keyon

Faking Extended Validation SSL Certificates in Internet Explorer 7

June 7th 2007, V1.1

Martin Christinat, CTO, christinat@keyon.ch

Abstract

Extended Validation (EV) SSL certificates are a new offering by trusted third parties like VeriSign and are only issued to companies after a standardized vetting process. When an EV enabled browser like Internet Explorer 7 connects to a server that presents an EV certificate, the address bar is highlighted green and a certificate status bar containing the company name and the identifying company is shown. This visual feedback should give the user the assurance to be connected to a valid secure site of the named company.

While the process to get an EV certificate ensures that only valid companies will get an EV certificate, the technical implementation of EV support in IE7 allows enabling any root certificate and even self created root certificates to issue EV certificates. The EV status (the green bar) shown in IE7 does therefore currently not guarantee that the standardized vetting process before issuing an EV certificate was actually followed.

Since the import and EV enabling of a fake root certificate turned out to be quite easy and is possible without user intervention with both Windows XP and Windows Vista, new advanced Phishing attacks based on the implied trust by the EV visual feedback become possible.

Keyon created a small application which demonstrates the feasibility and simplicity of this kind of attack. A real web attack would combine the import with the exploit of a remote code execution vulnerability.

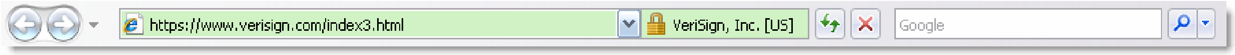
About Keyon

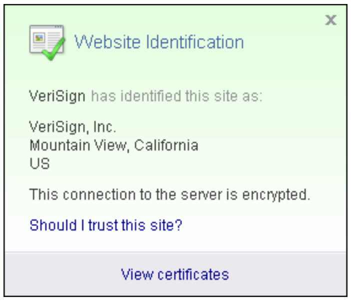
Keyon is a leading provider of solutions and services in the areas of e-business, IT security, software engineering and legally valid processing and long-term storage of data and invoices using electronic signatures. We have first-class references in the area of Finance, Insurance, Retail, Industry, Health, Telecommunication and Public Administration.

Extended Validation certificates (EV SSL certificates)

Extended Validation SSL Certificates are standard X.509 certificates which require more extensive investigation by the Certificate Authority before being issued [3]. The requirements for issuing EV certificates are defined by the CA/Browser forum [1].

To provide a visual feedback to the user when an EV SSL certificate is presented to the browser by a web site, an EV enabled browser like IE7 will highlight the address bar in green and show additional information about the certificate owner next to the address bar:





VeriSign has identified this site as:

VeriSign, Inc.
Mountain View, California
US

This connection to the server is encrypted.

Quotes from the CA/Browser forum:

The introduction of EV SSL Certificates will tighten the security of Internet transactions as certificate requestors will be subject to a thorough, standardized vetting process which all issuing CAs must adhere to.

The combination of EV SSL certificates and new browser versions thus will help Internet users ascertain that the Web sites they are visiting indeed are the ones they expect to access; not fraudulent ones masquerading as popular sites. This will make it considerably more difficult for perpetrators of phishing schemes to successfully impersonate high-traffic Web sites.

While the CA/Browser forum defines the process and the technical requirements regarding certificate content and revocation checking, the actual implementation to detect an EV certificate and act accordingly is up to the browser vendor.

For the rest of this paper we will focus on the technical implementation of EV SSL certificate support in Internet Explorer 7, the first browser supporting EV natively.

Our aim was to determine if it is possible to make our own EV certificates and use them with IE7. The first question thus was: How does IE7 know when to paint a green bar? Unfortunately there is no in-depth technical documentation available on this subject, only some generic documentation [1].

The magic of EV Updater

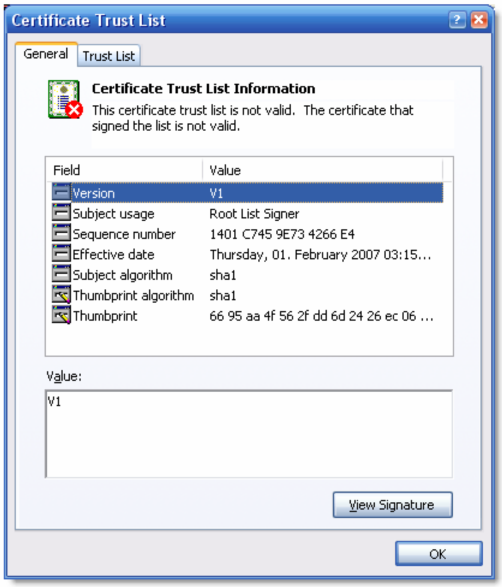
In order for IE7 to recognize EV SSL certificates a root certificate update is necessary. This is accomplished transparently for the user by using a mechanism commonly named or referred to by public certificate authorities as EV updater. This EV updater trick will simply trigger an automatic root certificate update by IE7. The information which root certificate is allowed to issue EV certificates must thus be part of the information downloaded during this update.

When the automatic root certificate update (ARCU) is triggered, the CryptoAPI downloads the following files from www.download.windowsupdate.com:

/msdownload/update/v3/static/trustedr/en/authrootseq.txt
/msdownload/update/v3/static/trustedr/en/authrootstl.cab

The first file (authrootseq.txt) contains only a string like 1401C7459E734266E4 and is used to determine if the root list on the download server was updated since the last check by the client.

The second file (authrootstl.cab) is only downloaded if the client determines that a new list is available. It's a CAB file containing only one file, a certificate trust list (CTL) named authroot.stl. If the CTL is opened, we find that it is signed by the Microsoft Certificate Trust List Publisher:



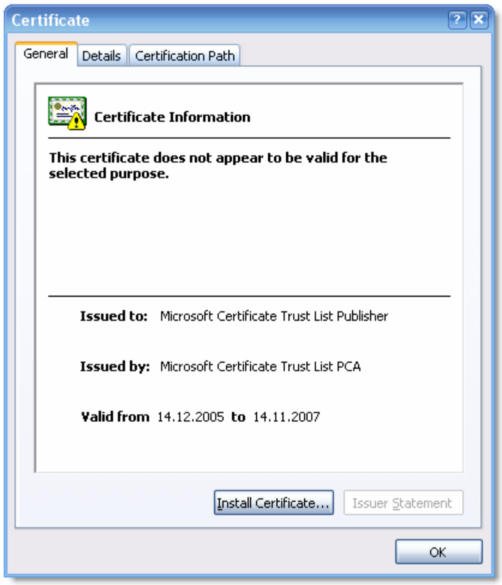
Certificate Trust List Information

This certificate trust list is not valid. The certificate that signed the list is not valid.

Field	Value
Version	V1
Subject usage	Root List Signer
Sequence number	1401 C745 9E73 4266 E4
Effective date	Thursday, 01. February 2007 03:15...
Subject algorithm	sha1
Thumbprint algorithm	sha1
Thumbprint	66 95 aa 4f 56 2f dd 6d 24 26 ec 06 ...

Value:

V1



Certificate Information

This certificate does not appear to be valid for the selected purpose.

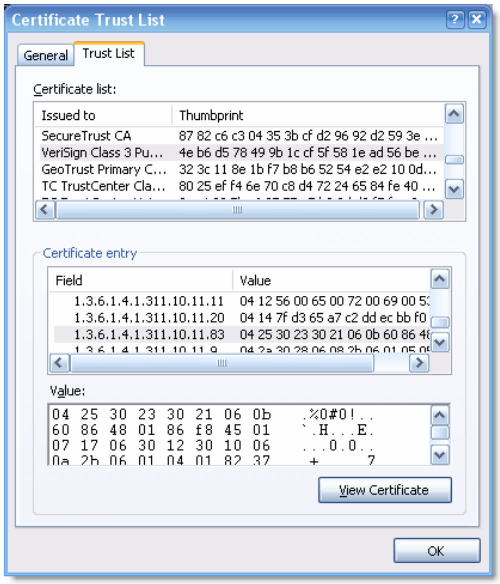
Issued to: Microsoft Certificate Trust List Publisher

Issued by: Microsoft Certificate Trust List PCA

Valid from 14.12.2005 to 14.11.2007

Interestingly, the certificate used to sign the file is shown as not valid, although the enhanced key usage lists it as Root List Signer (1.3.6.1.4.1.311.10.3.9). However the system can obviously import the contents of the CTL silently into the root certificate store.

To shed some light on the EV enabling mechanism, we look at the certificate entry of a root certificate that is allowed to issue EV certificates. For our investigation, we choose the *VeriSign Class 3 Public Primary Certification Authority - G5* certificate from the CTL:



Issued to	Thumbprint
SecureTrust CA	87 82 c6 c3 04 35 3b cf d2 96 92 d2 59 3e …
VeriSign Class 3 Pu…	4e b6 d5 78 49 9b 1c cf 5f 58 1e ad 56 be …
GeoTrust Primary C…	32 3c 11 8e 1b f7 b8 b6 52 54 e2 e2 10 0d…
TC TrustCenter Cla…	80 25 ef f4 6e 70 c8 d4 72 24 65 84 fe 40 …

Field	Value
1.3.6.1.4.1.311.10.11.11	04 12 56 00 65 00 72 00 69 00 53
1.3.6.1.4.1.311.10.11.20	04 14 7f d3 65 a7 c2 dd ec bb f0
1.3.6.1.4.1.311.10.11.83	04 25 30 23 30 21 06 0b 60 86 48
1.3.6.1.4.1.311.10.11.9	04 2a 30 28 06 08 2b 06 01 05 05

```
04 25 30 23 30 21 06 0b    .%0#0!..
60 86 48 01 86 f8 45 01    `.H...E.
07 17 06 30 12 30 10 06    ...0.0..
0a 2b 06 01 04 01 82 37     +     7
```

The interesting part is found in the certificate entry section under *Additional Attributes*. There are a few OIDs shown, all starting with 1.3.6.1.4.1.311.10.11. One of these attributes, 1.3.6.1.4.1.311.10.11.11, is defined in the WinCrypt.h header file of the Platform SDK as CERT_FRIENDLY_NAME_PROP_ID and contains the friendly name of the CA which is shown e.g. in the IE7 certificate content dialog.

A more interesting additional attribute which can only be found for EV enabled root certificates is 1.3.6.1.4.1.311.10.11.83, identified in the WinCrypt.h header file as CERT_ROOT_PROGRAM_CERT_POLICIES_PROP_ID. Since this attribute is unique for EV root certificates, this must be the key for the EV enabling mechanism.

After striping the two header bytes, the content of this attribute turns out to be an ASN.1 data structure with the following content:

```
   0 30   35: SEQUENCE {
   2 30   33:   SEQUENCE {
   4 06   11:     OBJECT IDENTIFIER '2 16 840 1 113733 1 7 23 6'
  17 30   18:     SEQUENCE {
  19 30   16:       SEQUENCE {
  21 06   10:         OBJECT IDENTIFIER '1 3 6 1 4 1 311 60 1 1'
  33 03    2:         BIT STRING 0 unused bits
            :           '00000011'B
            :           Error: Spurious zero bits in bitstring.
            :         }
            :       }
            :     }
            :   }
```

This simple ASN.1 structure contains mainly two OIDs and a bit string. Let's have a look at the OIDs and the bit string:

OID 2.16.840.1.113733.1.7.23.6:
The first OID can be found in the document VeriSign EV CPS v. 3.3 http://www.verisign.com/repository/CPS/VeriSignCPSv3.3.pdf) on page 87:

> *Each EV Certificate issued by VeriSign to a Subscriber will include VeriSign's EV OID in the certificate's certificatePolicies extension. VeriSigns EV OID used for this purpose is 2.16.840.1.113733.1.7.23.6*

OID 1.3.6.1.4.1.311.60.1.1:
The second OID is in the Microsoft OID namespace but a Google search yields no results, however a clue can be found in the Windows Platform SDK again where the WinCrypt.h file contains the following definition:

```
#define szOID_ROOT_PROGRAM_FLAGS          "1.3.6.1.4.1.311.60.1.1"
```

BIT STRING '00000011'B
No clue can be found on the bit string and the meaning of the bits set. However we can assume that the bits finally enable the green bar.

All EV issuing root certificates found on the CTL feature the same structure beneath the policy OID with the same bit string.

Enabling our own CA for EV

Based on our analysis of the CERT_ROOT_PROGRAM_CERT_POLICIES_PROP_ID property, we can now try to enable our own, fake root certificate for EV. The properties found along with the certificate in the CTL are also present in the root certificate store after the CTL was processed. While the content of the CTL is signed, the properties in the certificate store are not cryptographically protected. Fortunately for us, the CryptoAPI provides all the necessary methods to query and set those properties.

Prerequisites

- We create a fake self signed root certificate (Structure and content does not really matter, for this proof of concept we used *cn=keyon* as the issuer and subject DN)
- We create a server certificate with *cn=www.behind-the-green-bar.com, o=keyon, c=CH* issued by our fake root certificate which has a certificate policies extension featuring an EV OID and the structure required by the EV standard. (Note that we could make up our own OID, however taking an existing OID simplified the proof of concept.). The certificate policies extensions looked as follows:

```
SEQUENCE {
   OBJECT IDENTIFIER '2 5 29 32'
   OCTET STRING, encapsulates {
       SEQUENCE {
         SEQUENCE {
           OBJECT IDENTIFIER '2 16 840 1 113733 1 7 23 6'
           SEQUENCE {
             SEQUENCE {
               OBJECT IDENTIFIER '1 3 6 1 5 5 7 2 1'
               IA5String 'https://www.verisign.com/rpa'
               }
             }
           }
         }
       }
   }
```

- Since EV requires revocation checking we also issue an empty CRL. (We issued a long life CRL valid for a year)
- Finally we set up an SSL server which uses our issued server certificate

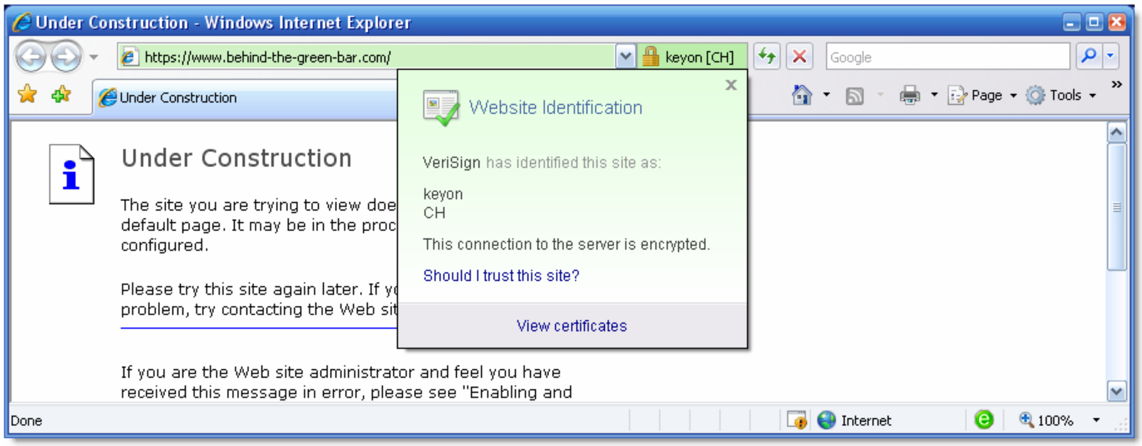
Enabling our CA for EV on a Windows XP machine

- We import the root certificate into the user root store using the appropriate CryptoAPI functions. In this import process, we also set the CERT_ROOT_PROGRAM_CERT_POLICIES_PROP_ID property for the certificate to have the same binary contents as the original property of the VeriSign root certificate found in the CTL. We also set the friendly name of the fake root certificate to *VeriSign*.
- We import the CRL using the explorer context menu. (This step is not needed if the CRL is available online and the certificate has a CRL distribution point pointing to the online CRL location.)

Connect to the server with IE7

- We add an entry to %SystemRoot%\system32\drivers\etc\hosts which resolves our fake server name www.behind-the-green-bar.com to the static IP address of the server. (This step needs administrative rights, however it is not necessary if the server is registered in the DNS system)
- For the green bar to work, either the *Phishing Filter* or *Check for server certificate revocation* must be enabled in IE7 as a prerequisite.
- We connect to our server https://www.behind-the-green-bar.com/ in IE7

Bingo! We have a green bar when we connect to our server:


Under Construction - Windows Internet Explorer
https://www.behind-the-green-bar.com/
keyon [CH]
Google
Under Construction
Page
Tools
Done
Internet
100%

Under Construction

The site you are trying to view doe
default page. It may be in the proc
configured.

Please try this site again later. If y
problem, try contacting the Web sit

If you are the Web site administrator and feel you have
received this message in error, please see "Enabling and

Website Identification

VeriSign has identified this site as:

keyon
CH

This connection to the server is encrypted.

Should I trust this site?

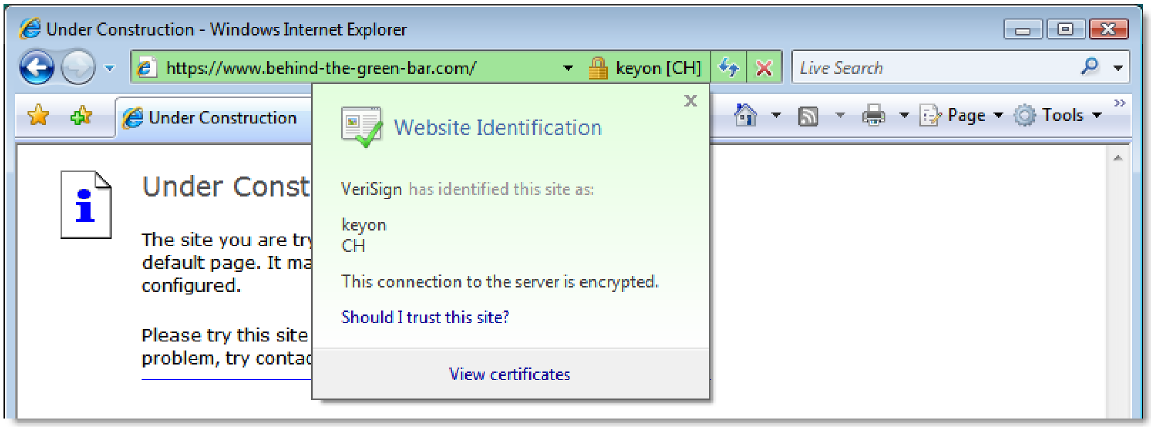
View certificates

Note that the friendly name of our fake root certificate is used for the company that has ‘identified’ the server. The friendly name is again not part of the certificate and as a simple certificate store property is not cryptographically secured.

The site information is taken from the server certificate subject DN and can be made up by the issuer of the server certificate at will.

Windows Vista

Once the fake EV root is imported the same way as with Windows XP we have the green bar under Vista as well:



Website Identification

VeriSign has identified this site as:

keyon
CH

This connection to the server is encrypted.

Should I trust this site?

View certificates

Fake EV Root Certificate Import

The critical step on the client was the import of the fake root certificate and the setting of the EV property. The question thus is how easy can an attacker import such a certificate?

The import with our small tool that was tested under Windows XP Professional and Windows Vista Enterprise with the latest security patches available as of June 2nd. The following table shows the result of our tests:

OS	Logged in account type	Import to User Store	Import to Machine Store
Windows XP Professional	Standard User	Yes, with programmatically confirmable notification	No
	Administrator	Yes, completely silent	Yes, completely silent
Windows Vista Enterprise	Standard User	Yes, with programmatically confirmable notification	No
	User in Administrative Group	Yes, with programmatically confirmable notification	No
	Administrator	Yes, completely silent	Yes, completely silent

In all cases it was possible to add our fake EV root to at least the user root store by starting a small executable on the system. Adding the fake root certificate to the local machine store will make the certificate available to all users of the system. Interestingly, no confirmation dialog was shown when an Administrator imported the root certificate to his user or the local machine store. This insecure behavior of a totally silent import can be verified using e.g. the certificates snap-in of mmc to import a root certificate.

If a notification dialog was shown asking the user to confirm the import, it was possible to confirm the import from the import application itself using the simple method described in [4]. Only a short flicker of the dialog was noticed in this case. Even if a user would notice the root certificate confirmation dialog, he could almost certainly not determine which root certificate was imported if the subject is crafted after e.g. one of the many VeriSign certificates.

In a more primitive variant, the root certificate import using a small tool could be justified e.g. by telling the user that he must install the root certificate in order to use a freeware application or access an interesting web site.

Dangers of advanced Phishing attacks making use of the trust implied by the green bar

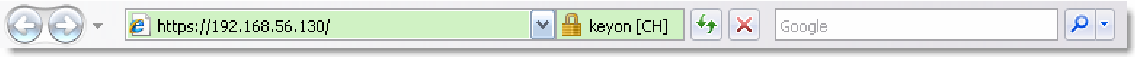
The automated root key import when combined with a remote code execution web attack could lead to the following possible Phishing attack under Windows XP:

1. Send out Phishing mails which trick the user to connect to a web site. The mail content could for example tell the user the following persuasive story:

> Dear customer of X,
>
> We have now new extended validation certificates in place to enhance the security of our secure web site and protect you from Phishing websites. You can learn more about EV certificates under the following links:
>
> http://www.microsoft.com/windows/products/winfamily/ie/ev/default.mspx
>
> http://www.microsoft.com/windows/products/winfamily/ie/ev/security.mspx
>
> http://www.verisign.com/ssl/ssl-information-center/faq/extended-validation-ssl-certificates.html
>
> You can connect to our new EV secure server using the following link
>
> <URL with IP pointing to the server which uses the remote code vulnerability>
>
> The server uses an IP address during a transition period to allow you to access the old server if you do not want to make use of the higher security provided by EV certificates.
>
> Note: You must make sure that the company name is shown in the green bar. If the company name is not shown as explained by the Microsoft or VeriSign EV website, please do not login or enter any data as it is possibly a Phishing website. Only the green EV bar ensures that you are connected to a valid server of our company.
>
> …

2. Upon load the web site runs a fake EV root import code using a remote code execution vulnerability (e.g. using a mechanism similar to the recent WMF exploit).

3. The user is automatically redirected by the web site to the target fake EV website running on a different IP address after a few seconds. If the import worked, the user will see the green bar after the redirection. Note that the URL can contain an IP address instead of a DNS:



IE7 shows a green bar for certificates with an IP address as the common name, although the CA/Browser forum EV guidelines mandate to use domain names:

> *commonName: This field MUST contain one or more host domain name(s) owned or controlled by the Subject and to be associated with Subject's publicly accessible server.*

A user may trust such an IP identified web site since the company name is shown in the green bar and detail information.

Under Windows Vista IE7 runs with very low privileges so this direct kind of attack may not work. An advanced Phishing attack for Vista would thus probably require the user to run a disguised importer first. (Or the Phishing mail tells the user to enable EV on his system by download and running an application first).

However since most of the users still run Windows XP, such an exploit of the trust implied by the green bar may be highly successful.

Conclusion

While the EV standard defines the rules under which a CA may issue EV certificate, the technical implementation in IE7 does currently not match the security level of the issuing process.

The enabling mechanism if a root certificate can issue EV certificates is only a simple property in the certificate store and not cryptographically secured (e.g. signed) in any way. If an attacker can force or trick the user into running a program that imports the root certificate and creates the appropriate properties for the certificate, he can create his own server certificates that will show the green bar. Since he can select the issuer and subject DNs similar to those of well known CA, a user has no other way to determine if the site certificate does really belong to the entity shown but by verifying the fingerprint of either the root certificate or the end entity certificate.

For Windows XP it is at the time of writing possible to add a fake EV enabled root certificate completely silently if the user works as an administrator, which is unfortunately the most widely used configuration today. In all other cases, including Windows Vista, the root certificate import confirmation dialog can be programmatically confirmed without any user interaction required.

If a marketing campaign breaks EV down to *'if it's green and the company name shown is ok, you're safe'* this is simply unjustifiable with the current implementation of IE7. Since the green bar shows the company, users may even pay less attention to the address entered or shown in the address bar.

A Phishing attack using a fake root certificate import along with a hosts file manipulation which directs the user to a fake server even when he enters the server name himself in the address bar could make it practically for a user to detect a man in the middle attack impossible (the user must known the correct fingerprints to detect this). The green bar and all presented information would assure him that he is connected to the intended server while in fact he is connected to the server of an attacker.

Without a better security model in the EV implementation, i.e. a cryptographically secured method to define EV allowed root certificates the green bar is currently not an effective means to prevent clever Phishing attacks.

References

[1] Guidelines for Extended Validation Certificates, CA/Browser Forum, http://cabforum.org/EV_Certificate_Guidelines.pdf

[2] Implementing Extended Validation Certificates for Internet Explorer 7, Microsoft, http://technet.microsoft.com/en-us/ie/bb381619.aspx

[3] Extended Validation, WIKI http://en.wikipedia.org/wiki/Extended_Validation_Certificate

[4] Installing Fake Root Keys in a PC, Adil Alsaid and Chris J. Mitchel, http://www.isg.rhul.ac.uk/~cjm/ifrkia2.pdf

Trademark Notice

All company and product names are only used for reference purposes. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.