RSA-512 Certificates abused in the wild

During recent weeks we have observed several interesting publications which have a direct relation to an investigation we worked on recently. On one hand there was a Certificate Authority being revoked by Mozilla, Microsoft and Google (Chrome), on the other hand there was the disclosure of a malware attack by Mikko Hypponen (FSecure) using a government issued certificate signed by the same Certificate Authority. That case however is not self-contained and a whole range of malicious software had been signed with valid certificates. The malicious software involved was used in targeted attacks focused on governments, political organisations and the defence industry. The big question is of course, what happened and how did the attackers obtain access to these certificates? We will explain here in detail how the attackers have used known techniques to bypass the Microsoft Windows code signing security model.

Recently Mikko Hypponen wrote a blog on the F-Secure weblog (http://www.f-secure.com/weblog/archives/20100226.html) detailing the discovery of a certificate used to sign in the wild malware. Specifically this malware was embedded in a PDF exploit and shipped in August 2011. Initially Mikko also believed the certificate was stolen, as that is very common in these days, with a large amount of malware families having support, or optional support, for stealing certificates from the infected system. Apparently someone Mikko spoke to mentioned something along the lines that it had been stolen a long time ago. During the GovCert.nl symposium Mikko mentioned the certificate again, but now he mentioned that according to the people involved with investigating the case in Malaysia it likely wasn’t stolen.

The reason why Mikko looked at this specific sample and this certificate was likely the recent revocation of Digisign Server ID (Enrich) by Microsoft (http://technet.microsoft.com/en-us/security/advisory/2441491) and earlier by Mozilla (http://blog.mozilla.org/security/2011/11/09/verifying-trust-in-digisign-cert-ssl-what-intermediate-certificate-authority/). The interesting part in those articles is that Microsoft does not mention anything about the code signing abilities of certificates while Mozilla does. Microsoft does mention that the certificates were not fraudiously issued but were duplicated due to cryptographically weak keys. The option of stolen certificates is left completely in the middle here.

The whole commotion around Digisign was actually caused by an investigation by Fox-IT completed in mid-October, in which we recovered a signed executable embedded in exploits and downloaded additionally by any of the executables. While our investigation did not focus on the signing of those executables, the report was shared in the relevant community, and if you looked at the 4 certificates initially found, it was easy to determine that all were 512 RSA and used on HTTPS websites, which were still up at the time of writing. Later during our investigation we encountered 5 more certificates which were still at use to writing. With the lifetime of these certificates being a couple of years, the attackers had plenty of time to do the factoring.

So the reason why Digisign Server ID (Enrich) (DigiCert Sdn. Bhd) was revoked was because their certificates had no CRL in the certificate which allowed to easily revoke the certificate. Also all those certificates were issued without a purpose, in which case the certificates can be used for anything. The certificates we found to be used in the wild recently are:

Subject: C=AN, O=Digicert Sdn. Bhd., CN=SSL Secure.com
Issuer: C=US, O=VeriSign, Inc., CN=VeriSign Secure Server CA
Subject: C=CA, O=GlobalSign Inc, CN=GlobalSign SureSign
Issuer: C=US, O=Thawte, CN=Thawte
Subject: C=US, O=Thawte, CN=Thawte
Issuer: C=US, O=Thawte, CN=Thawte
Subject: C=CN, O=China Telecom, CN=China Telecom
Issuer: C=CN, O=China Telecom
Subject: C=CN, O=China Telecom, CN=China Telecom
Issuer: C=CN, O=China Telecom

Additionally an external party found several other samples which contained 512 bit RSA certificates signed by DigiCert Sdn. Bhd:

Issuer: C=MY, O=Digicert Sdn. Bhd., CN=SSL Secure.com
Subject: C=MY, O=Digicert Sdn. Bhd., CN=www.esupplies.com
Issuer: C=MY, O=Digicert Sdn. Bhd., CN=SSL Secure.com
Subject: C=MY, O=Digicert Sdn. Bhd., CN=www.457608.com
Issuer: C=MY, O=Digicert Sdn. Bhd., CN=SSL Secure.com

One of those samples was found in August 2011, and possibly used back in March 2010, indicating how long this issue has been going on without any clear action from the industry. Microsoft whose platform has been targeted by this is the victim of this, and I think that Microsoft should not have relied on weak security properties for a security solution that can apparently be bypassed by parties far outside of the control of Microsoft. Microsoft could simply deny verification of executables which have been signed with a 512 bit RSA key after a certain date, as 512 bit RSA has been considered weak for a long time. For the article at TechNet it is clear that Microsoft understand the problem and that they have acted on this accordingly. But the question is if it was not a bit late. Also interesting is that none of the samples have an actual timestamp, we think this is another design decision made that makes those executables pass verification, but it might cause the executables to no longer pass verification after the certificate has expired, we were unable to test this however.

Also one certificate that was used, ehi.anthem.com, did not have the "Digital Signature" property in "Key Usage", thus it should not have passed verification. But we wonder if that indeed is true, as why would the attackers go through great lengths of factoring the RSA key and using it to sign their executables, if it did not pass verification? Either the attackers overlooked something here, or the digital signature verification system in Windows is at fault. We are however unable to verify this as the relevant certificate has expired in April 2011.
So the problem will solve itself eventually with CAs no longer signing 512 bit and more attention is given on the subject. Th

So the problem will solve itself eventually with CAs no longer signing 512 bit and more attention is given on the subject. Th