

# WoSign Incidents Final Report

(September 16, 2016)

WoSign received an email from Mozilla for 3 incidents on August 24<sup>th</sup> 2016. WoSign responded to that email and subsequent emails from Mozilla-Dev-Security-Policy mailing list trying to clarify the issues. Due to the number of email threads, WoSign released a [Report](#) on Sept. 4<sup>th</sup> 2016 explaining what had happened and what was done to fix the issues. Later on, Mozilla, created a wiki page ([https://wiki.mozilla.org/CA:WoSign\\_Issues](https://wiki.mozilla.org/CA:WoSign_Issues)) recapping all the issues and asking WoSign to provide additional information, especially for some of them.

This is the final report for all related issues that have been listed in the Mozilla wiki page, WoSign would like to make it transparent for everybody to know what has happened and what was done to fix them and to prevent those issues happen again in the future.

For a better understanding and transparency, WoSign posted all 2015 issued SSL certificates to Google CT log server and WoSign CT log server, making a total of **101,485** certificates. And recently WoSign also published all certificates that have been issued from Jan. 1<sup>st</sup> 2016 to July 4<sup>th</sup> 2016, making a total of **94,073** certificates.

Even more, since July 5<sup>th</sup> 2016, WoSign decided to post all issued SSL certificate to CT log servers and embedded the SCT data in the certificate meaning that all WoSign issued SSL certificates are now in the CT log server for full transparency.

WoSign is truly aware of the severity and importance of these issues and that's why was investigated deeply and thoroughly in all of our systems trying to find evidences of what has happened. WoSign knows the importance to follow the standards and policies of the CAB Forum and browsers and this has been always WoSign goal. Some of the issues below were not a violation of these norms as has been pointed out, but in any case, and for having best and secure practices, solutions have been applied.

## 1. Issue D: Long-Lived SHA-1 Certs (Jan - Mar 2015)

*(a.k.a. "Issue -2")*

Between 16th January 2015 and 5th March 2015, WoSign issued 1,132 SHA-1 certificates whose validity extended beyond 1st January 2017. This is documented in their [BR audit](#).

### WoSign Response

#### 1.1. What happened

WoSign issued 1,132 SHA-1 certificates from Jan. 16<sup>th</sup> 2015 to Mar. 5<sup>th</sup> 2015. WoSign reported this

issue to the WebTrust auditor, and this incident is included in the 2015 WebTrust BR report that was sent to all browsers.

WoSign was aware of the issue and the reason why this was done and thus notified to the auditor to include in the report for transparency

### **1.2. Why this happened**

The BRs recommended all CAs for not issuing SHA-1 SSL certificates with a validity period beyond January 2017, starting on 16 Jan 2015, so WoSign started to update its PKI system when CAB Forum ballot 118 passed, but due to unexpected delays in the systems upgrade, WoSign could not finish it until Mar. 5<sup>th</sup> 2015.

During this time WoSign was aware to be issuing certificates beyond that end date

### **1.3. What has been done**

WoSign decided to contact all “affected” customers and offer a revocation and replacement to provide them a new one with SHA-2. Up to now, only 171 certificates have been revoked and replaced to SHA-2, for the remaining WoSign is waiting, knowing the issues these customers may have due to conversations with them, for example, subscriber’s web server or equipment obsolete, upgrading pending, ....

Anyway, WoSign will wait and try to replace these remaining certs but if no answer is received and following the decision taken and presented to the auditor, WoSign will revoke those certificates before Dec. 31<sup>th</sup> 2016.

### **1.4. Current situation**

As indicated above, WoSign updated their systems at Mar. 5<sup>th</sup> 2015 that no more SHA-1 certificates can be issued.

## **2. Issue F: Certs Identical Except For NotBefore (Mar 2015)**

---

WoSign issued two certificates in March 2015. These certificates are identical in all ways (including their serial numbers) except for their notBefore dates, which are 37 seconds apart.

- [Cert 1](#)
- [Cert 2](#)

### **WoSign Response**

## 2.1 What happened

This incident is reported in the “Incidents involving the CA WoSign” mail list thread.

-----Original Message-----

From: Peter Bowen [mailto:[pbowen@gmail.com](mailto:pbowen@gmail.com)]

Sent: Friday, September 2, 2016 11:21 PM

To: Richard Wang <[richard@wosign.com](mailto:richard@wosign.com)>

Cc: Ryan Clayton <[ryan@clayton.com](mailto:ryan@clayton.com)>; [mozilla-dev-security-policy@lists.mozilla.org](mailto:mozilla-dev-security-policy@lists.mozilla.org)

Subject: Re: Incidents involving the CA WoSign

Richard,

It seems then there is a newly exposed bug.

<https://www.censys.io/certificates/e2665bb07940b5bee73145f47c99dcf5781edbe9d78f9cada8f1d702d5e340ad> shows a certificate issued by your CA that has a notBefore in March 2015. It does not appear in the CT log. However another certificate with identical serial number and subject, but different Validity, does appear in the log.

Are you aware of a bug where you were issuing certificates identical except for validity period?

This issue with same serial number in certificates with same subject information but different signing time was detected internally time ago, in fact this is not the unique case because there are 16 similar issues. This was caused by the CMS (Certificate Management System), when it sent the signing request of the certificate to the signing server A, which had no response, then the CMS sent it to the other newly added signing server B. After a while the signing server A signed the certificate and sent to the CMS and also to the subscriber, then the subscriber installed the cert in its website and that’s why Censys recorded this certificate; in the meantime, the signing server B also signed this certificate some time later (in seconds) and sent it to the CMS, the CMS accepted it and rewrote it in the DB.

Of course the subscriber didn’t know this issue, and only the first signed one, cert 1, is known by the subscriber and public. But internally, cert 2 replaced cert 1, so in the CMS and PKI DB only cert 2 appeared. When decided to publish all issued certificates to CT log server then only cert 2 was published because replaced cert 1, which is the one in the subscriber website. WoSign only had internally cert 2.

## 2.2 Why this happened

This issue happened after adding another signing server on **Jan 5<sup>th</sup> 2015**, and found it on **April 9<sup>th</sup> 2015**. When had the two signing servers added a load balancer, but the configuration was not properly done because it didn’t lock the request.

This case also exposed a bug in the CMS that didn’t lock the order’s record after getting the signed certificate.

Here is the crt.sh link for all certificates with apart time (seconds) and issued time:

Crt.sh link	Issued time 1	Issued time 2	Apart time
<a href="https://crt.sh/?serial=6c5f294a0b7838a51b96d33adf3fb774">https://crt.sh/?serial=6c5f294a0b7838a51b96d33adf3fb774</a>	2015/1/5 8:08:23	2015/1/5 8:08:30	7

<a href="https://crt.sh/?serial=25e5400b6bffdbd74c8ce82878b44188">https://crt.sh/?serial=25e5400b6bffdbd74c8ce82878b44188</a>	2015/1/10 15:07:41	2015/1/10 15:09:08	87
<a href="https://crt.sh/?serial=1e3b88dac8846ea1a268f93c911a19e4">https://crt.sh/?serial=1e3b88dac8846ea1a268f93c911a19e4</a>	2015/1/21 13:25:24	2015/1/21 13:25:30	6
<a href="https://crt.sh/?serial=12fe10c1ea652c1472eb6e17a42e5ecd">https://crt.sh/?serial=12fe10c1ea652c1472eb6e17a42e5ecd</a>	2015/3/10 7:30:38	2015/3/10 7:30:43	5
<a href="https://crt.sh/?serial=1a8121304718800638ea6899ede21d45">https://crt.sh/?serial=1a8121304718800638ea6899ede21d45</a>	2015/3/10 7:35:09	2015/3/10 7:35:15	6
<a href="https://crt.sh/?serial=320a71d69fcff608e87766b918ecd07a">https://crt.sh/?serial=320a71d69fcff608e87766b918ecd07a</a>	2015/3/10 12:33:28	2015/3/10 12:33:36	8
<a href="https://crt.sh/?serial=177f1598e211b6ad3877d2a8729a48e1">https://crt.sh/?serial=177f1598e211b6ad3877d2a8729a48e1</a>	2015/3/11 13:52:21	2015/3/11 13:55:45	204
<a href="https://crt.sh/?serial=50bbf6f23fb69c2b541bc75e09f02ff7">https://crt.sh/?serial=50bbf6f23fb69c2b541bc75e09f02ff7</a>	2015/3/13 14:32:20	2015/3/13 14:33:07	47
<a href="https://crt.sh/?serial=32e2b62f4467a6ed69c01d87675dc93f">https://crt.sh/?serial=32e2b62f4467a6ed69c01d87675dc93f</a>	2015/3/13 14:34:53	2015/3/13 14:35:05	12
<a href="https://crt.sh/?serial=12df3f24d92bb30798d68e9c477a48a2">https://crt.sh/?serial=12df3f24d92bb30798d68e9c477a48a2</a>	2015/3/13 14:41:51	2015/3/13 14:42:04	13
<a href="https://crt.sh/?serial=46a20ef79b21d30b015aeac84b22747b">https://crt.sh/?serial=46a20ef79b21d30b015aeac84b22747b</a>	2015/3/13 14:42:27	2015/3/13 14:42:50	23
<a href="https://crt.sh/?serial=1115b96b885686f5dbb014150dc23b91">https://crt.sh/?serial=1115b96b885686f5dbb014150dc23b91</a>	2015/3/13 14:43:24	2015/3/13 14:43:55	31
<a href="https://crt.sh/?serial=12529e7bef44097e49c5d2800070f0ea">https://crt.sh/?serial=12529e7bef44097e49c5d2800070f0ea</a>	2015/3/13 14:44:20	2015/3/13 14:44:46	26
<a href="https://crt.sh/?serial=4c139e39e6050269681ca264f1429fac">https://crt.sh/?serial=4c139e39e6050269681ca264f1429fac</a>	2015/3/13 14:46:56	2015/3/13 14:47:05	9
<a href="https://crt.sh/?serial=4f6b5d70f0d9fcdda6347a006f3e98c5">https://crt.sh/?serial=4f6b5d70f0d9fcdda6347a006f3e98c5</a>	<b>2015/3/13 14:50:43</b>	<b>2015/3/13 14:51:20</b>	<b>37</b>
<a href="https://crt.sh/?serial=4b6e1b196ad9e713122a34f118605683">https://crt.sh/?serial=4b6e1b196ad9e713122a34f118605683</a>	2015/4/9 7:24:07	2015/4/9 7:24:08	1

### 2.3. What has been done

WoSign fixed the load balance system and changed the signing mechanism, and updated the configuration of the load balancer in such a way that now, all signing request from CMS will write to database first, and the signing server will get the signing task from the database. If one request is assigned to one signing server, it will be locked exclusively.

In the CMS side, it will reject the signed certificate return from PKI system after received, this can prevent the PKI signing server still send the signed-again certificate in any case. The idea behind this is that the CMS will reject the cert 2 because the CMS already sent the cert 1 to the subscriber

### 2.4. Current situation

This is an incident caused by system bug and wrong configuration, but all certificates were well validated

WoSign considers that there's no reason to revoke these certificates because cert 2 was never used in public, and only aware of it when published.

## 3. Issue H: Duplicate Serial Numbers (Apr 2015)

(a.k.a. "Issue X")

Between 9th April 2015 and 14th April 2015, WoSign issued 392 certificates with duplicate serial numbers, across a handful of different serial numbers. Here is one example. This is documented in their most recent BR audit.

## WoSign Response

### 3.1. What happened

WoSign issued 392 certificates with several duplicated serial numbers from April 9<sup>th</sup> 2015 to April 14<sup>th</sup> 2015. WoSign reported this to the WebTrust auditor, and this incident is included in the 2015 WebTrust BR report that was sent to all browsers.

### 3.2. Why this happened

This issue can be divided into 2 different ones because the source of the issue is different even the result is the same.

Firstly 313 certificates and secondly 27 certificates were affected by a system bug with the serial number generation, generating a serial number starting with “0” in the first left position. The signing system had a bug that didn’t know how to deal with this kind of serial number.

The others were due to a load balancer configuration issue with the two signing servers, different from the above one, because this time signed two certificates with the same time using the same serial number, for example: <https://crt.sh/?serial=112A93A547BC6A7701A2BBDD0B4E67FF>, the issuing time is as “Apr 9 08:12:40 2015 GMT”

### 3.3. What has been done

One of our customers notified us of this incidence on April 14<sup>th</sup> 2015, then after internal checks and reviews an email was sent to these subscribers offering a replacement within one week. Those certificates with duplicated serial numbers were revoked from April 21<sup>st</sup> 2015 till to April 28<sup>th</sup> 2015.

WoSign revoked the 392 certificates. See below details:

CT log URL in crt.sh	Quantity
<a href="https://crt.sh/?serial=56D1570DA645BF6B44C0A7077CC6769">https://crt.sh/?serial=56D1570DA645BF6B44C0A7077CC6769</a>	313
<a href="https://crt.sh/?serial=D3BBDC3A0175E38F9D0070CD050986A">https://crt.sh/?serial=D3BBDC3A0175E38F9D0070CD050986A</a>	27
<a href="https://crt.sh/?serial=112A93A547BC6A7701A2BBDD0B4E67FF">https://crt.sh/?serial=112A93A547BC6A7701A2BBDD0B4E67FF</a>	2
<a href="https://crt.sh/?serial=12E9B15E3FF1CDED3EC86BF132063AB8">https://crt.sh/?serial=12E9B15E3FF1CDED3EC86BF132063AB8</a>	2
<a href="https://crt.sh/?serial=16E60CC1BEFE5C243F65AD7D85F9328D">https://crt.sh/?serial=16E60CC1BEFE5C243F65AD7D85F9328D</a>	2
<a href="https://crt.sh/?serial=1FD8B07664CEEE58E779AA200AF0A9A6">https://crt.sh/?serial=1FD8B07664CEEE58E779AA200AF0A9A6</a>	2

<a href="https://crt.sh/?serial=20AD94DE69A0EE25428BB6CA5EAF0395">https://crt.sh/?serial=20AD94DE69A0EE25428BB6CA5EAF0395</a>	2
<a href="https://crt.sh/?serial=23876A3F46E65EB83500914B87409C2C">https://crt.sh/?serial=23876A3F46E65EB83500914B87409C2C</a>	2
<a href="https://crt.sh/?serial=29C6BC8782E004B26BED594FBEF3122B">https://crt.sh/?serial=29C6BC8782E004B26BED594FBEF3122B</a>	2
<a href="https://crt.sh/?serial=2F7099A8DF3EBE2E0DEBD6DC9BB908B9">https://crt.sh/?serial=2F7099A8DF3EBE2E0DEBD6DC9BB908B9</a>	2
<a href="https://crt.sh/?serial=302388004A3479660553EF2A79A44B78">https://crt.sh/?serial=302388004A3479660553EF2A79A44B78</a>	2
<a href="https://crt.sh/?serial=3045B73FE2FCCB51CC5DA9887B4C6ED0">https://crt.sh/?serial=3045B73FE2FCCB51CC5DA9887B4C6ED0</a>	2
<a href="https://crt.sh/?serial=318FE0819849A5F9C1D1854C7465D9A5">https://crt.sh/?serial=318FE0819849A5F9C1D1854C7465D9A5</a>	2
<a href="https://crt.sh/?serial=35CE1A681804BC007A30513A4AA042AE">https://crt.sh/?serial=35CE1A681804BC007A30513A4AA042AE</a>	2
<a href="https://crt.sh/?serial=390DE9D519163B3D1976A20CF484C515">https://crt.sh/?serial=390DE9D519163B3D1976A20CF484C515</a>	2
<a href="https://crt.sh/?serial=3B9B2E8670B65ECFABAF06345346CAD8">https://crt.sh/?serial=3B9B2E8670B65ECFABAF06345346CAD8</a>	2
<a href="https://crt.sh/?serial=4333FA3C03F27340425E3E44C62DF6F4">https://crt.sh/?serial=4333FA3C03F27340425E3E44C62DF6F4</a>	2
<a href="https://crt.sh/?serial=44E44D23471518D4A9E1C91AC30DA77B">https://crt.sh/?serial=44E44D23471518D4A9E1C91AC30DA77B</a>	2
<a href="https://crt.sh/?serial=47E3642E70C6463C03A6A23B4A7B98A9">https://crt.sh/?serial=47E3642E70C6463C03A6A23B4A7B98A9</a>	2
<a href="https://crt.sh/?serial=49D978B3CF1229BE1B32B537E53C8972">https://crt.sh/?serial=49D978B3CF1229BE1B32B537E53C8972</a>	2
<a href="https://crt.sh/?serial=4CEDEAE1148775B749EACAF890DAB4C0">https://crt.sh/?serial=4CEDEAE1148775B749EACAF890DAB4C0</a>	2
<a href="https://crt.sh/?serial=4F8226EF661F99B282561B3B308B6186">https://crt.sh/?serial=4F8226EF661F99B282561B3B308B6186</a>	2
<a href="https://crt.sh/?serial=504261656EE8447B56166E77BFE52548">https://crt.sh/?serial=504261656EE8447B56166E77BFE52548</a>	2
<a href="https://crt.sh/?serial=527D513DC9806DAEC778D1B876B3B764">https://crt.sh/?serial=527D513DC9806DAEC778D1B876B3B764</a>	2
<a href="https://crt.sh/?serial=5F1AAF998410578D64D0A783BCDFA8C1">https://crt.sh/?serial=5F1AAF998410578D64D0A783BCDFA8C1</a>	2
<a href="https://crt.sh/?serial=67034EA2ED2A1DF42C563F577911FD45">https://crt.sh/?serial=67034EA2ED2A1DF42C563F577911FD45</a>	2
<a href="https://crt.sh/?serial=6B87C339A14F8CEA541421883BB72085">https://crt.sh/?serial=6B87C339A14F8CEA541421883BB72085</a>	2
<a href="https://crt.sh/?serial=6C311A6B744F8CC66794F53ADF99CE60">https://crt.sh/?serial=6C311A6B744F8CC66794F53ADF99CE60</a>	2

### 3.4. Current situation

The 392 certificates are revoked. For the first case scenario problem, WoSign fixed the code that can recognize the “0” at first in the serial number.

For the second case, WoSign added more checking and verification in the system’s code before signing for each signing server.

At the same time WoSign have improved its internal quality control system that all certificates must pass these checks before sending to subscribers.

## 4. Issue J: Various BR Violations (Apr 2015)

(a.k.a. "Issue -1")

On April 3<sup>rd</sup> 2015, WoSign was contacted by Google, who were concerned about Baseline Requirements violations in recently-issued certificates from WoSign. Instead of specifying the violations directly, Google asked WoSign to check their certificates against their CPS.

### WoSign Response

#### 4.1. What happened

WoSign was notified by email from Google on Apr. 4<sup>th</sup> 2015 09:25AM about an issue regarding some violations of the BRs. WoSign's CEO replied Google to check it asap.

From: Richard Wang  
Sent: Saturday, April 4, 2015 11:05 AM  
To: [liyan@cs.cmu.edu](mailto:liyan@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu); [rls@cs.cmu.edu](mailto:rls@cs.cmu.edu)  
Subject: Re: WoSign Irregularities

Hi Ryan,

For irregularities, we will check it carefully next week.  
Thanks.

Regards,

Richard

> On Apr 4, 2015, at 09:25, [liyan@cs.cmu.edu](mailto:liyan@cs.cmu.edu) wrote:  
>  
> Hi Richard,  
>  
> It's come to our attention that WoSign may be issuing certificates that are  
> not conforming to your CPS and not conforming to the Baseline Requirements.

The main problem was due to adding a description in the free SSL certificate in two languages:

```
CN = wosign@11labs.com  
Description = Free SSL Cert apply URL: https://buy.wosign.com/free
```

```
CN = myoopah.org  
Description = Apply Free SSL Certificate at https://buy.wosign.com
```

```
CN = raf@lcz.de  
Description = 免费SSL证书 申请网址 : https://buy.wosign.com/free
```

```
CN = cle@ky.me  
Description = 免费SSL证书 申请网址 : https://buy.wosign.com
```





by opening a high-numbered port (>50,000) and getting WoSign to use that port for validation of control.

This problem was reported by Google, and WoSign resolved. Mozilla only became aware of it recently.

## WoSign Response

### 5.1. What happened

WoSign got report from Google at 8:55 AM April 24<sup>th</sup> 2015 that point out this high port problem that allowed the applicant to choose any port for website control validation. Richard Wang replied Google email within **2 minutes**, and promised to fix this bug within **1 hour**. Richard sent email to Google at 10:09AM after fixed the bug.

We searched our certificates orders from January 10<sup>th</sup> 2015 to April 24<sup>th</sup> 2015, there were 72 certificates issued using higher numbered ports website control validation, those certificates were validated by website control validation\* method that using other port instead of 80 and 443.

\* “Website Control Validation” means subscriber must upload the html file with verification code into its website root directory.

### 5.2. Why this happened

WoSign was aware of some customers couldn't use the 80 or 443 ports for performing the website control validation and requested a change to use any port for this validation. This change was made on January 10<sup>th</sup> 2015.

### 5.3. What has been done

WoSign changed their system to fix the problem and closed all ports except 80 and 443. So the high port validation allowed period is from Jan. 10<sup>th</sup>, 2015 to April 24<sup>th</sup>, 2015.

WoSign posted all those certificates to WoSign CT log server at Aug. 26<sup>th</sup> 2016 and Google CT log server at Sept. 03<sup>rd</sup> 2016

### 5.4. Current situation

WoSign fixed the bug and disabled the website control validation for ports different of 80 or 443. Also have investigated every certificate and decided to not revoke these certificates. The certificates were not violating the BRs.

On the other hand, it's WoSign fault not having notified the WebTrust auditor of this issue and hence not communicated to the browsers.

## 6. Issue N: Additional Domain Errors (June 2015)

---

*(a.k.a. "Issue 1")*

In June 2015, an applicant found some problems with WoSign's free certificate service. There were actually two bugs, which we will denote N1 and N2.

Bug N1 was an issue where someone proving control of <subdomain>.example.tld also was given a cert covering example.tld.

Bug N2 was an issue where arbitrary domains can be added to an existing request after validation.

### WoSign Response

#### 6.1 What happened

##### 6.1.1 Bug N1

This is a system bug come from website control validation that when a subscriber passes the subdomain validation, then our system added the top domain in the certificate automatically. We searched our database and there were 21 mis-issued certificates of this type, all certificates were revoked and posted to CT log servers.

##### 6.1.2 Bug N2

This is another system bug that when the subscriber finished the domain control validation, he/she can use a special professional method to add other un-validated domain to the order, then our system issued the certificate including all domains in the order. We searched our database and found 12 mis-issued certificates with this bug including the certificate issued to the domain "github.com, all certificates were revoked and posted to CT log servers.

#### 6.2 Why this happened

##### 6.2.1 Bug N1

This mis-issued case was caused by the engineer misunderstood the adding of an additional domain rule. The rule is; if you validate the domain: wosign.com, and you apply for a certificate for wosign.com, then the system will add a subdomain www.wosign.com in the SAN for free, this is for the subscriber convenience and there's no problem if the site visitor visits https://wosign.com and https://www.wosign.com.

This is not a problem in Domain Whois Control Validation\*, but for website control validation method, it would be a problem if the subscriber validated a subdomain that added the top domain to the certificate. This bug is fixed completely at Aug. 10th, 2015 system update since we changed the order procedure that subscriber submit all the domains first to database, then validate it one by one, so the vulnerability was fixed.

\* “Domain Whois Control Validation” means “BR - 3.2.2.4.4 Constructed Email to Domain Contact” that system send verification code to domain name whois admin email, subscriber must input this code in the application process.

### 6.2.2 Bug N2

These mis-issued certificates were a system bug that when the subscriber finished the domain validation, they can add any other domain before submitting this order to system., the vulnerability got fixed on the August 10<sup>th</sup> 2015 system upgrade, this upgrade changed the order procedure that subscriber submit the all domains first to database, then validate it one by one, the vulnerability was fixed.

The reason that we found the github issue but did not found others is we have a protected domain list that github is in the list, other mis-issued certificate is not recognized as a famous brand that not in the list and was issued automatically.

The following screenshot is the current keyword setting for github, “f”=flag; “r”=reject, we changed the class 1 certificate from “f” to “r” after we found out the mis-issued certificate case for github.

Keyword	Level	Enable	Sub Domain	Top Domain	Root Domain
github	Class3	✔	f	f	r
github	Class4	✔	f	f	r
github	Class2	✔	r	r	r
github	Class1	✔	r	r	r

### 6.3 What has been done

The two bugs were caused by the unreliable order procedure that our system needed to verify every parameter in the server side. So we changed the order procedure that all orders info, including the domain list, write into database first, then the subscriber need to validate the domain name one by one, with no chance for the subscriber to modify the order data.

And the problem with the website control validation, we think it is not a secure method for validation, so finally decided to disable this method to prevent this case happen in the future.

### 6.4 Current situation

WoSign fixed the bug and disabled the website control validation since Aug. 27<sup>th</sup> 2015 even the BR allows this method

## 7. Issue P: Use of SM2 Algorithm (Nov 2015)

In November 2015, WoSign issued two certificates that have subject public keys which are for the [SM2 algorithm](#). SM2 is an elliptic-curve-based algorithm but it does not use the US NIST P-256, P-384, or P-521 curves. The CA/Browser Forum Baseline Requirements section 6.1.5 requires that only these three curves be used for elliptic curve keys in certs covered by the BRs.

In addition to including subjects keys using unapproved parameters, it seems these each share their serial number with another certificate for the same subject.

- [1st SM2 cert in crt.sh](#); [cert with same serial number in crt.sh](#)
- [2nd SM2 cert in crt.sh](#); [cert with same serial number in crt.sh](#)

Secondly, for the first pair of certs, the validity period is 4 years, which is 9 months longer than allowed by the BRs.

## WoSign Response

### 7.1 What happened

WoSign issued two SM2 algorithm SSL certificates for testing in 2015, and also issued 4 SM2 algorithm certificates in 2016 for testing again.

Here is the 2016 issued SM2 certificate in crt.sh:

2016-01-13 sm2 signature: <https://crt.sh/?id=31753567>

2016-01-13 sm2 encryption: <https://crt.sh/?id=31753575>

2016-01-25 sm2 signature: <https://crt.sh/?id=31753571>

2016-01-25 m2 encryption: <https://crt.sh/?id=31753573>

### 7.2 Why this happened

For year 2015 issue, these two certificates were issued manually in the test lab since the CMS and PKI system can't issue this SM2 algorithm certificate, and doing manually, a human mistake issued these certificates exceeding the 39-month limit.

We used the same serial number as the RSA certificate (same subject) to test if we can setup a server side gateway that install this two type certificates, it can make the handshake automatically using different certificate based on the browser algorithm support.

The reason why these certificates were issued from a trusted root was for testing the real scenario in the Internet using standard browser without SM2 support and browser with SM2 support, and for the effect and probing if Windows can display the certificate path correctly.

For year 2016 issue, this is a small change of the previous one but this time not using the same serial number with the RSA certificate, these certificates were issued in the test lab manually again because the CMS and PKI system can't issue this SM2 algorithm certificate. These 4 SM2 certificates were for testing the effect for different key usage in browser side and in server side.

### 7.3 Current situation

The test is finished and there's no need to test any more.

To avoid future testing incidents, WoSign updated the internal test systems to avoid issuing test certificates from public trusted root that violate the BR.

## 8. Issue R: Purchase of StartCom (Nov 2015)

---

WoSign purchased the CA "StartCom" and did not disclose the transaction as a change of ownership, which we believe violates section 5 of the [Mozilla CA Certificate Maintenance Policy](#). Furthermore, when this clause was brought to their attention, they denied that any changes fell under it, and they attempted to suppress further information about the ownership transfer when it was brought to the community's attention.

Full details can be found in [the post in mozilla.dev.security.policy](#).

### WoSign Response

An announcement and disclosure will be made shortly pending completion of the business transaction.

## 9. Issue S: Backdated SHA-1 Certs (January 2016)

WoSign has issued certificates after January 1st 2016 but backdated the notBefore date to be in December 2015. This has the effect of avoiding the blocks in browsers regarding SHA-1 certs issued after January 1st 2016. The number of certs affected is probably 67, but may be a few more or less.

### WoSign Response

#### 9.1. What happened

We researched our system and found only **8** SHA-1 SSL certificates that were mis-issued after January 1<sup>st</sup> 2016 until June 28<sup>th</sup> 2016.

#### 9.2. Why this happened

There are two type cases for those 8 certificates that were backdated:

- (1) System bug: 6 certificates
- (2) API bug: 2 certificates

At Dec. 30<sup>th</sup> 2015 17:32, Richard Wang sent email to the related team about SHA-1 deadline, it says "System can't issue SHA-1 SSL certificate after Dec. 30<sup>th</sup> 2015 24:00, **NO EXCEPTION!**"

So <https://buy.wosign.com> website closed the SHA-1 option for subscribers from Dec. 30<sup>th</sup> 2015.

From: Richard Wang

RE: SHA1使用期限

30 December 2015 at 17:32

过了12点就不能签发了，没有商量的

But we have many history orders in the system even some orders are placed at September 2015, and we checked our daily SSL certificate issued from Dec. 18<sup>th</sup> to Dec. 31<sup>st</sup> that no any special amount for Dec. 20<sup>th</sup> increase, the second Saturday and Sunday also have many certificate issued, and there are no any special amount SHA-1 certificate at some day.

There are suspicious **64** SHA-1 certificates listed in [Github](#), we checked it one by one carefully, and asked our customer service team to check and recall each order.

This one-by-one checking took almost one-week time, here is the investigation result:

- 1) The following **6** SHA-1 EV SSL certificates are mis-issued, those orders were placed before Jan 1<sup>st</sup> 2016 and were pre-signed, but the process stopped due to some reason like payment problem, proof document problem etc. and after those orders were ready to be issued, the system automatically changed to SHA-2 signature to resign this certificate since it is after Jan. 1<sup>st</sup> 2016 that can't issue SHA-1 certificate, and post to CT log server to get SCT data.

The system should have disabled the pre-signed certificate with the SHA-1 algorithm to CT, but it had a **bug** that post two related certificates all to log server. This bug caused the following 6 EV SSL certificates were mis-issued, all are two certificates with SHA1 and SHA2 certificate with same subject info.

We found this bug at Jan. 18<sup>th</sup> 2016 and fixed it. This scenario can't happen now. Those 6 SHA1 certificate are revoked.

Here are the 6 SHA-1 certificates in crt.sh:

<https://crt.sh/?serial=6D24E483E27F55479C5C555B37745353>  
<https://crt.sh/?serial=179A6D058F50116D62E422F49ABB8686>  
<https://crt.sh/?serial=5ACF9A707E8E32D0A36F947ACD6C8981>  
<https://crt.sh/?serial=15AE547B1136CA1074EEBADE368F9054>  
<https://crt.sh/?serial=5DF26F6A29304CE8C559DBFFABBB37D1>  
<https://crt.sh/?serial=5A47B7074267A7D44441618D84686547>

- 2) The following **9** certificates is normal SHA-1 certificate without any doubt, we issued SHA1 certificate at every day till Dec. 31<sup>st</sup> 2015, not just Dec. 30<sup>th</sup> 2015.  
<https://crt.sh/?serial=383D5C00F511AD0BEE3A83DCA382FC8D>  
<https://crt.sh/?serial=4153D33AB18525012B5D461778E32327>  
<https://crt.sh/?serial=5EE14DC27F910CDB2BCDF39A8635AD11>  
<https://crt.sh/?serial=53ED5BC73D09C2A838482230EE552D4F>  
<https://crt.sh/?serial=3BE9494816A5C5F3B138B524CEFEB9F2>

https://crt.sh/?serial=32B37DE8C629127E76434054531A6347  
https://crt.sh/?serial=3423292E4FCFFA22F982D470CC27E5EE  
https://crt.sh/?serial=100FD9B985D145EA7FA14C57A59FFC5F  
https://crt.sh/?serial=47F9762177383469123846D22B1929A6

- 3) The following **2** certificates were issued from an API bug that was found by Computest, just two test certificates that we revoked it after we got the report.  
https://crt.sh/?serial=6565E1710A48FBBE1E2B61835C789C39      issued at 2016-06-23  
https://crt.sh/?serial=6745ED57FE25880FB7D93A774310CF59      issued at 2016-06-28
- 4) The rest **47** SHA-1 certificate are normal SHA-1 orders; the suspicion reason is that there were two same domain order before Jan 1<sup>st</sup> 2016 and after Jan 1<sup>st</sup> 2016. The reason is that the subscriber ordered a SHA-1 certificate, but due to the no-allow SHA-1 certificate issuance, some customer ordered another SHA-2 certificate one month later after Jan 1<sup>st</sup> 2016, some are two months and even 4 months later.

Another case is customer bought SHA-1 OV SSL certificate before Jan 1<sup>st</sup> 2016, but wanted to upgrade to EV SSL certificate after Jan. 1<sup>st</sup> 2016, we resign its CSR to SHA-2 EV SSL certificate that it can be imaged as signing it in the same time after Jan 1<sup>st</sup> 2016.

### 9.3. What has been done

For bug (1), we fixed it at Jan. 18<sup>th</sup> 2016, and we added more certificate parameter check before posting to CT log server. For bug (2), we deleted the API bug code, see issue #11.

We closed the SHA-1 signing in the whole system at July 2<sup>nd</sup> 2016 after the issue #11 happened. To make transparency of this kind of case, WoSign decided to log all issued SSL certificate to Google CT log server at **July 4<sup>th</sup> 2015** that released a news:

[https://www.wosign.com/english/News/2016\\_wosign\\_CT.htm](https://www.wosign.com/english/News/2016_wosign_CT.htm), promised to all browsers that if the certificate issued after July 5<sup>th</sup> 2016 without SCT data embedded in the certificate, browsers can distrust this certificate.

### 9.4 Current situation

WoSign fixed the bug and disabled the SHA-1 signing for SSL certificate, no more issued since July 2<sup>nd</sup> 2016.

## 10. Issue T: alicdn.com Misissuance (June 2016)

---

A certificate has been issued in June 2016 to alicdn.com which, it is claimed, was not requested by the owner of that domain. However, it has not yet been possible to confirm that this cert has been mis-issued because the owner of the private key has not been located. The domains in question currently use certificates from Symantec.

- [Cert on Github Gist](#)
- [Cert on crt.sh](#)

## WoSign Response

### 10.1 What happened

This certificate is reported at the Mozilla mail list at August 26, 2016 1:13 PM that it claimed it is misissued certificate for “alicdn.com”

-----Original Message-----

From: dev-security-policy [mailto:dev-security-policy-bounces+...@wosign.com@lists.mozilla.org] On Behalf Of 233sec Team  
Sent: Friday, August 26, 2016 1:13 PM  
To: mozilla-dev-security-policy@lists.mozilla.org  
Subject: Re: Incidents involving the CA WoSign

Wosign's Issue mechanism is high risking for large enterprise.  
This is one prove:

<https://gist.github.com/xiaohuilam/8589f2dfaac435bae4bf8dfe0984f69e>

Alicdn.com is the cdn asset domain name of Taobao/tmall who belong to alibaba, which are Chinese biggest online shopping websites.  
With the fake cert's middle man attack, password stealing, information leaking...

---

dev-security-policy mailing list  
dev-security-policy@lists.mozilla.org  
<https://lists.mozilla.org/listinfo/dev-security-policy>

### 10.2 Why this happened

We checked our system, there were two orders related to domain “alicdn.com”, both orders passed the website control validation, since it is free DV SSL certificate that issued after it is a well-validated, no more manual check took.

The two certificates were post to CT log server, here is the crt.sh link:

<https://crt.sh/?id=31104164>

<https://crt.sh/?id=29884704>

#### 10.2.1 Website control validation log

(1) Certificate: <https://crt.sh/?id=31104164>

**2016-06-23 01:34:39**, validation system received domain "alicdn.com" website control validation request, the URL is <http://alicdn.com/alicdn.com.html>, the domain random ID is 2e3baabe989fad9f143517796ed4941c13e7177b.

Validation system used GET/[alicdn.com.html](http://alicdn.com.html) HTTP/1.1 to host: <http://alicdn.com:80/alicdn.com.html>, the server returns “HTTP/1.1 400 Bad Request”. Then the validation system used POST to <http://alicdn.com:80/alicdn.com.html>, the sever returns “HTTP/1.1 200 OK”, then system get the correct verification code that passed the website control validation, then issued the certificate.

Here is the screen shot from validation system log:



```

2016-06-23 01:34:39 v_random=2e3baabe989fad9f143517796ed4941c13e7177b
2016-06-23 01:34:39 v_domain=alicdn.com
2016-06-23 01:34:39 verify file url=http://alicdn.com/alicdn.com.html
2016-06-23 01:34:39 Open connection to alicdn.com:80
2016-06-23 01:34:39 "GET /alicdn.com.html HTTP/1.1[\r][\n]"
2016-06-23 01:34:39 Adding Host request header
2016-06-23 01:34:39 "Content-Type: text/html;charset=utf-8[\r][\n]"
2016-06-23 01:34:39 "User-Agent: Jakarta Commons-HttpClient/3.1-rc1[\r][\n]"
2016-06-23 01:34:39 >> "Host: alicdn.com[\r][\n]"
2016-06-23 01:34:39 << "HTTP/1.1 400 Bad Request[\r][\n]"
2016-06-23 01:34:39 << "Server: Tengine[\r][\n]"
2016-06-23 01:34:39 << "Content-Type: application/xml[\r][\n]"
2016-06-23 01:34:39 << "Content-Length: 268[\r][\n]"
2016-06-23 01:34:39 << "Connection: keep-alive[\r][\n]"
2016-06-23 01:34:39 << "Date: Wed, 22 Jun 2016 17:34:37 GMT[\r][\n]"
2016-06-23 01:34:39 << "x-oss-request-id: 576ACC2DCDF474A7DFFA1A9F[\r][\n]"
2016-06-23 01:34:39 << "Via: cache18.l2et15-1[9,400-0,M], cache26.l2et15-1[9,0], cache2.cn297[66,400-0,M], cache4.cn297[67,0][\r][\n]"
2016-06-23 01:34:39 << "X-Cache: MISS TCP_MISS dirn:-2:-2[\r][\n]"
2016-06-23 01:34:39 << "X-Swift-SaveTime: Wed, 22 Jun 2016 17:34:37 GMT[\r][\n]"
2016-06-23 01:34:39 << "X-Swift-CacheTime: 1[\r][\n]"
2016-06-23 01:34:39 << "Timing-Allow-Origin: *[\r][\n]"
2016-06-23 01:34:39 << "EagleId: 8ccd4dcc14666168778245767e[\r][\n]"
2016-06-23 01:34:39 get failure!
2016-06-23 01:34:39 << "<?xml version="1.0" encoding="UTF-8"?>[\n]"
2016-06-23 01:34:39 << "<Error>[\n]"
2016-06-23 01:34:39 << " <Code>InvalidBucketName</Code>[\n]"
2016-06-23 01:34:39 << " <Message>The specified bucket is not valid.</Message>[\n]"
2016-06-23 01:34:39 << " <RequestId>576ACC2DCDF474A7DFFA1A9F</RequestId>[\n]"
2016-06-23 01:34:39 << " <HostId>alicdn.com</HostId>[\n]"
2016-06-23 01:34:39 << " <BucketName>alicdn.com.html</BucketName>[\n]"
2016-06-23 01:34:39 << "</Error>[\n]"
2016-06-23 01:34:39 Releasing connection back to connection manager.
2016-06-23 01:34:39 Open connection to alicdn.com:80
2016-06-23 01:34:39 >> "POST /alicdn.com.html HTTP/1.1[\r][\n]"
2016-06-23 01:34:39 Adding Host request header
2016-06-23 01:34:39 >> "User-Agent: Jakarta Commons-HttpClient/3.1-rc1[\r][\n]"
2016-06-23 01:34:39 >> "Host: alicdn.com[\r][\n]"
2016-06-23 01:34:39 >> "Content-Length: 0[\r][\n]"
2016-06-23 01:34:39 Request body has not been specified
2016-06-23 01:34:39 << "HTTP/1.1 200 OK[\r][\n]"
2016-06-23 01:34:39 << "Server: Tengine[\r][\n]"
2016-06-23 01:34:39 << "Content-Type: text/html[\r][\n]"
2016-06-23 01:34:39 << "Content-Length: 60[\r][\n]"
2016-06-23 01:34:39 << "Connection: keep-alive[\r][\n]"
2016-06-23 01:34:39 << "Date: Wed, 22 Jun 2016 17:34:38 GMT[\r][\n]"
2016-06-23 01:34:39 << "Via: cache61.l2nu16-1[159,200-0,M], cache52.l2nu16-1[160,0], cache2.cn297[248,200-0,M], cache8.cn297[250,0][\r][\n]"
2016-06-23 01:34:39 << "X-Cache: MISS TCP_MISS dirn:-2:-2[\r][\n]"
2016-06-23 01:34:39 << "X-Swift-SaveTime: Wed, 22 Jun 2016 17:34:38 GMT[\r][\n]"
2016-06-23 01:34:39 << "X-Swift-CacheTime: 0[\r][\n]"
2016-06-23 01:34:39 << "Timing-Allow-Origin: *[\r][\n]"
2016-06-23 01:34:39 << "EagleId: 8ccd4dd014666168779512340e[\r][\n]"
2016-06-23 01:34:39 Buffering response body
2016-06-23 01:34:39 << "bE0vdXpWdWQyQWJoT012RGx2bG5lVUtBVG9JUK1ZamxzbXlycUQxVmJDTT0[\n]"
2016-06-23 01:34:39 Releasing connection back to connection manager.
2016-06-23 01:34:39 Default charset used: ISO-8859-1
2016-06-23 01:34:39 result=bE0vdXpWdWQyQWJoT012RGx2bG5lVUtBVG9JUK1ZamxzbXlycUQxVmJDTT0
2016-06-23 01:34:39 fileContent=bE0vdXpWdWQyQWJoT012RGx2bG5lVUtBVG9JUK1ZamxzbXlycUQxVmJDTT0
2016-06-23 01:34:40 result=0||success
2016-06-23 01:34:40 Cost time=838ms

```

(2) Certificate: <https://crt.sh/?id=29884704>

**2016-06-23 09:17:01**, validation system received domain "alicdn.com" website control validation request, the URL is "http://alicdn.com/alicdn.com.html", domain random ID is bf5d1e3cc3f29b599c20d2280431d70b7ddc1a58.

Validation system used GET to <http://alicdn.com:80/alicdn.com.html>, the server returns “HTTP/1.1 400 Bad Request”. Then the validation system used POST to <http://alicdn.com:80/alicdn.com.html>, the sever returns “HTTP/1.1 400 Bad Request”. Then system used GET <https://alicdn.com:443/alicdn.com.html>, the server returns “HTTP/1.1 200 OK”, then system gets the correct verification code that passed the website control validation, and then issued this certificate.

Here is the screen shot from validation system log:

```
2016-06-23 09:17:01 v_random=bf5d1e3cc3f29b599c20d2280431d70b7ddc1a58
2016-06-23 09:17:01 v_domain=alicdn.com
2016-06-23 09:17:01 punycodeDomain=alicdn.com
2016-06-23 09:17:01 v_name:https://alicdn.com/alicdn.com.html
2016-06-23 09:17:01 verify file url=https://alicdn.com/alicdn.com.html
2016-06-23 09:17:02 get verify file succeed,strRep=RjVHeVpsdnYxdlo1U1UwWWZrV1JlN204ZkFJT1RlY3RQVTJ5U0w00EpYWT0
2016-06-23 09:17:02 fileContent=RjVHeVpsdnYxdlo1U1UwWWZrV1JlN204ZkFJT1RlY3RQVTJ5U0w00EpYWT0
2016-06-23 09:17:02 Call cmsWebVerification, Cost time=627ms
```

In this website control validation, system doesn't verify the domain in the certificate. Some website disabled port 80 for security, so we always try 443 if 80 fails.

### 10.2.2 DNS resolution record

Here is the dig record in the validation server:

```
[root@localhost ~]# dig alicdn.com +trace
; <<>> DiG 9.7.3-P3-RedHat-9.7.3-8.P3.el6 <<>> alicdn.com +trace
;; global options: +cmd
.           1225 IN    NS     j.root-servers.net.
.           1225 IN    NS     g.root-servers.net.
.           1225 IN    NS     i.root-servers.net.
.           1225 IN    NS     e.root-servers.net.
.           1225 IN    NS     l.root-servers.net.
.           1225 IN    NS     f.root-servers.net.
.           1225 IN    NS     h.root-servers.net.
.           1225 IN    NS     m.root-servers.net.
.           1225 IN    NS     d.root-servers.net.
.           1225 IN    NS     k.root-servers.net.
.           1225 IN    NS     c.root-servers.net.
.           1225 IN    NS     b.root-servers.net.
```

```

.           1225 IN   NS    a.root-servers.net.
;; Received 228 bytes from 101.226.4.6#53(101.226.4.6) in 52 ms
com.       172800 IN   NS    a.gtld-servers.net.
com.       172800 IN   NS    b.gtld-servers.net.
com.       172800 IN   NS    c.gtld-servers.net.
com.       172800 IN   NS    d.gtld-servers.net.
com.       172800 IN   NS    e.gtld-servers.net.
com.       172800 IN   NS    f.gtld-servers.net.
com.       172800 IN   NS    g.gtld-servers.net.
com.       172800 IN   NS    h.gtld-servers.net.
com.       172800 IN   NS    i.gtld-servers.net.
com.       172800 IN   NS    j.gtld-servers.net.
com.       172800 IN   NS    k.gtld-servers.net.
com.       172800 IN   NS    l.gtld-servers.net.
com.       172800 IN   NS    m.gtld-servers.net.
;; Received 488 bytes from 193.0.14.129#53(k.root-servers.net) in 168 ms
alicdn.com. 172800 IN   NS    nsp.alibabaonline.com.
alicdn.com. 172800 IN   NS    ns8.alibabaonline.com.
alicdn.com. 172800 IN   NS    nshz.alibabaonline.com.
alicdn.com. 172800 IN   NS    nsp2.alibabaonline.com.
;; Received 244 bytes from 192.43.172.30#53(i.gtld-servers.net) in 10158 ms
alicdn.com. 300  IN   A    140.205.77.240
alicdn.com. 300  IN   A    115.238.23.240
alicdn.com. 172800 IN   NS    nsp2.alibabaonline.com.
alicdn.com. 172800 IN   NS    nsp.alibabaonline.com.
alicdn.com. 172800 IN   NS    ns8.alibabaonline.com.
alicdn.com. 172800 IN   NS    nshz.alibabaonline.com.
;; Received 276 bytes from 140.205.2.184#53(nsp2.alibabaonline.com) in 34 ms

```

Here is the nslookup using Google DNS:

```
C:\Users\Richard>nslookup - 8.8.8.8
默认服务器: google-public-dns-a.google.com
Address: 8.8.8.8

> alicdn.com
服务器: google-public-dns-a.google.com
Address: 8.8.8.8

非权威应答:
名称: alicdn.com
Addresses: 115.238.23.240
          140.205.77.240
```

### 10.2.3 Other related information

We noticed Aliyun (Alibaba Cloud) team to check this problem, they confirmed this order is not from Aliyun, and they checked this case that confirmed this website control validation is done successfully by short time traffic hijack, but we don't have more details for this hijack. We gave the Alibaba Cloud related person email to Mozilla to contact Alibaba Cloud directly.

And the incident reporter also confirmed this is not a validation problem, he thinks this is a problem that we must do more human validation.

-----Original Message-----

From: dev-security-policy [mailto:dev-security-policy-bounces+richard=wosign.com@lists.mozilla.org] On Behalf Of 233sec Team  
Sent: Saturday, August 27, 2016 8:34 AM  
To: mozilla-dev-security-policy@lists.mozilla.org  
Subject: Re: Incidents involving the CA WoSign

Not vulnerabilities mentioned in this thread, but a Human-Audit weak process.  
Detail you can see the reply content i send to Mr.Wang

Here is his email to Mr. Wang:

**From:** 蓝小灰 [mailto:inccccft@gmail.com]  
**Sent:** Saturday, August 27, 2016 8:38 AM  
**To:** Richard Wang [mailto:richard.wang@wosign.com]  
**Subject:** Re: Incidents involving the CA WoSign

Yes, issued from wosign order system.  
But this a high risk domain which can brings big websites' Middle-man Problem,

With important domains(alexa top 5000),  
Don't your company do anything manually to review the certificate?  
Don't do some phone callback mention?

### 10.3 What has been done

After we got report, we revoked this certificate and add keyword "alicdn" "aliyun" to our Flag-Reject system (alibaba is in the system), it will be rejected for those 3 domain for class 1 and Class 2 SSL certificate in the future.

Keyword	Level	Create Date	Enable	Sub Domain	Top Domain	Root Domain
alicdn	Class2	2016-08-29 09:53:13	✔	r	r	r
alicdn	Class3	2016-08-29 09:53:55	✔	f	f	f
alicdn	Class4	2016-08-29 09:54:16	✔	f	f	f
alicdn	Class1	2016-08-29 09:53:00	✔	r	r	r

Keyword	Level	Create Date	Enable	Sub Domain	Top Domain	Root Domain
aliyun	Class1	2016-08-29 09:55:22	✔	r	r	r
aliyun	Class2	2016-08-29 09:55:48	✔	r	r	r
aliyun	Class3	2016-08-29 09:56:06	✔	f	f	f
aliyun	Class4	2016-08-29 09:56:25	✔	f	f	f

Keyword	Level	Create Date	Enable	Sub Domain	Top Domain	Root Domain
alibaba	Class4	2016-02-01 11:52:15	✔	r	f	r
alibaba	Class3	2016-02-01 11:52:07	✔	r	f	r
alibaba	Class2	2016-02-01 11:52:00	✔	r	r	r
alibaba	Class1	2016-02-01 11:51:48	✔	r	r	r

Considering the website control validation method has potential risk, we have closed this method at **Aug. 27<sup>th</sup> 2016** even the BR allow this method. There are many famous Internet service providers provide subdomain to its customer, we can't add all of their domains to our Flag-Reject system. So we decided to close this validation method, only support domain control validation.

WoSign doesn't think this case is a misissuance mistake since it passed the website control validation. After we got report, we revoked the two related certificates, and added Alibaba related domain to Flag-Rejection system to prevent it will not happen in the future. And we even closed the website control validation method to all subscribers to prevent other Internet Service Provider's domain certificate is mis-issued.

## 11. Issue V: StartEncrypt (July 2016)

---

*(a.k.a. "Issue 2")*

In July 2016, it became clear that there were some problems with the StartEncrypt automatic issuance service recently deployed by the CA StartCom. This was a StartCom-branded service and was not publicized as being able to issue certificates from WoSign. However, changing a simple API parameter in the POST request on the submission page changed the intermediate/root certificate to which the resulting certificate chained up.

## WoSign Response

### 11.1 What happened

Computest reported this bug in June 30<sup>th</sup> 2016 that using StartEncrypt API can issue SHA-1 certificates from WoSign intermediate CA backdating the certificates to Dec. 20<sup>th</sup> 2015.

### 11.2 Why this happened

This is not the case that we want to issue backdated SHA1 certificate intentionally, this is a bug that used by the test company to issued two SHA-1 certificates only. StartCom and WoSign used the same auto-generation script, set different parameter to go to different CA API URL. The API is designed at 2015 that allowed to issue SHA-1 certificate, and this code is used for StartEncrypt that didn't disable this SHA-1 function. The test company posted SHA-1 parameter request to WoSign API server that issued a backdated certificate, it is an API bug that classify it to Issue #9.

### 11.3 What has been done

WoSign deleted this bug code in API instantly, and closed the API service and deleted the API domain name resolution and stopped to use StartEncrypt service.

Stopping this API service is the quickest way to prevent this case in the future.

**Finally**, thanks for everyone's contribution in the Mozilla-Dev-Security-Policy mail list to help us find our problem that let us know our problem and to improve our system to be more secure and more reliable in the future.

**WoSign** remains committed to continually evolve our technology, processes, and offerings to help keep our customers and the Internet safe. We believe that the steps we have taken will ensure that this type of incident never happens again, and we believe that full support for CT is our commitment of supervision.

**Thanks.**

**WoSign CA Limited**