2013

Exchanging Demands: Weaknesses in SSL Implemenations for Mobile Platforms

Peter Hannay
Edith Cowan University, p.hannay@ecu.edu.au

Clinton Carpene
Edith Cowan University, c.carpene@ecu.edu.au

Craig Valli
Edith Cowan University, c.valli@ecu.edu.au

Andrew Woodward
Edith Cowan University, a.woodward@ecu.edu.au

Mike Johnstone
Edith Cowan University, m.johnstone@ecu.edu.au

DOI: 10.4225/75/57b5646ccd8e3
Originally published in the Proceedings of the 11th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia, 2nd-4th December, 2013

EXCHANGING DEMANDS: WEAKNESSES IN SSL IMPLEMENTATIONS FOR MOBILE PLATFORMS

Peter Hannay, Clinton Carpene, Craig Valli, Andrew Woodward, Mike Johnstone
Security Research Institute, Edith Cowan University, Perth, Australia
p.hannay@ecu.edu.au; c.carpene@ecu.edu.au; c.valli@ecu.edu.au;
a.woodward@ecu.edu.au; m.johnstone@ecu.edu.au

Abstract
The ActiveSync protocol's implementation on some embedded devices leaves clients vulnerable to unauthorised remote policy enforcement. This paper discusses a proof of concept attack against the implementation of ActiveSync in common Smart phones including Android devices and iOS devices. A two-phase approach to exploiting the ActiveSync protocol is introduced. Phase 1 details the usage of a man-in-the-middle attack to gain a vantage point over the client device, whilst Phase 2 involves spoofing the server-side ActiveSync responses to initiate the unauthorised policy enforcement. These vulnerabilities are demonstrated by experiment, highlighting how the system can be exploited to perform a remote factory reset upon an Exchange-integrated Smart phone.

Keywords
ActiveSync, Exchange, Man-in-the-Middle, Secure Sockets Layer (SSL), Android, iOS

INTRODUCTION

The use of mobile phones is continuing to increase, while they concurrently become interconnected mobile computing devices with ever increasing computing power. Initially, this domain was the domicile of limited phone services on limited handset devices, with limited data speed and connectivity (Li and Lyons 2012). These devices evolved into Smartphones, which are effectively handheld computers which are also able to make phone calls. The latest progression in the evolution of the mobile phone is the development of the tablet, which incorporates general computing power, data storage and widespread network access methods (Lee, Suh et al. 2012). The networks used by Smart phones now encompass a wide range of technologies and capabilities available for use in mobile, commodity networked devices. The protocols themselves are diverse and in constant flux, with significant change over time as both technology and services have evolved (Clarke 2013). Service is no longer restricted to voice and SMS but includes not only data and Internet services but also accurate geo-locational services using the GPS networks, wireless networks and phone cell towers. The current generation of smart phones have considerable computation, multimedia applications and have increasingly accurate locational services (Poslad 2011).

The Smart phone's ubiquity and increasing capabilities has seen them become repositories, often singular, for our highly personal and private information. As of June 2012, over 48% of Australians accessed the Internet using a mobile wireless connection (ABS 2012), and email was one of the most heavily utilised services. Corporate users regularly synchronise a range of corporate data to their Smart phone's (Ferrer, Camacho-Martinez et al. 2013), and again email would be the most frequent of the data accessed, corporate email data often contains sensitive or otherwise valuable information for attackers. These factors make the Smartphone an attractive target for attackers (Wang, Streff et al. 2012). The Smartphone market is dominated by Android based devices, with a significant number of iPhones also being used for corporate data. The use of Blackberry phones and Smartphones has decreased dramatically, although it was the phone of choice for those requiring secure communications (Carton 2008). The increase in production of malcode for Android Smartphone's has almost been geometric,

with one report indicating that in the third quarter of 2012, 51447 unique samples were detected, up from 5033 in the previous quarter (F-Secure 2012). Apple's iOS has not seen the same level of malicious code targeting as Android devices, but nonetheless jail broken Apple iOS devices are also prone to infection (F-Secure 2012).

This paper outlines a proven attack method that exploits a vulnerability in certain implementations of the ActiveSync protocol. ActiveSync is a trusted protocol to exchange mail between devices and servers participating in Microsoft Exchange environments. Microsoft Exchange has unique relationship with its clients. Through the ActiveSync protocol various messages are exchanged between the parties. These messages can include control messages, policy messages, and many other functions that enable various Exchange services. As an example, policy messages may be issued from a server to enforce passcode restrictions on a client device. The exploit explored in this paper focuses upon issuing unauthorised policy messages to MS Exchange clients. The attack is comprised of two phases. The first phase involves establishing a man in the middle condition through the use of a hostile access point known as the pineapple. The second phase of the attack is to issue a command to initiate a remote device wipe.

PHASE 1: MAN IN THE MIDDLE

Achieving MitM for the attack

A man in the middle (MitM) attack involves intercepting the flow of communications between devices. In order to pose as the Exchange server we needed to establish a MitM connection as described above. In this instance we achieved this through the use of a Wi-Fi Pineapple. The Wi-Fi pineapple is a device that listens for beacons sent by prospective Wi-Fi clients searching for remembered access points on a network. Once receipt of these probes the interdicting device broadcasts an SSID inviting the client to connect to it. This solicits connections from any device that has an insecure Wi-Fi network saved in its configuration (Sood and Enbody 2012). Figure 1 gives an overview of the typical communication paths for mobile ActiveSync clients that are not under attack.

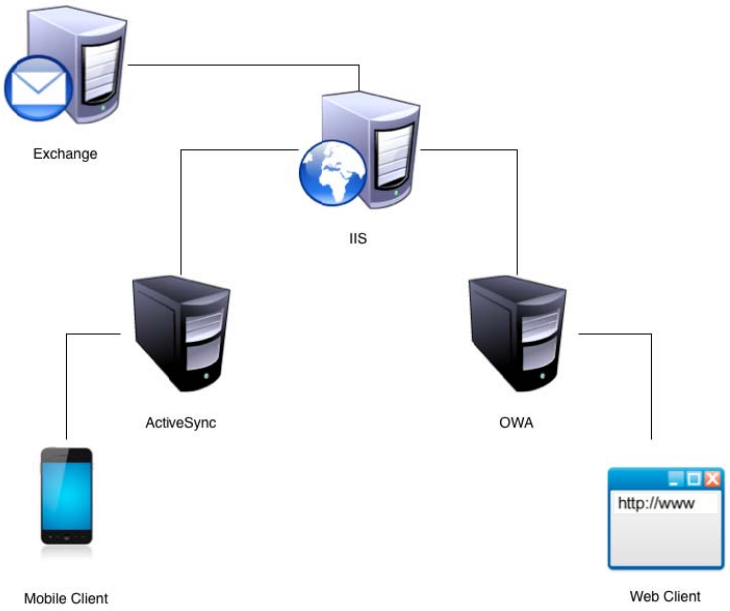

Figure 1: The flow of data between an Exchange Server and Mobile / Web Clients

Once the device is ensnared by the interdicting device (in this case the Wi-Fi pineapple), the next phase is to provide fake DNS responses and accept connections intended for an Exchange server (Figure 2). This outcome was achieved through the use of custom Python scripts running on an auxiliary computer listening and responding to DNS requests. It is at this point that potential certificate issues may arise. It is expected that a device would ensure that secure socket layer (SSL) certificates are valid, having been signed by a root certificate authority (CA), and valid in terms of the certificate's common name matching the server address being represented. Based on the results of this verification process, the device will either connect, or not connect depending on how SSL is handled.

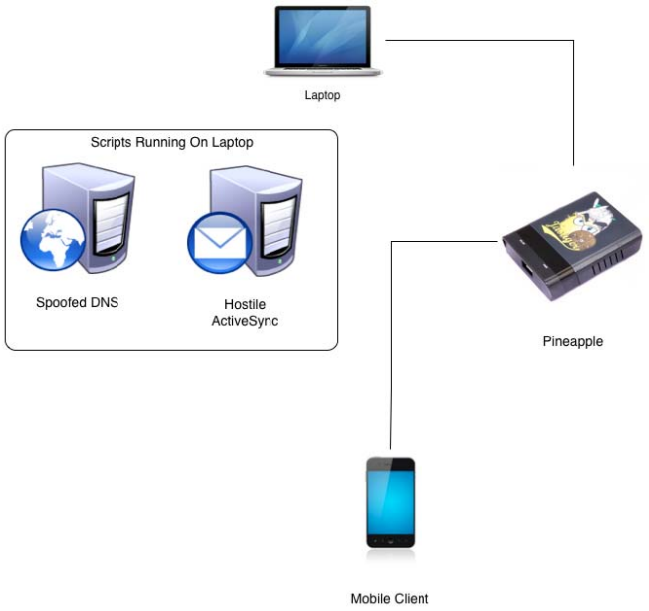

Figure 2: The mobile client communicating with the attacker

It is at this point that the certificate issue comes up it is expected that a device would ensure SSL certificates are valid, having been signed by a root Certificate Authority and valid in terms of the address being represented, this has been tested. Based on the results above the device will either connect, or not connect, depending on how SSL is handled.

PHASE 2: INITIATING THE WIPE

The vast majority of Exchange mail protocols and mobile device deployments make use of SSL to implement some level of security. In order to successfully accept the connection from the device we need to negotiate an SSL handshake with the server. It is assumed that we do not possess the private key for the Exchange server to which our victim is attempting a connection. As such we will be making use of a self-signed certificate to accomplish the attack. Self-signed certificates can be generated by anyone, and without having to undergo any kind of verification process that is normally associated with a trusted root signed certificate. The details of the certificates do not need to match those of the intended legitimate servers. This is significant because the lack of verification of the certificate completely bypasses the authentication purpose for which certificates are intended.

The issue allows for certain phones to be remotely wiped assuming a MiTM attack can be achieved. It should be noted that at no time is the Exchange server required to be part of the conversation, however it does not need to be excluded either.

Issuing Remote Wipe

With the device connected it is a simple process to issue a remote wipe of that device, on any connection from an email client when an HTTP 449 error is sent. The 449 error demands the remote device issue a PROVISION request prior to continuing communication, this is shown in figure 3 (Microsoft 2008). When the device makes a PROVISION request (essentially requesting new policy) we respond with a policy command initiating a remote wipe of the device, this command is shown in Figure 4.

```
HTTP/1.1 449 Retry after sending a PROVISION command
Cache-Control: private
Content-Type: text/html
Server: Microsoft-IIS/7.5
MS-Server-ActiveSync: 14.0
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Tue, 08 May 2012 07:08:22 GMT
Content-Length: 54

The custom error module does not recognize this error.
```

Figure 3 – The HTTP Error 449

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/vnd.ms-sync.wbxml
Server: Microsoft-IIS/7.5
MS-Server-ActiveSync: 14.0
Date: Tue, 08 May 2012 07:00:04 GMT
Content-Length: 123

..j...EK.1..FGH.MS-EAS-Provisioning-WBXML..K.1..I.
2761868790..JMN.0..O.0..Q.0..P.0..S.1..T.4..U.900.
.V.8...X.1...Z.0.......
```

Figure 4 – The Wipe Command

Impacted Devices

In testing we evaluated Android, iOS & Windows Phone 7.5. We set up two legitimate Exchange servers, both running Exchange Server 2010. The first server made use of an expired self-signed certificate, the second made use of a certificate signed by a valid CA. The attack was then conducted to issue remote wipe commands with a third, self-signed certificate.

Tested Device	Self-Signed Cert	Trusted Cert
Android (2.3 & 4.0)	Wipe, no prompt	No wipe (err)
iPhone/iPad (iOS 6.0.1)	Wipe, cert prompt	Wipe, cert prompt
Windows Phone 7.5	No wipe (err)	No wipe (err)

Table 1 - Table of devices tested against the ActiveSync SSL exploit, as tested with self-signed and trusted certificates

From the results above, clients of Exchange servers making use of self-signed certificates (our research indicates this is the most common deployment style for small to medium businesses) are most vulnerable to this form of attack. Any handsets that participate in Microsoft Exchange systems configured in such a way are subject to; remote wipe without prompt for Android handhelds; and with a certificate error prompt for Apple iOS devices. In the case of Apple iOS devices, the prompt displayed was for a certificate error, providing no information about this error, but with a clearly available "continue" button. Windows Phone 7.5 provided no mechanism to easily accept a self-signed certificate: it had to be installed manually onto the device. Even when the certificate was changed by manual methods there was no easy mechanism to accept the new certificate. This indicates high resilience to this attack.

The clients of Exchange servers that were using certificates signed by a Trusted Certificate Authority fared somewhat better, with Android & Windows Phone devices simply refusing to connect to the service. Android devices displayed a security error, whilst Windows phone displayed a certificate error. There was no mechanism provided to continue connecting regardless in either case. The iOS devices tested provided a prompt to accept the new certificate, again with no advice and an easily available continue button.

Discussion

As previously highlighted in the introduction of this paper, the use of Smartphone's for both private and commercial use has significantly increased over the last five years. In particular, corporate users of these devices have moved away from what is arguably a secure device in the Blackberry, to more vulnerable platforms in Android and iOS devices (Plotkin 2011). These devices contain functionality to automatically synchronise email from corporate servers to these mobile devices, which are both logically and physically vulnerable. They are logically vulnerable in that this type of attack is easy to perpetrate against an unsuspecting user, and physically in that it can be easy to obtain local access to the device. The attack described in this paper highlights this fragility, in that a user operating a Smart phone within reasonable parameters, could still find them with a compromised phone. At this stage, whilst the attack simply wipes the phone, the vulnerability will be further explored to determine whether data can be stolen from a target device. The impact of having ones Smartphone wiped could range from mild annoyance, where there is strictly corporate data involved, through to potentially life threatening if the device in question was being used in a hospital or other medical facility. Although the attack process detailed in this paper ultimately results in the victim's device being factory reset, the attack could have more significant ramifications. Once the PROVISION notice is sent by the client, the attacker has access to the entire supported ActiveSync policy set to engage with the client (Microsoft 2008). This could involve specifying to the client that the destination Exchange server has changed to a server address the attacker operates, providing persistent control of the device.

An additional complication is the increasing prevalence of bring your own device (BYOD) in the corporate workspace (James and Griffiths 2012). It is feasible that an organisation could put steps and measures in place to reduce the level of risk associated with corporate use of Smartphones, but it may not always be possible to implement controls on a device for which administrator access is not available. Strong policy measures combined with appropriately resourced seat education awareness and training programs need to be implemented around the use of BYOD, as loss of corporate data on an employee's inadequately protected device could occur.

CONCLUSION

Mobile devices have been deployed widely in corporate environments, with a large number of these being connected to corporate Exchange servers. It has been determined that the most commonly implemented configuration of these servers leaves devices, and by extension a corporation, open to compromise. The vulnerability is primarily due to the way that SSL is implemented on the tested smartphone devices, without proper authentication or by simply allowing users to accept an invalid SSL certificate, bypassing the security which should prevent the attack. It is shown that using certificates, issued by trusted CAs, in some cases can mitigate the vulnerability, but this was not true for iOS. It is concluded that such actions should not be available to the end user. The responsibility for the vulnerability lies with the OS manufacturers, and therefore responsibility for correcting this vulnerability also lies with these developers to ensure correct implementations of certificate validation occurs. In the interim, use of appropriate security policy and user education should be employed as risk treatments.

REFERENCES

ABS (2012). 8153.0 - Internet Activity, Australia.

Carton, P. (2008). "iPhone vs. Blackberry: Which do consumers love most." ChangeWave Research.

Clarke, R. (2013). "Expanding mobile wireless capacity: The challenges presented by technology and economics." Available at SSRN 2197416.

F-Secure (2012). Mobile Threat Report Q3 2012, F-Secure**:** 1-40.

Ferrer, E., A. Camacho-Martinez, et al. (2013). "THE IMPACT OF MOBILE TECHNOLOGY ON ORGANIZATIONAL COMMUNICATION: RETHINKING THE SOCIAL PRESENCE THEORY." Continental Journal of Information Technology **6**(2).

James, P. and D. Griffiths (2012). "The Mobile Execution Environment: A Secure and Non-Intrusive Approach to Implement a Bring You Own Device Policy for Laptops."

Lee, K., E. Suh, et al. (2012). "A study on determinant factors to purchase for tablet PC and smartphone by a comparative analysis."

Li, Y. and B. Lyons (2012). "Market structure, regulation and the speed of mobile network penetration." International Journal of Industrial Organization.

Microsoft (2008). "[MS-ASWBXML]: ActiveSync WAP Binary XML (WBXML) Protocol Specification." Retrieved December 16th, 2012.

Plotkin, J. (2011). "BlackBerry: The FINRA Compliant Smartphone."

Poslad, S. (2011). Ubiquitous computing: smart devices, environments and interactions, Wiley.

Sood, A. and R. Enbody (2012). "Targeted Cyber Attacks-A Superset of Advanced Persistent Threats."

Wang, Y., K. Streff, et al. (2012). "Security Threats and Analysis of Security Challenges in Smartphones."