/reservoirolabs/bro-scripts
Bob Rotsted
Normal Outbound Traffic

Bytes/s

Time
File Transfer Outbound

Bytes/s

Time
Threshold Byte Count and Byte Rate
<table>
<thead>
<tr>
<th>src</th>
<th>dst</th>
<th>port</th>
<th>service</th>
<th>cert.subject</th>
<th>bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.2.3</td>
<td>50.1.2.3</td>
<td>443</td>
<td>HTTPS</td>
<td>CN=*.dropbox</td>
<td>2098220</td>
</tr>
</tbody>
</table>
Log File Transfers

Add to script:
Service
Domain
SSL Cert
JA3
JA3ClientApplication

```plaintext
type Info: record {
    ## Domain from SNI
domain: string &log &optional;
    ## Subject of the X.509 certificate
subject: string &log &optional;
    ## JA3 hash
ja3: string &log &optional;

    ...
    if (c?$ssl) rec$subject = c$ssl$subject;
    if (c?$ssl) rec$domain = c$ssl$server_name;
    if (c?$ssl) rec$ja3 = c$ssl$ja3;
}
```
End Result: Valuable Logs

Source IP: 10.1.2.3
Destination IPs: 50.1.2.3, 50.1.2.4, 50.1.2.5 ...
Destination Port: 443
Service: HTTPS
Destination Certificate: CN=*.dropbox.com ...
Certificate Valid: True
Files Transferred: 512
TotalBytes Transferred: 2,048MB
JA3: fa030dbcb2e3c7141d3c2803780ee8db
JA3ClientApplication: Dropbox
End Result: Valuable Logs

Source IP: 10.1.2.3
Destination IPs: 50.1.2.3, 50.1.2.4, 50.1.2.5 ...
Destination Port: 443
Service: HTTPS
Destination Certificate: CN=*.dropbox.com ...
Certificate Valid: True
Files Transferred: 512
TotalBytes Transferred: 2,048MB
JA3: 17b69de9188f4c205a00fe5ae9c1151f
JA3ClientApplication: Powershell
Enrich SSL/TLS Analysis

Evilginx - Phishing 2FA Tokens
Evilginx

Internet

Corp Net

Username
Password
Cookie 2FA

EVILGNIX

NIDS

or something

LinkedIn
## Evilginx

Evilginx is a tool for testing web applications for vulnerabilities. It allows attackers to inject malicious code into websites and then test how the website responds to these attacks.

### Loading Phishlets
Phishlets are the core components of Evilginx that are loaded into the application. They contain the logic for phishing attacks.

```plaintext
[03:46:32] [inf] loading phishlets from: /app/phishlets
[03:46:32] [inf] redirect parameter set to: rs
[03:46:32] [inf] verification parameter set to: xt
[03:46:32] [inf] verification token set to: ae9e
[03:46:33] [inf] unauthorized request redirection URL set to: https://www.youtube.com/watch?v=dQw4w9WgXcQ
[03:46:33] [inf] loaded phishlet 'amazon' made by @customsync from 'amazon.yaml'
[03:46:33] [inf] loaded phishlet 'facebook' made by @mrgretzky from 'facebook.yaml'
[03:46:33] [inf] loaded phishlet 'linkedin' made by @mrgretzky from 'linkedin.yaml'
[03:46:33] [inf] loaded phishlet 'outlook' made by @mrgretzky from 'outlook.yaml'
[03:46:33] [inf] loaded phishlet 'reddit' made by @customsync from 'reddit.yaml'
[03:46:33] [inf] loaded phishlet 'twitter-mobile' made by @white_fi from 'twitter-mobile.yaml'
[03:46:33] [inf] loaded phishlet 'twitter' made by @white_fi from 'twitter.yaml'
```

### Server Domain
The server domain is necessary for Evilginx to function properly.

```plaintext
[03:29:35] [+++] [1] Password: [hello]
[03:29:35] [+++] [1] Username: [test@gmail.com]
```

### Session Table

<table>
<thead>
<tr>
<th>id</th>
<th>phishlet</th>
<th>username</th>
<th>password</th>
<th>tokens</th>
<th>remote ip</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>linkedin</td>
<td></td>
<td></td>
<td>none</td>
<td>172.17.0.1</td>
<td>2018-09-14 02:59</td>
</tr>
<tr>
<td>2</td>
<td>linkedin</td>
<td><a href="mailto:test@gmail.com">test@gmail.com</a></td>
<td>hello</td>
<td>none</td>
<td>172.17.0.1</td>
<td>2018-09-14 03:29</td>
</tr>
</tbody>
</table>
evilginx2 is a man-in-the-middle attack framework used for phishing login credentials along with session cookies, which in turn allows to bypass 2-factor authentication protection.

Released July and gaining momentum.

Written in GO, can be noisy.

JA3: d3e1de2ca313c6c0a639f69cc3e924a4

TODO:
- Need to combine with unencrypted URI for login page
- Need access to HTTP User Agents?
Remember:

Collisions can happen

There are OS APIs
JA3S
Fingerprinting Server Hellos

It’s a thing, but who’s got time for that today.
Conclusion

● JA3 is not a silver bullet
  ○ Collisions can happen
  ○ Applications can connect through OS APIs
  ○ There can be up to 5 JA3s for the same application
  ○ But it is always valuable as a pivot point for analysis
● JA3 is a silver bullet (sometimes)
  ○ Each environment is different
● JA3S adds even more context
● Please contribute and iterate
  ○ Let’s push the industry forward!
JA3 Support
Jeff Atkinson

neslog[at]gmail[dot]com
LinkedIn

https://github.com/salesforce/ja3

bro-pkg install ja3