PKI Layer Cake: New Collision Attacks against the Global X.509 Infrastructure

Dan Kaminsky[1], Meredith L. Patterson[1], and Len Sassaman[2]

[1] IOActive, Inc.
[2] Katholieke Universiteit Leuven

1 Introduction

Research unveiled in December of 2008 [15] showed how MD5's long-known flaws could be actively exploited to attack the real-world Certification Authority infrastructure. In this paper, we demonstrate two new classes of collision, which will be somewhat trickier to address than previous attacks against X.509: the applicability of MD2 preimage attacks against the primary root certificate for Verisign, and the difficulty of validating X.509 Names contained within PKCS#10 Certificate Requests. We also draw particular attention to two possibly unrecognized vectors for implementation flaws that have been problematic in the past: the ASN.1 BER decoder required to parse PKCS#10, and the potential for SQL injection from text contained within its requests. Finally, we explore why the implications of these attacks are broader than some have realized — first, because *Client* Authentication is sometimes tied to X.509, and second, because Extended Validation certificates were only intended to stop phishing attacks from names similar to trusted brands. As per the work of Adam Barth and Collin Jackson [4], EV does not prevent an attacker who can synthesize or acquire a "low assurance" certificate for a given name from acquiring the "green bar" EV experience.

The attacks we will discuss in this paper fall into the following categories:

1. **MD2RSA Signature Transfer:** Verisign's MD2 root can be exploited by creating a malicious intermediate with the same MD2 hash as its parent and transfering the signature from the root to the malicious intermediate.
2. **Subject Name Confusion:** Inconsistent interpretation of the Subject X.509 Name in a PKCS#10 request can cause a CA to emit a certificate for an unauthorized Common Name. Existing PKCS APIs vary in their handling of Common Names and Subject Names, and these differences can be exploited in a number of ways which we will explore in detail.
3. **PKCS#10-Tunneled SQL Injection:** Certificate Authorities inserting PKCS#10 Subject Names into a database do not necessarily employ comprehensive string validation for the BMPString, UTF8String and UniversalString types, which allows SQL injection attacks. Given the special trust that a CA's database backend presupposes for the rest of the Internet, SQL injection is especially problematic.
4. **PKCS#10-Tunneled ASN.1 Attacks:** Certificate Authorities exposing a PKCS#10 receiver may be exposing unhardened ASN.1 BER listeners.

R. Sion (Ed.): FC 2010, LNCS 6052, pp. 289–303, 2010.

ASN.1 BER is tricky to parse, with many possibilities for consistent and predictably exploitable attack surfaces. The PROTOS project [12] found a large number of vulnerabilities via the SNMP consumer, but it is possible that some of the ASN.1 BER parsers found in commercial CA implementations were not covered in the 2002 PROTOS lockdown and thus are still vulnerable.

5. **Generic SSL Client Authentication Bypass:** The MD2 attacks in this paper may have larger implications in certain deployments. An attacker with the ability to directly issue certificates — rather than just the ability to get an arbitrary X.509 Subject Name past a validator — gets access to the "Client Authentication" EKU (Extended Key Usage) attribute that controls whether a certificate allows for authenticating a client to a server. Since Root CAs do not normally issue certificates with "Client Authentication" set, some systems may not test for what would happen if such a certificate arrived. This may create a generic authentication bypass in some systems. A similar bypass may be extended from Stevens and Sotirov's MD5 collisions, in situations where the Client Authentication EKU (which is not present in the root certificate they attacked) is insufficiently validated.
6. **EV Hijacking:** EV certs were designed to address phishing attacks where a bank at `www.bankoffoo.com` is suffering attacks from the owner of `www.bank-of-foo.com` or `www.bankofoo.com`. They were specifically not designed to deal with the case where an attacker has a certificate, even a low assurance certificate, for `www.bankoffoo.com`, and the attacker has a DNS or other route manipulation attack, e.g. DNS cache poisoning [19]. Barth and Jackson have shown that browsers do not enforce a scripting barrier between `https://www.bankoffoo.com` (EV certified) and `https://www.bankoffoo.com` (Low Assurance certified). Thus, an attacker need simply proxy enough of an SSL session to get the main HTML of a page loaded in EV (thus causing the green bar), then kill the TCP session. After that, the attacker can host any script from the Low Assurance cert, and that script will inevitably be merged with the real site with no negative impact on the EV experience.

We will summarize the recent history of attacks against the CA infrastructure; describe the methodology used to discover the attacks listed above; investigate these attacks in greater detail; outline a principled approach for remediating these issues and what steps browser manufacturers, cryptographic API manufacturers and certificate authorities need to take; and finally, identify directions for future work.

2 Background

The SSL protocol [2] is used for encrypting reliable data flows from one endpoint to another. But encryption without authentication is worthless: one can easily end up encrypting information with the key of an attacker! SSL manages authentication via *certificates* — assertions of identity that are cryptographically