



 Eleven Paths

La Oscuridad de Certificate Transparency

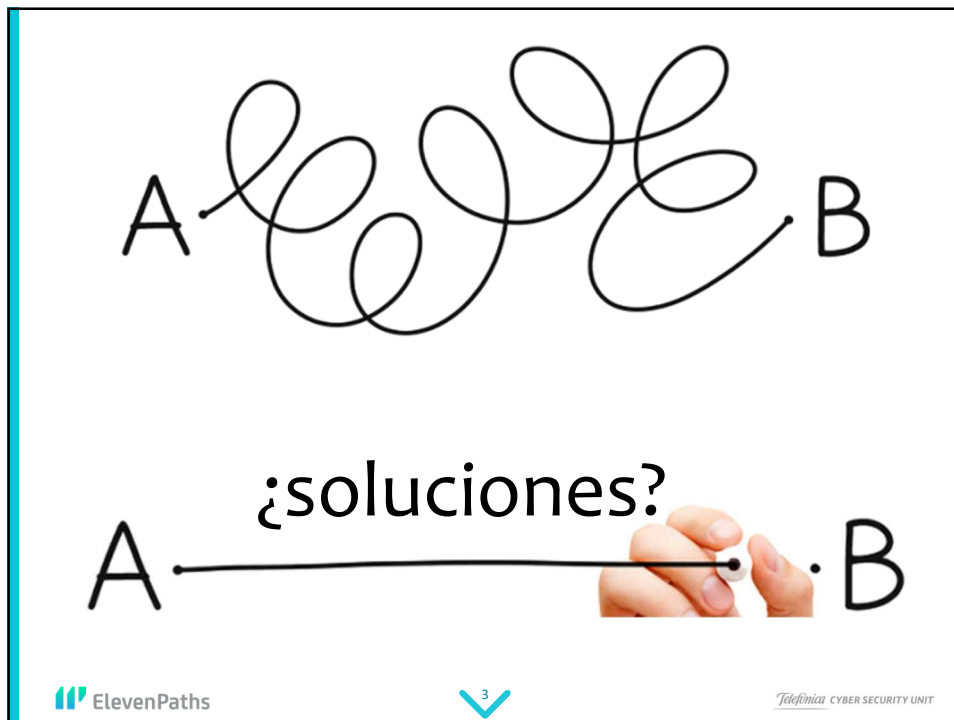
Lic. Cristian Borghello, CISSP - CCSK
@crisborghe / @seguinfo

Sheila Berta
@unapibageek

Telefónica CYBER SECURITY UNIT

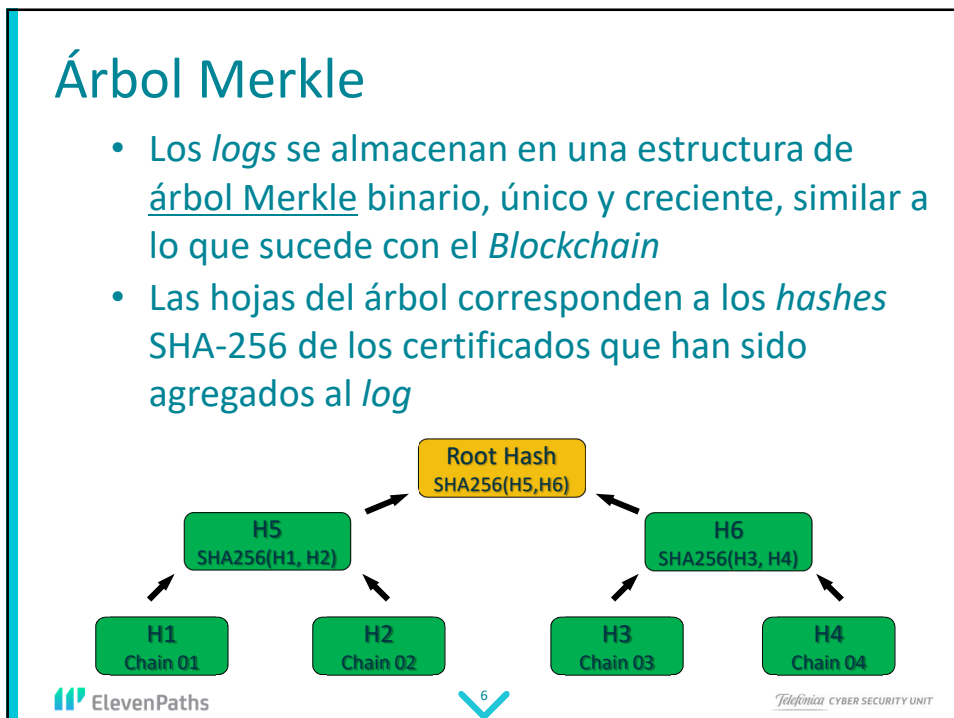
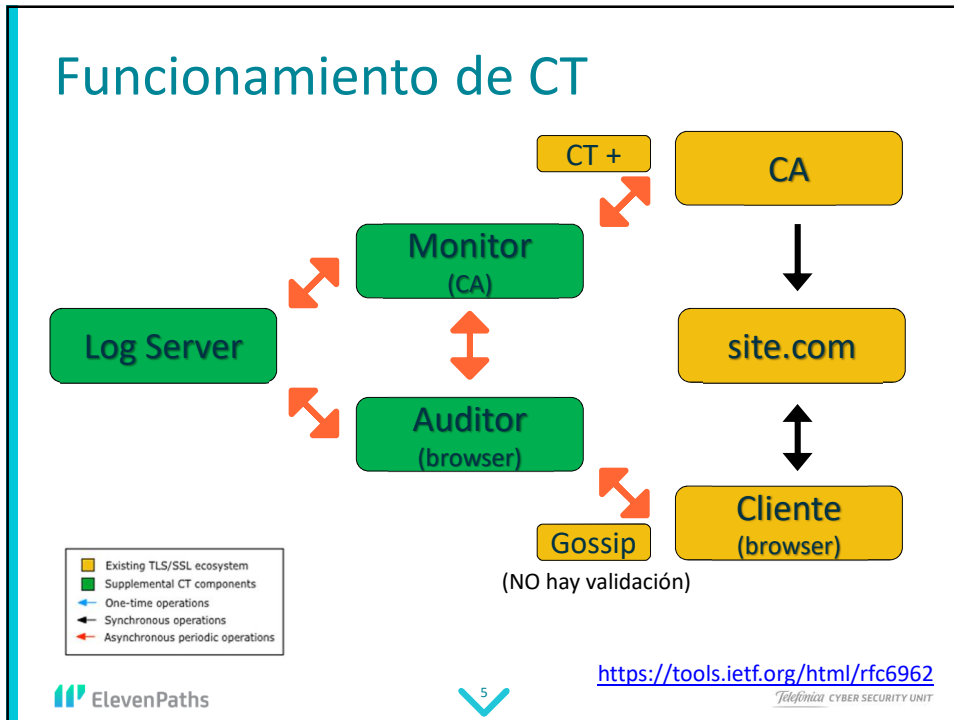
Lo que todos ya sabemos

- ¿Para qué sirve HTTPS?
- ¿Cómo funciona la negociación SSL/TLS?
- ¿Qué ataques existen para SSL/TLS?
- ¿Qué es HSTS?
- ¿Qué es Certificate Pinning?
- **Y la más importante, ¿Qué hacen las CA?**



Certificate Transparency (RFC 6962)

- CT es un protocolo (*gossip*) para registrar públicamente la existencia de certificados TLS a medida que se emiten, de manera que cualquiera pueda auditar la actividad de la CA y notificar la emisión de certificados sospechosos
- La intención es que los navegadores no puedan validar certificados que no aparecen en el registro. Actualmente funciona sólo con EV
- Impulsado por Google desde la versión 45 (09/15) y de obligatorio desde octubre de 2017



Registro SCT (Signed Certificate Timestamp)

X509v3 Extended Key Usage:
 TLS Web Server Authentication, TLS Web Client Authentication
 X509v3 Authority Key Identifier:
 keyid:DB:CF:5C:50:B7:AB:02:1F:15:17:AA:16:BB:0D:B5:28:9D:6A:5A:F3

Authority Information Access:
 OCSP - URI:http://gm.symantec.com
 CA Issuers - URI:http://gm.symcb.com/gm.crt

CT Precertificate SCTs:
 Signed Certificate Timestamp:
 Version : 02(0)
 Log Name : Symantec Log Server
 Log ID : DD:BB:1D:2B:7A:0D:4F:A6:20:8B:81:AD:81:68:70:7E:
 2B:9B:8D:01:6D:5C:8B:8D:20:11:04:CD:86:BC:8E:CC
 Timestamp : Oct 20 09:35:05.702 2016 GMT
 Extensions: none
 Signature : ecda8a-wLtb-8HA256
 30:46:02:21:0D:CD:65:A5:B1:D8:65:DD:B6:06:4D:1B:
 0E:84:3A:95:B8:F6:AA:60:35:3F:75:13:5A:7B:82:6F:
 F9:70:08:F1:1D:02:21:00:F7:3D:9E:49:4C:03:36:D4:
 AD:56:39:BD:ED:54:09:8F:71:06:25:7B:BF:D1:BB:23:
 27:F6:50:F1:4B:86:A6:7C

Signed Certificate Timestamp:
 Version : 02(0)
 Log Name : Google 'Aviator' log
 Log ID : 68:F6:98:F8:1F:64:82:BB:3A:8C:BB:89:28:1D:4C:FC:

ct.ws.symantec.com

Base64 Log ID: 3esdK3oNT6Ygi4GtGWhwfi6OnQHVXiINPRHEzbbvsVsw=
 Operator: Symantec
 Submitted for inclusion in Chrome: 2015-05-1
 HTTPS supported: yes
 Maximum Merge Delay: 24 hours
 Contact: DL-ENG-Symantec-CT-Log@symantec.com
 Chrome inclusion status: Included (since M45).

ct.googleapis.com/aviator

Base64 Log ID: aPaY+B9kgr46j065KB1M/HFRXWt1ETRCmesu09P+8Q=
 Operator: Google
 Started: 2013-09-30
 HTTPS supported: yes
 Contact: google-ct-logs@googlegroups.com
 Chrome inclusion status: Frozen.

Lugar del SCT

- Como una Extensión X.509v3: la CA lo “incrusta” al crear el certificado (OID: 1.3.6.1.4.1.11129.2.4.2)

Lugar del SCT

- **A través de la extensión 0x12 del protocolo TLS:** la CA entrega un certificado normal y el administrador del servidor debe enviarlo al *log* de forma manual
- **En la respuesta OCSP:** la CA envía el certificado al sitio y al *log* y luego el servidor realiza una petición OCSP a la CA



El código fuente

```

https://chromium.googlesource.com/chromium/src/+master/net/cert/ct_known_logs_static-inc.h
24 // The set of all presently-qualifying CT logs.
25 // Google provides DNS frontends for all of the logs.
26 const CTLogInfo kCTLogList[] = {
27     {"\x30\x59\x30\x13\x06\x07\x2a\x86\x48\xce\x3d\x02\x01\x06\x08\x2a\x86"
28      "\x48\xce\x3d\x03\x01\x07\x03\x42\x00\x04\x7d\xa8\x4b\x12\x29\x80\xa3"
29      "\x3d\xad\x35a\x77\xb8\xcc\xe2\x88\xb3\xa5\xfd\xfd\x3c\x0c\xcd\x18"
30      "\x0c\xe8\x41\x46\xe8\x81\x01\x1b\x15\xe1\x4b\xf1\x1b\x62\xdd\x36\xa0a"
31      "\x08\x18\xba\xed\x0b\x35\x84\xd0\x9e\x40\x3c\x2d\x9e\x9b\x82\x65\xbd"
32      "\x1f\x04\x10\x41\x4c\xa0",
33      91, "Google 'Pilot' log", "https://ct.googleapis.com/pilot/",
34      "pilot.ct.googleapis.com"},
35     {"\x30\x59\x30\x13\x06\x07\x2a\x86\x48\xce\x3d\x02\x01\x06\x08\x2a\x86"
36      "\x48\xce\x3d\x03\x01\x07\x03\x42\x00\x04\x7d\xf4\xcc\x69\xb2\xe4\xe0"
37      "\x90\xa3\x8a\xea\x5a\x70\x09\x4f\xef\x13\x62\xd0\x8d\x49\x60\xff\x1b"
38      "\x40\x50\x07\x0c\x6d\x71\x86\xda\x25\x49\x8d\x65\xe1\x08\x0d\x47\x34"
39      "\x6b\xbd\x27\xbc\x96\x21\x3e\x34\xf5\x87\x76\x31\xb1\x7f\x1d\x9c\x85"
40      "\x3b\x0d\xf7\x1f\x3f\xe9",
41      91, "Google 'Aviator' log", "https://ct.googleapis.com/aviator/",
42      "aviator.ct.googleapis.com"},

```

La cantidad de días de verificación

shopping .net
whois information

Whois Website Info History DNS Records Diagnostics

cache expires in 19 hours, 16 minutes and 52 seconds

Registrar Info

Name	Ascio Technologies, Inc
Whois Server	whois.ascio.com
Referral URL	http://www.ascio.com
Status	clientTransferProhibited http://www.icann.org/epp

Important Dates

Expires On	2018-03-17	Hoy es 27/03 (pasaron 10 días)
Registered On	2017-03-17	
Updated On	2017-03-17	

openssl s_client -connect shopping .net:443
notBefore=Apr 9 00:00:00 2014 GMT
notAfter=Apr 8 23:59:59 2017 GMT

La cantidad de días de verificación

ct-lta.tk

cert.sh ID	109353487			
Summary	Leaf certificate			
Certificate Transparency	Timestamp	Entry #	Log Operator	Log URL
	2017-03-27 15:21:30 GMT	81642411	Google	https://ct.googleapis.com/rocketeer
	2017-03-27 15:21:30 GMT	848036	WoSign	https://ctlog.wosign.com
	2017-03-27 15:21:31 GMT	787477	Google	https://ct.googleapis.com/skydiver
2017-03-29 17:07:46 GMT	84651209	Google	https://ct.googleapis.com/pilot	

cert.sh ID	1124			
Summary	Leaf certificate			
Certificate Transparency	Timestamp	Entry #	Log Operator	Log URL
	2017-04-04 07:35:03 GMT	41220789	Google	https://ct.googleapis.com/icarus
	2017-04-06 22:45:09 GMT	235200	Venafi	https://ctlog-gen2.api.venafi.com

CT no se usará para revocar certificados, sólo se usará para detectar el uso inapropiado de los mismos!

¿Esto va en serio?

Add support for `Expect-CT` header #224

Open craigfrancis opened this issue 22 days ago · 7 comments



craigfrancis commented 22 days ago

Contributor

Might be a little early yet, but the `Expect-CT` header is likely to be implemented in Chrome and Firefox:

<https://www.chromestatus.com/feature/567711733430272>

```
Expect-CT: enforce;
          max-age=1440;
          report-uri=https://blah.com/CTReport/
```

enforce: el navegador espera un SCT válido o aborta la conexión

max-age: tiempo de caché de esta directiva en el navegador (en seg.)

report-uri: URL a la que el navegador deber enviar reportes de fallos

<https://tools.ietf.org/id/draft-stark-expect-ct-01.txt>

Nuestro aporte

```

function extractSCTData() {
    var payloadLength = 0;
    var pos = sctSequence.indexOf(sctOid) + sctOid.length + 3;
    var length = sctSequence.substr(pos, 1);
    pos += parseInt(length, 16)*2 + 4;
    length = sctSequence.substr(pos, 1);
    pos += parseInt(length, 16)*2 + 1;
    payloadLength = parseInt(sctSequence.substr(pos,4), 16)*2;
    pos += 4;

    var i;
    for(i = 0; pos < sctSequence.length; i++) {
        pos += 4;
        sctVersion[i] = sctSequence.substr(pos, 2);
        pos += 2;
        sctLogId[i] = sctSequence.substr(pos, 64);
        pos += 64;
        sctTimestamp[i] = sctSequence.substr(pos, 16);
        pos += 16;
        sctExtensions[i] = sctSequence.substr(pos, 4);
        pos += 4;
        sctSignatureAlgorithm[i] = sctSequence.substr(pos, 4);
        pos += 4;
        length = sctSequence.substr(pos, 4);
        pos += 4;
        sctSignature[i] = sctSequence.substr(pos, parseInt(length, 16)*2);
        pos += parseInt(length, 16)*2;
    }
}
    
```

Y...

En el paper hay una sorpresa 😊

Una aproximación práctica al Certificate Transparency

<https://bit.ly/certificate-transparency>







Implementación en Chrome (cont.)

ct.googleapis.com = 64.233.186.95

<p>Remote IP address</p> <p><input type="radio"/> Any IP address</p> <p><input checked="" type="radio"/> These IP addresses:</p> <p>64.233.190.95 <input type="button" value="Add"/></p>	<p>Action</p> <p><input type="radio"/> Allow the connection</p> <p><input type="radio"/> Allow the connection if it is secure</p> <p><input type="button" value="Customize..."/></p> <p><input checked="" type="radio"/> Block the connection</p>
--	---

No se bloquea el funcionamiento, ya que la verificación **no** se realiza a través de esta conexión

Google verifica de forma asíncrona para agregar más sitios a sus servidores de Transparencia

Implementación en Chrome (cont.)

The screenshot shows a Windows Explorer window with the address bar path: This PC > Windows (C:) > Users > Usuario > AppData > Local > Google > Chrome > User Data. The main pane shows a list of folders: CertificateTransparency (4/3/2017 10:20 PM, File folder) and Crashpad (3/20/2017 8:06 PM, File folder). Below, the CertificateTransparency folder is expanded, showing a list of files with names like 34bb6adfc3d9c03ee8a499f7891486c9d5e5ca92d01f7bf1d1bce19db48ef.sth, all dated 3/31/2017 3:11 AM and 1 KB in size. The breadcrumb path at the top of the file list is: Users > Usuario > AppData > Local > Google > Chrome > User Data > CertificateTransparency > 342 > .platform_specific > all > sths.

Implementación en Chrome (cont.)

File: sth_set_component_installer.h

```
67 // Reads and parses the on-disk json.
68 void LoadSTHsFromDisk(const base::FilePath& sths_file_path,
69                      const base::Version& version);
70
```

```
115
116 void STHSetComponentInstallerTraits::LoadSTHsFromDisk(
117     const base::FilePath& sths_path,
118     const base::Version& version) {
119     if (sths_path.empty())
120         return;
121
```



Implementación en Chrome (cont.)

```
121
122 base::FileEnumerator sth_file_enumerator(sth_path, false,
123                                         base::FileEnumerator::FILES,
124                                         FILE_PATH_LITERAL("*.sth"));
125 base::FilePath sth_file_path;
126
127 while (!(sth_file_path = sth_file_enumerator.Next()).empty()) {
128     DVLOG(1) << "Reading STH from file: " << sth_file_path.value();
129
130     const std::string log_id_hex =
131         sth_file_path.BaseName().RemoveExtension().MaybeAsASCII();
132     if (log_id_hex.empty()) {
133         DVLOG(1) << "Error extracting log_id from: "
134                 << sth_file_path.BaseName().LossyDisplayName();
135         continue;
136     }
137
138     std::vector<uint8_t> decoding_output;
139     if (!base::HexStringToBytes(log_id_hex, &decoding_output)) {
140         DVLOG(1) << "Failed to decode Log ID: " << log_id_hex;
141         continue;
142     }
```

Conclusiones

- *Work in progress...* veremos en octubre
- CT debería ayudar a la detección de certificados fraudulentos en un mejor tiempo
- CT no reemplaza nada de certificados digitales, lo complementa
- Más trabajo para los monitores (CA)
- Solo Google por ahora ha implementado CT y su funcionamiento sigue siendo un poco oscuro
- CT es una fuente de información valiosa que puede ser explotada (para bien o mal)

