Investigating the leakage of sensitive personal and organisational information in email headers

Jason R. C. Nurse*, Arnau Erola, Michael Goldsmith, and Sadie Creese
Cyber Security Centre, Department of Computer Science, University of Oxford.

Abstract

Email is undoubtedly the most used communications mechanism in society today. Within business alone, it is estimated that 100 billion emails are sent and received daily across the world. While the security and privacy of email has been of concern to enterprises and individuals for decades, this has predominately been focused on protecting against malicious content in incoming emails and explicit data exfiltration, rather than inadvertent leaks in outgoing emails. In this paper, we consider this topic of outgoing emails and unintentional information leakage to better appreciate the security and privacy concerns related to the simple activity of sending an email. Specifically, our research seeks to investigate the extent to which potentially sensitive information could be leaked, in even blank emails, by considering the metadata that is a natural part of email headers. Through findings from a user-based experiment, we demonstrate that there is a noteworthy level of exposure of organisational and personal identity information, much of which can be further used by an attacker for reconnaissance or develop a more targeted and sophisticated attack.

Keywords: Email analysis, Information leakage, Digital forensics, Unintentional information exposure, Attack reconnaissance, Security and privacy risks

1 Introduction

For enterprises and individuals alike, the security and privacy of applications, systems and data is a crucial concern. Recent reports and the spate of successful breaches (e.g., at Target, JP Morgan and others [2, 11]) highlight the range of attacks being launched and the lengths to which criminals are willing to go to achieve their malevolent goals. Although there are a plethora of techniques that can be levied to compromise systems, one of the most prevalent is that of email. The classic case is that of an attacker sending an email infected with malware to an individual, that once accessed, automatically propagates to other individuals, possibly deleting or encrypting files or otherwise disrupting systems. The Melissa Virus was one popular example of this [3]. Email as an attack vector has become so significant, that it is now commonplace for organisations and email providers to conduct a range of security awareness campaigns. Amongst other things, these campaigns emphasise simple yet key points, such as not opening emails or attachments from unknown parties, and being vigilant for malicious individuals masquerading as existing contacts.

While the security of incoming emails is a challenge well worth the research effort that has been dedicated to it, we posit that there is another significant growing problem. This issue relates particularly to outgoing emails and the leakage of potentially sensitive information via them. In this paper we focus on this problem in an attempt to highlight the potential risks to the security and privacy of sensitive information, and also to encourage further research in the area.

Journal of Internet Services and Information Security (JISIS), volume: 5, number: 1 (February 2015), pp. 70-84

*Corresponding author: Cyber Security Centre, Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK, Tel: +44-(0)1865-273838, Email: jason.nurse@cs.ox.ac.uk

As a motivating example, let us assume the case where an attacker poses as a potential customer and sends an email to the sales department of a company requesting information on a product. The sales team quickly responds remotely with brief details on the product, but indicates that they will provide more detail once back in the office. The question of most interest to us here is: does the attacker gain any information about the individual or their enterprise? From the content, we can easily infer that they are not in the office, but can any other information be garnered? The answer to this question is a resounding 'yes', and the source of this information is the email headers.

Within every email, there is a set of metadata included called the *email header*. This metadata contains information about the sender and receiver of the email, and the route (servers) that the email traversed to go from sender to receiver [15, 18, 14, 16]. The problem with headers currently is the amount of detail on the sender that is increasingly being added by applications and systems, with little regard for the privacy implications. In the scenario above for instance, there is a very real chance that the attacker could examine the header to determine: whether the message was sent from a smartphone and if so, the type; the application used to send the email and its version number; the general location where the email was sent from; the hostnames, IPs and mail agents of the email servers used by the company; and even the security systems (inclusive of version details) implemented to protect the enterprise. This information could be further used by an attacker to gather intelligence or to develop a more sophisticated attack to target the individual or on their enterprise.

This paper concentrates on these security problems and aims to investigate the potential leakage of sensitive information via email headers. There are several articles that hint at the fact that sensitive information on the email sender can be discovered (e.g., [12, 19]) but little rigorous research on how, or to what extent. To address this gap, we have adopted an experiment-based approach where we collect a sample of emails from a range of enterprises and individuals and examine exactly what information is leaked, and whether such information could be used by an attacker.

The remainder of this paper is structured as follows. In Section 2, we examine the related work both on email security and the wider problem of information leakage and system fingerprinting. Section 3 introduces the research goals and experimental method that we apply to conduct our investigation. Next, Section 4 presents the findings from the experiment and discusses them in detail. We reflect on these findings and the more substantial issues of information leakage and the risk it poses in Section 5. Finally, Section 6 concludes the report and presents avenues for future research.

2 Related Work

Various investigations have been conducted in the past which aim to study the content and headers of email messages for abuse prevention (e.g., spam) and forensic analysis. In digital forensics for instance, content analysis is often used to determine the true authorship of email messages. In de Vel et al., the authors propose to use stylometry, the statistical analysis of variations in literary style between users [7]. Using machine learning, they were able to verify the authorship of the emails in a majority of cases. This gives a general idea of the ability to identify and use basic email content to gain insight, and in this case, useful attribution intelligence.

Considering email headers in particular, their use in the literature has been predominately on detecting abuse such as spam [4], and on tracing emails. In Al-Zarouni [1] for example, a detailed discussion is presented on how information in the email headers can be assessed to trace and ascertain the source of an email, even when there might be attempts to deceive an investigator. This highlights an important point with email headers, i.e., that they can, at least in part, be faked. Spam is one area where this tends to occur often, and also specialised cases where cyber-criminals create synthetic headers to impersonate a legitimate individual or user.

Fortunately, a detailed analysis of an email header can often help to detect incongruent information that might suggest a potentially fake email. For example, a mail-server which does not match the server of the email address; or, to detect a suspicious amount of extra fields, or indeed, unusually few fields in the header itself. A key point with respect to our current research, however, is the fact that although the analysis of the headers can help to detect misuse, it can also lead to a security or privacy risk when used for malevolent purposes (e.g., identifying the Internet Protocol (IP)addresses of an organisation's email servers for subsequent denial-of-service attacks).

A study on information leakage and its privacy implications pertinent to this discussion is that of Eckersley [8]. In that work, the author demonstrates that Web browsers (via HTTP request headers) can leak a variety of information on the browser itself and the computer used. This leakage is often to the extent that the information could allow Internet users to be fingerprinted (uniquely identified) by Web sites and even across multiple sites. The real privacy concern here is that all of this could occur without a user's knowledge or consent, and could be used for carefully targeted but unsolicited advertisements.

The leakage of information via HTTP request headers is a very similar problem to that faced today with email headers. Unfortunately there has not been much research directed towards the latter, as indicated in part by the length of our review, potentially because of greater attention on problems such as spear-phishing and spam. While these are important issues, there is an increasing risk to individuals and companies because of the leakage of information via email headers. As attackers continue to probe enterprises for vulnerabilities, email headers provide a wealth of insight that could be used as a basis for a range of attacks.

3 Research Aim and Approach

The aim of this research is to better understand the potential risk that individuals face based on emails that they send. Specifically, we seek to investigate the extent to which email headers currently leak potentially sensitive information about the sender and the company they work for, and how any information leaked might be used for malicious ends. To address this question, our research approach has two main steps as shown in Figure 1.

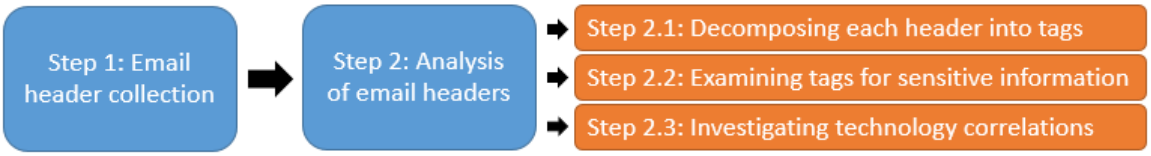


Figure 1: Research approach adopted in our study

The first step is recruiting participants and collecting their email headers. For recruitment we have chosen to use a convenience sample based on a set of enterprises known to the researchers. The main reason for this is that it allows us to target particular industries, while also increasing likelihood of participation in the study – such an experiment could be regarded as intrusive and therefore not palatable to some enterprises. Once we recruit a set of willing participants, they are then asked to send a series of emails (with no content) from their organisational email account to a specified email address.

In detail, we request that they send: (i) a fresh email from a computer mail client (e.g., Microsoft Outlook or Thunderbird); (ii) a reply to an email (to be sent to us from their computer mail client); (iii) a forwarded email (to be sent to us from their computer mail client); (iv) an email using the enterprise's Web-mail interface (using their browser of preference); and (v) an email from a mobile device (e.g., smartphone or tablet PC). This diversity in sources would allow us to gather data from a range of different

media to also investigate whether there might be any difference in what is exposed. To provide us with ground truth, participants were asked to include the type of email (i.e., (i)–(v)), the email client and the device used, in the subject line of the email.

The second step in our experiment is the analysis of the email header. Our objective in this stage is to examine the collected headers to determine exactly what information is exposed, how potentially sensitive that information might be, and if possible, identify where and when such exposures are likely to be found (e.g., whether it is specific to certain types of email client, device setups, or email servers). The main tasks in this analysis are:

1) Decomposing each header into a list of header tags. Figure 2 shows an example of header output and some tags commonly available, such as "Received", "From", "To", "X-Mailer" and "Message-ID".

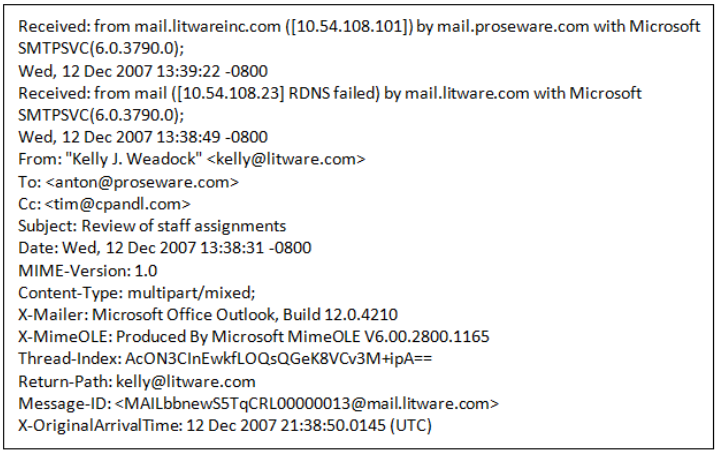


Figure 2: An example of an email header's tags (excerpt from [16])

2) Examining these tags individually for potentially sensitive information. For instance, tags such as "X-Mailer" can reveal the software used to send an email (in Figure 2, it is Microsoft Outlook), and the "Received" tag can indicate the type of email servers in use by the sender's organisation (the example in Figure 2 points to "Microsoft SMTPSVC(6.0.3790.0)" which is a version of an SMTP server bundled with Microsoft Exchange).

3) Investigating header tags for correlation to technologies. In detail, we seek to assess tags across the full set of participants to determine whether there is any correlation in what is exposed based on software, devices or email servers in use. In the simple case, if the sender is emailing from an Apple device, are we more likely to find "Apple", "Mac", or "iOS" in header tags? This assessment also has further implications in identifying the 'naturalness' of an email. For instance, if one knows that certain tags will be present if a colleague (who always uses Outlook) is the true sender of an email, and yet such tags are not present, this might indicate that the sender is not who they claim to be.

4 Experiment Findings and Discussion

4.1 Overview

In total, 75 individuals were contacted and of these, 50 agreed to participate. Overall, we gathered 225 emails from participants. There was a shortfall in the volume of email collected because some individuals did not set up work email on their smartphones (7 participants), others did not use a dedicated desktop

Information inferred	No. (%) of emails
Device IP	***79 (35%)***
- Device IP - internal (using 'Received')	56 (25%)
- Device IP - internal (using 'x-originating-ip')	20 (9%)
- Device IP - public (using 'Received')	3 (1%)
Device manufacturer (using 'Message-ID', 'MIME-Version')	***23 (10%)***
Device name (using 'Received' and 'helo=')	***3 (1%)***
Device operating system	***54 (24%)***
- Device operating system (using 'Message-ID')	14 (6%)
- Device operating system (using 'MIME-Version')	18 (8%)
- Device operating system (using 'Received')	1 (1%)
- Device operating system (using 'User Agent')	21 (9%)
Device type	***61 (27%)***
- Device type (pc — mobile)* [via 'Device OS']	48 (21%)
- Device type (pc — mobile)* [via 'Email client software']	45 (20%)
Email client software	***64 (28%)***
- Email client software (using 'User Agent')	26 (12%)
- Email client software (using 'X-Mailer')	38 (17%)
Email client type	***72 (32%)***
- Email client type (software — browser) (using 'Received')	6 (3%)
- Email client type (software — browser)* [via 'Client software']	51 (23%)
- Email client type (software — browser)* [via 'Device OS']	51 (23%)
Email server(s) hostname (using 'Received')	***225 (100%)***
Email server(s) IP (external or internal) (using 'Received')	***225 (100%)***
Email server(s) IPV6 capable* [via 'Email server IP']	***47 (21%)***
Email server(s) software and protocols <e.g. 'Microsoft SMTP Server', 'Exim', 'MAPI', 'ESMTP'>	***225 (100%)***
Email server(s) software (security) (using 'X-' eg IronPort, XDCC)	***58 (26%)***
Email server(s) software for authentication/encryption use <e.g. 'SMTPSA', 'TLS'>	***97 (43%)***
Email service(s) outsourced <e.g. to Google or Microsoft>	***57 (25%)***
Employer address	***170 (76%)***
- Employer address* [via 'Email server IP/name']	156 (69%)
- Employer address* [via 'Employer public IP']	36 (16%)
Employer name	***175 (78%)***
- Employer name* [via 'Email server IP']	156 (69%)
- Employer name* [via 'Employer public IP']	44 (20%)
- Employer name (using 'Organisation')	9 (4%)
Employer public IP (potentially company servers) (using 'x-originating-ip')	***49 (22%)***
Internal network IP configuration* [via 'Device IP'/'Server IP'/'NATs/DHCPs/WLANs']	***83 (37%)***
Internal network NATs/DHCPs/WLANs info <e.g. 'WLAN' names>	***23 (10%)***
Internal username (using 'X-..-Username', 'X-Authenticated-User')	***31 (14%)***
Internet Service Provider gateway	***24 (11%)***
- Internet Service Provider gateway (using 'Received')	***22 (10%)***
- Internet Service Provider gateway (using 'x-originating-ip')	***3 (1%)***
Internet Service Provider* [via 'ISP gateway']	***23 (10%)***
Internet Service Provider address* [via 'ISP gateway']	***23 10%)***
Language (using 'Accept-Language', 'Content-Language')	***157 (70%)***

Table 1: Summary of the information that could be gathered from the emails. The term 'using' defines the header tag that allowed the new information to be discovered, while the term 'via' defines a secondary information inference allowed by newly discovered data.

email client such as Outlook or Thunderbird (4 participants), and for a few others, their organisations did not support Web browser access (6 participants). Below we present highlights from our investigation into the emails gathered.

From the analysis of the email headers, we were able to discover several pieces of new information about participants and their enterprises. While a notable amount of disclosures can be expected because

of how email works (for instance, being able to view the email servers of an organisation that an email traverses or 'hops' through), some of the information found was somewhat surprising. For example, in a few cases we could identify the participant's internal company username. In Table 1, we summarise what information was inferred (about the individual and their organisation) and through which header tag, across the entire sample of emails. For each, we also present the percentage of emails in which that inference could be made (i.e., the new information derived).

In the following sections, we discuss the leakages presented in Table 1, both at the individual and the enterprise level. After this, we then report on our assessment of header tags and the correlation of tags with particular technologies.

4.2 Information Exposed at an Individual Level

4.2.1 Internal Username

Reflecting on Table 1, there are several areas of particular interest from a security and privacy perspective. The first, and possibly one of the most concerning, pertains to the exposure of an individual's internal username. This piece of information (which we note is typically different to the prefix before the '@' in an email address) was present in 14% of the emails assessed. Although its presence was not at all substantial, the fact that this is exposed is alarming. This is especially given that it constitutes half of the information needed to access an employee's user account, and it could also certainly help launch a convincing spear-phishing campaign. In terms of the conditions of exposure, from our assessment these usernames appeared to be exposed mainly because of how the email account was set up on the device. That is, to authenticate to the outgoing SMTP email server (or simply, to send emails from the device), this username would need to be provided as a prerequisite to emails being allowed through the system. This is not uncommon as it helps prevent outgoing servers being abused, but it is somewhat surprising that the sensitive information such as a username is not removed from onward transmitted emails.

As it pertains to the five types of emails considered, we found that the username tended to be leaked mostly in emails from mobile phones (33%) and least in emails from browsers (10%). The leakage via mobiles could be explained by the email setup previously mentioned, while the latter leakage may be the result of the company system configuration. A general point worth noting here and throughout the discussion below is that, because there was limited interaction with participants (and as a result of broad concerns regarding privacy), we were unable to confirm some of the details discovered about their systems. In some situations therefore, we mention what *might* be the cause of leakages as opposed to what was the definite reason.

4.2.2 Email Clients, Device Details and Device IPs

Another notable finding on an individual identity level was the leakage of information about email clients used, devices from which the emails were sent, and device IPs. For instance, we found that in 28% of emails, the email client used by the sender was included. This was inferred using the *X-Mailer* and *User-Agent* tags. Take the following header line for example: "`X-Mailer: iPhone Mail (11D257)`". In this *X-Mailer* tag, we can clearly see that iPhone Mail (with a build number of 11D257) has sent the email. The *User-Agent* tag also leaked a reasonable amount of information, for example, consider this line: "`User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101 Thunderbird/24.2.0`". Here, the email client (Thunderbird) can be identified as well as its version number (24.2.0). This can also be seen with other clients, for example, the header line: "`User-Agent: Microsoft-MacOutlook/14.4.3.140616`"; from this, we can easily identify that Microsoft Outlook for Mac, version 14.4.3.140616, has been used to send the email. From an attacker's perspective, knowing the version numbers of these clients can be especially advantageous in

crafting an attack. Take the case of the individuals using Thunderbird/24.2.0 for instance. This is a very outdated version of Thunderbird (31.0 is current), and more importantly, there are several vulnerabilities (many critical) that have been openly published regarding it [17].

In addition to its use in identifying email clients, the *User-Agent* tag was found to be partially useful (i.e., only in 9% of emails) at indicating the operating system (OS) of the device used by the sender. In the cases above, there are explicit mentions of iPhone (which is known to run the iOS operating system), Linux and Mac OS respectively. Two other noteworthy tags where device details tended to be leaked were in the *MIME-Version* and *Message-ID*. An example of the former is "`MIME-Version: 1.0 (Mac OS X Mail 7.3 (1878.6))`" (i.e., Mac OS X operating system), while an example of the latter is "`Message-ID: <mqfenfoimkeid.873487@email.android.com>`" (i.e., suggesting an implementation of the Android OS is in use by the individual).

An interesting point here is that in most cases where the *MIME-Version* leaked an OS, it was Mac OS X. In a similar manner, when the *Message-ID* leaked the OS, the Android OS was mainly mentioned. *Message-ID* did also mention Blackberry OS (RIM) a few times, suggesting a potential use of this tag particularly by mobile devices. From an exposure perspective, the leakage of such device information is concerning particularly in the first case (i.e., in the *MIME-Version* tag) because much more information is available, and it could potentially be used to prepare a targeted attack taking advantage of known (published) vulnerabilities of the specific system.

Although arguably not as important as a username, the device IP is another crucial piece of information (especially if it is public / routable) that was found in the emails assessed. Specifically, the sender's device IP was leaked in 35% of emails, most of which leaked their internal address through the *Received* and *X-Originating-IP* tags. An example of such a leakage is: "`Received: from [192.168.18.67] (dhcp.organisati.on [XXX.XXX.XXX.XXX]) by mailserver4.organisati.on (Postfix) with ESMTPSA`". This tag is the first (from bottom in the email header) and describes the hop from the network the sender's device is on, to the organisation's email server. Considering that this is an internal IP, there is little real exposure from the sender's perspective as the IP is not directly reachable (and thus, able to be easily targeted) by a malicious entity on the Internet. In the case of the three public IPs that were discovered in our analysis (see Table 1), however, there is no protection, and therefore there is a real risk to the individual and their employer. For instance, the IP could be scanned for open ports and then targeted with specific attacks (e.g., to hack into the device, and steal data or upload malware), or denial-of-service attacks might be launched to render the device unusable.

Reflecting on the leakage of information pertaining to email clients, device information and device IPs across the five email types, there were no significant unexpected differences in what was leaked. There are two points worth highlighting nonetheless. First, in the cases where an email was sent from a dedicated email client, we were able to discover that client's name from 35% of the emails by considering only the information explicitly mentioned in the tags. If an email was sent from a browser or mobile phone however, this leakage could only be seen 21% and 14% of the time respectively. To some extent, this can be expected given the increasing tendency to include more information about a client software in tags; this, of course, would not apply to email access using Web browsers. The second point is that device IP seems to be present slightly more when an email originated from a mobile phone — i.e., the IP was present in 33% of emails from a mobile phone, but only in 28% of the emails from a desktop client. This might be explained by the setup of the mobile devices (i.e., needing to authenticate to an SMTP server to send emails) or how they were connected to the network (e.g., the use of WiFi access points).

4.2.3 Internet Service Providers (ISPs)

From the emails gathered, we were also able to discover information about the Internet Service Providers (ISPs) used by the individuals; specifically, this was seen in 8% of emails. This information was

found predominantly in the first hop (i.e., the first *Received* tag) and was most prevalent in emails sent from mobile phones. Observe the following as an example: "`Received: from XXX.02.net ([XXX.XXX.XXX.XXX]) by mailserver6.organisati.on with esmtpsa (Exim 4.69)`". From this line of header text, we can find out that the sender was on the "`02.net`" network, which is operated by the UK telecommunications company O2. The assumption that could be made therefore, is that O2 is the individual's ISP (and possibly mobile phone operator). The main reason why mobile phones tend to leak this information more is simply because these hops would not be present if an email is sent from a desktop computer on a corporate network; we did, however, find theses hops in some cases where emails were sent from laptops on home computer networks.

Although the leakage of this information may not be critical from a privacy standpoint, the implications of an attacker knowing an individual's ISP can affect security. Most notably, social-engineering attacks might be conducted where a threat poses as an employee of O2 (e.g., over email, phone or in person) either to gather more information on the individual or for purposes such as fraud (e.g., requesting a bill be paid to a new account) [13].

4.2.4 Other Areas of Concern

Another notable finding was that connections to mail servers using IMAP or POP may leak more information than connections using MAPI (and generally Microsoft Exchange servers with clients using Microsoft Outlook). More specifically, email connections using the latter tended not to expose details about the sender's computer (e.g., IP, name) or network in the first *Received* hop, but instead, only recorded the hop between the first two servers (at least one of which, would be running Exchange). Reflecting on the risk of exposure because of the presence of this information, the main concern would be social engineering or spear-phishing attacks. For instance, an attacker might craft an email claiming to be the person's ISP and using other intelligence gathered (e.g., name, location, and even device details) to trick the individual into conducting some action or releasing even more sensitive data.

4.3 Information Exposed at an Organisational Level

4.3.1 Email Server Hostnames and IPs

At the organisation (or employer) level, we were able to find a reasonable amount of information about how enterprises setup their email systems. This was not indicative of careless deployments but rather a side-effect of how email and email services (e.g., forwarding, filtering, spam-checking) operate. For instance, given that email works by using hops between machines to traverse from one network to the next, in all emails we have been able to identify the hostnames and IPs (internal and external) from the email servers using the *Received* header tags.

Using the hostname and IP information, we, as any other message receiver, could then make further inferences about: (i) the Employer's name (in 78% of emails) and address (in 76% of emails) using Whois directories or IP Geo-locator services; (ii) IP versions in use — from the Server IP, it was possible to detect whether IPv4 and/or IPv6 was in use (IPv6 was seen in 21% of emails); and (iii) Internal network IP configurations (in 37% of emails), i.e., whether the company's internal networks are set up to use `10.*.*.*` or `192.168.*.*`, and so on. There is also the fact that one could map out a flow of email traffic through the enterprise by looking at the route that email takes through various organisational servers. Some of these hops could even indicate internal Dynamic Host Configuration Protocol (DHCP) gateway devices, and the names of WLANs; these were identified in 10% of emails, and usually from the first or second internal *Received* hops. All of this information could be useful to an attacker engaged in intelligence gathering about an enterprise, especially in preparation for highly targeted attack, potentially even using social engineering.

4.3.2 Email Server Software and Protocols

From our assessment, we discovered a few other key areas of exposure of the enterprise. One of these areas was the release of information pertaining to the email software and specific protocols in use by the servers through which the email passes; this was present in all of the gathered emails. A typical example of this leakage can be seen in the line: "`Received: from bluehalk.organisati.on ([XXX.XXX.XXX.XXX]) by mlserver2.organisati.on with esmtp (Exim 4.72)`". From this, one can identify that the Extended SMTP (ESMTP) protocol is in use by the mail servers and that the receiving server is running the Exim mail transfer agent [10], and specifically, an outdated (version 4.8 is current) and thus, potentially vulnerable version (see [9] for published details on Exim vulnerabilities). We might also further infer that a Unix-based operating system is installed on the receiving server because Exim was created expressly for this platform, and not for systems such as Microsoft Windows.

Another common protocol that was used to transfer email between mail servers was `MAPI` (or Messaging Application Programming Interface) – this is an interface that is commonly used by Microsoft Outlook to communicate with a Microsoft Exchange mail service. In those cases, we might therefore deduce that the receiving server has Exchange, and thus, the Windows operating system. Arguably these leaks might not be considered that significant from an exposure perspective, however, one way in which an enterprise may be targeted is by an attacker using the knowledge of email software (e.g., `Exim 4.80`) to research or craft dedicated security exploits.

4.3.3 Email Security Systems

Security-related information could also be leaked in email headers. Specifically, in 26% of emails we were able to discover information related to the Email security software in use by the enterprise. Take this line for instance, "`X-Ironport-AV: E=Sophos; i=""3.56,87,1900""; d=""scan'214,228""; a=""76:rtetl4724""`" (slightly adapted for privacy reasons). Here, we can identify that Cisco IronPort email and Web security gateway [5] has been deployed in the enterprise, and that the Sophos anti-virus engine is scanning emails. Moreover, the tag includes the version number of the gateway (via `i=3.56`), amongst other details (e.g., file information using `d=scan'214,228`); Cisco [6] provides additional details on the parameters of X-Ironport-AV. Other security software identifiable in the emails gathered include ForeFront (a Microsoft security product), StarScan (a security application from Willow Starcom), and references to Distributed Checksum Clearing (DCC) systems (commonly used for email spam filtering).

Another way in which security mechanisms manifested themselves was in the outsourcing of security functions such as spam filtering and virus checking. For example, in the header line, "`Received: from mailserver2.organisati.on (XXX.XXX.XXX.XXX) by XXX.messagelabs.com with AES128-SHA encrypted SMTP`", the sender's organisation is redirecting its email to messagelabs.com, which is an email and end-point security service offered by Symantec. Microsoft and Google also offer similar email services, and those were seen in email forwards to `XXX.exchangelabs.com` and `XXX.postini.com` respectively. At a general level, we noticed outsourcing of email services in 25% of emails; this spanned both outsourcing for security and general email service hosting and management. There was no real variation to report in the exposure of identity information across the five email types.

The main concern in exposing such security software details or that email services are outsourced, would be an attacker using this insight to search for known vulnerabilities which might then be exploited. Similar to the email software example from the previous paragraph, this could be as straightforward as searching online for an exploit to the software's specific version, or using all of the information gathered as the basis for a social-engineering attack to gather even more insight (e.g., posing as a Microsoft Exchange Labs employee to the organisation to discover more about their systems).

4.4 Identifying the Technologies of Email Senders based on Basic Header Tags

In addition to examining email headers for obvious leakage of potentially sensitive information, we also were interested in investigating header tags for any correlation with particular technologies. Unlike the two sections above, this assessment was more focused on the presence and absence of tags themselves (rather than on their values) and what that might leak about the technology being used by the message sender. For instance, if an individual is emailing from AppleMail (on an Apple Mac), are there certain header tags that are more likely to be present than if they were emailing from Outlook? The main difference to the work above is that we are not relying only on explicit mentions of technologies to identify what those technologies are.

For this assessment, we adopted a simple approach where we divided the set of emails into categories according to the technology used by the email sender. We focused on two areas, desktop email clients and emails from smartphones. Next, we grouped the emails from the same clients together, and those from the same phones together. There were three desktop clients, namely Microsoft Outlook (Windows/Mac), AppleMail and Thunderbird, and four types of smartphones, i.e., iPhone, Android-based, Blackberry and Windows-based (for smartphones, emails were predominately sent from the native email apps). After creating the groups, we then analysed the presence and absence of all header tags across the emails and noted the percentage of emails within each group with the tag present. To ensure that the tags were generated by the clients and not the mail servers, we considered and removed that tags that were present when the email was sent from the participant's respective Web-mail interface.

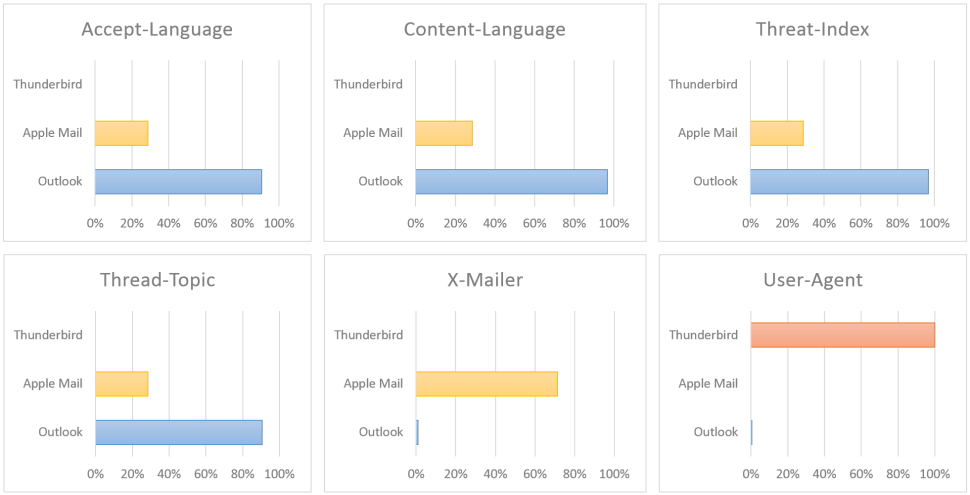


Figure 3: Examining the presence of six header tags according to the email client technology used

From the analysis of percentages, there were several intriguing findings. To start with the email clients, it was found that the there was a marked difference in the presence of some tags as highlighted in Figure 3. For instance, *Accept-Language*, *Content-Language*, *Thread-Index* and *Thread-Topic* were much more likely to be present in emails sent from Outlook than those sent from AppleMail or Thunderbird. In the case of Outlook and Thunderbird, 97% of emails sent from Outlook had a *Content-Language* header tag while 0% from Thunderbird included the tag. AppleMail could be distinguished from Outlook and Thunderbird through its persistent inclusion of the *X-Mailer* tag; this tag was found in 71% of AppleMail emails but only in 1% and 0% of Outlook and Thunderbird emails respectively. With respect to Thunderbird, the main differentiator between it and the other clients was the *User-Agent* tag. This was present in all of the emails sent from Thunderbird but just 1% of emails from Outlook and none of the emails from the AppleMail client. From this high-level assessment, we can already conclude that it

is possible to make intelligent assertions about the email clients in use by senders using the most basic of tag information. To properly examine this further however, and know whether it is truly feasible, one would need a large amount of header data and correlational detail.

Distinguishing smartphones based on basic tags was a considerably more challenging task. Although there were four types to assess, the Blackberry and Windows-based devices were poorly represented (only 5 emails in total) which led to their data not being substantial enough to thoroughly compare with iPhone and Android. In terms of the iPhone and Android comparison, the main finding as shown in Figure 4, was that iPhone tended to use *X-Mailer*, *Content-ID* and *Content-Transfer-Encoding* noticeably more. For example, *Content-Transfer-Encoding* was present in 100% of iPhone emails but just 64% of Android emails. While this is a useful finding, 64% is still a noteworthy amount and given that our sample size is not that large, this finding should be relied upon with caution. Such caution should also be applied to the other two header tags. To comment briefly on emails sent from Blackberry devices, one conclusion, albeit insignificant for reasons mentioned prior, is that these devices appear to always have tags for *Accept-Language*, *Content-Language*, *Content-Transfer-Encoding*, *Thread-Index* and *Thread-Topic*. Together, therefore, these tags could potentially be used to distinguish Blackberry devices from other types of devices; this, of course, assumes that our initial observation holds true for all of these smartphones.

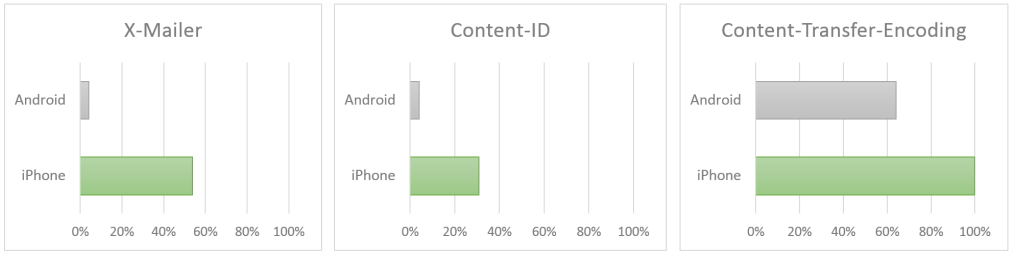


Figure 4: Examining the presence of three header tags according to the type of smartphone used

In addition to the findings above, there were a few other points worthy of attention. The first is the fact that the tag-value combination "`Content-Type: text/plain; charset="us-ascii"`" seems to mostly be present in emails sent from an iPhone (default client) or AppleMail client; in 70% of emails with that tag, this was the case. *Content-Type* was also useful in identifying other users. For instance, we found that the tag "`Content-Type: text/plain; charset="ISO-8859-1"; format=flowed`" only occurred with Thunderbird users on Linux. Although not as telling, we were also able to identify situations were systems were not in use. Two cases of such are demonstrated with the header tag-value combinations: "`Content-Transfer-Encoding: 7bit`" and "`Content-Transfer-Encoding: quoted-printable`". In the former case, we found that this combination only appeared to be included in emails not originating from the Windows operating system. In the latter case, none of the emails with this tag were from an Android-based device. Of course, there needs to be some caution with relying on these "not in use" results in particular given that the size of our sample could have affected our coverage of these individual groups.

To briefly summarise the findings in this section, we have demonstrated that it is possible, to some extent, to characterise types of technologies used (or not used) by email senders based only on basic header information. Although our results are mainly exploratory at this stage and by no means definitive, they shed further light on the real issues associated with inadvertent information leakage. If a determined attacker were able to gather enough information from a wide variety of clients and email systems, techniques such as supervised machine learning could easily be applied to mine for useful correlations. From this, a predictive model could be developed to determine the systems and software in use by an

email sender based on just basic header information and no explicit leakage of technology names. This knowledge could be used in combination with the techniques in previous sections to gather insight as a platform for an attack.

5 Reflection and Recommendations

Reflecting on the goal of this research and the experiment's findings discussed above, we can clearly see evidence of potentially sensitive information being exposed in email headers. This includes information about the individual (e.g., usernames, devices used, and ISPs) and the organisation that they work for (e.g., email server details, email software used, and to some extent, internal network configurations).

To comment on the overall experiment, there was not as much exposure across the participants as we had imagined or was found in our exploratory pilot studies. For example, in only 28% of emails could we explicitly (via information in tags) identify the email client used, and in merely 14% of emails were we able to discover a username. One reason for this could be the fact that our sample was not sufficiently diverse, and that the (good) practices of the organisations studied influenced our findings. There is, of course, also the possibility that leakage is not widespread and only happens in a few situations – the best-case scenario. Either way, we were able to infer pieces of sensitive information about individuals and organisations in our study that could be used as a platform for further attacks. This, therefore, does highlight some worrying level of exposure, albeit not tremendous.

Critically speaking, the real concern with the leakage found is the fact that a noteworthy portion of this information, especially information about a company's internal systems, is otherwise difficult for an attacker to obtain. Email headers, therefore, provide a valuable reconnaissance technique and only require a single, unsuspecting individual within a company to send an email to the attacker. The information gathered could then be used to research and target exploits in the company's software or used as a basis for a social-engineering attack to gather more actionable intelligence about the individual or enterprise.

In terms of protective strategies against such leakages, we strongly recommend that enterprises and individuals: (i) conduct an audit of what they may be exposing via their email headers; and (ii) if possible, aim to remove or redact any information that might be sensitive or used for nefarious purposes. In this paper we have provided a detailed example of the types of information that could result in an increased exposure to risk, both to privacy and to security. Another, more drastic option to avoid the leakage of information is to synthesize email headers. Therefore, instead of exposing sensitive details, misleading information could be included in sent emails. This option may be appealing to certain high-security companies or industries such as defence. One significant issue that protective approaches which seek to remove or synthesize headers will need to tackle is that of Bring-Your-Own-Device (BYOD) setups. This could require changes in enterprise policy or deployment of specialised device clients (or extensions) that can ensure that all possible avenues of information leakage have been addressed.

6 Conclusion and Future Work

The dominant use of emails as a form of communication for companies and organisations has opened the door to a range of potential attacks. Although much effort thus far has been dedicated to preventing attacks originating from incoming emails, there is a significant security problem regarding the disclosure of potentially sensitive information in email headers. The research in this paper specifically concentrates on the information that can be leaked by email headers to detect the extent that it occurs, where and when it tends to happen, and how it can affect the security and privacy of an individual or an organisation.

As our study has demonstrated, there exists a real and present risk of information disclosure within email headers. Although this inadvertent leakage was not as significant as we had worried might be the case, the information discovered could still be used for a number of malevolent purposes by an attacker. These range from intelligence-gathering activities as a preparation for large-scale attacks on an enterprise, to social-engineering attacks focused on more easy and quick targets. Our recommendation to enterprises and individuals to preserve their privacy and reduce risk exposure is to remove overly verbose tags such as *X-Mailer* and *User-Agent* and redact any technology-specific information not required by transmitting servers or the message receiver.

There are two main avenues that we are exploring for future work. The first is a broader study involving a larger participant cohort and a range of industries (security-focused and otherwise). This would serve to confirm the findings in this study and also to identify any additional information leaks not encountered previously. We would be especially interested using this larger dataset to look for correlations in email clients or devices, and the presence and absence of certain header tags. The second area that we aim to engage in future work on is that of tools to understand where information leaks occur. Specifically, we will aim to develop a set of plug-ins for email systems that would allow individuals to assess whether they expose any information, and if they do, the types of attacks that might be launched against them. We are also considering ways in which information can be safely redacted without affecting performance, and how this can be integrated into existing email systems. Addresses these areas of future work will be essential in ensuring the privacy and security of the systems and data of enterprises and individuals.

References

[1] M. Al-Zarouni. Tracing e-mail headers, 2004. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.67.6733 (Accessed 12/11/2014).

[2] BBC. Target data theft affected 70 million customers, 2014. http://www.bbc.co.uk/news/technology-25681013 (Accessed 29/10/2014).

[3] Carnegie Mellon University – Computer Emergency Response Team (CERT). Melissa Macro Virus, 1999. http://www.cert.org/historical/advisories/CA-1999-04.cfm (Accessed 12/11/2014).

[4] P.-A. Chirita, J. Diederich, and W. Nejdl. MailRank: using ranking for spam detection. In *Proc. of the 14th ACM international conference on Information and knowledge management (CIKM'05), Bremen, Germany*, pages 373–380. ACM, October-November 2005.

[5] Cisco Systems, Inc. Cisco IronPort, 2007. http://www.cisco.com/web/about/ac49/ac0/ac1/ac259/ironport.html (Accessed 12/11/2014).

[6] Cisco Systems, Inc. How do I decode the X-IronPort-AV header on the ESA?, 2014. http://www.cisco.com/c/en/us/support/docs/security/email-security-appliance/117887-qanda-esa-00.html (Accessed 12/11/2014).

[7] O. de Vel, A. Anderson, M. Corney, and G. Mohay. Mining e-mail content for author identification forensics. *SIGMOD Rec.*, 30(4):55–64, 2001.

[8] P. Eckersley. How unique is your web browser? In *Proc. of the 10th international conference on Privacy enhancing technologies (PETS'10), Berlin, Germany, LNCS*, volume 6205, pages 1–18. Springer-Verlag Berlin, July 2010.

[9] Exim. Exim Security, 2014. https://github.com/Exim/exim/wiki/EximSecurity (Accessed 13/11/2014).

[10] Exim. Exim Internet Mailer, n.d. http://www.exim.org/ (Accessed 12/11/2014).

[11] Forbes. Data Breach Bulletin: Dairy Queen, JP Morgan Chase, AT&T, 2014. http://www.forbes.com/sites/katevinton/2014/10/10/data-breach-bulletin-dairy-queen-jp-morgan-chase-att/ (Accessed 12/11/2014).

[12] InfoSec Institute. Fun with email headers, 2013. http://resources.infosecinstitute.com/fun-with-email-headers/ (Accessed 12/11/2014).

[13] Kaspersky Lab. Social Engineering, Hacking The Human OS, 2014. http://blog.kaspersky.com/social-engineering-hacking-the-human-os/ (Accessed 12/11/2014).

[14] J. Klensin. Simple Mail Transfer Protocol. IETF RFC 5321, October 2008. http://www.ietf.org/rfc/rfc5321.txt.

[15] G. Klyne, M. Nottingham, and J. C. Mogul. Registration Procedures for Message Header Fields. IETF RFC 3864, September 2004. http://www.ietf.org/rfc/rfc3864.txt.

[16] Microsoft Corp. View e-mail message headers, n.d. https://support.office.com/en-us/article/View-e-mail-message-headers-cd039382-dc6e-4264-ac74-c048563d212c (Accessed 12/11/2014).

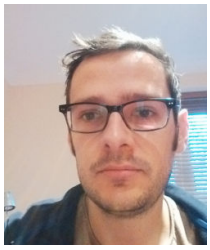
[17] Mozilla. Security Advisories for Thunderbird, 2014. https://www.mozilla.org/security/known-vulnerabilities/thunderbird/ (Accessed 13/11/2014).

[18] P. Resnick. Internet Message Format. IETF RFC 5322, October 2008. http://www.ietf.org/rfc/rfc5322.txt.

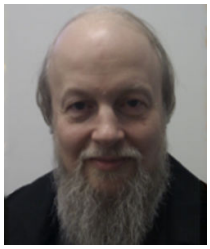
[19] The Wire. What your email metadata told the NSA about you, 2013. http://www.thewire.com/technology/2013/06/email-metadata-nsa/66657/ (Accessed 12/11/2014).

Author Biography

Jason R.C. Nurse is researcher in the Cyber Security Centre at the University of Oxford. He received his B.Sc. in Computer Science and Accounting (UWI, Barbados – 2001), M.Sc. in Internet Computing (Hull, UK – 2006), and Ph.D. degree in Computer Science specialising inWeb Services Security and e-Business (Warwick, UK – 2010). He has worked within industry and academia throughout his career. This has included various IT roles within industry, and academic posts such as Research Fellow at Warwick University, and more recently, Researcher at Oxford. Jason has published several articles at both journal and conference levels and also sits on the programme committee of related venues. His research interests include assessing the risks to identity security and privacy in cyberspace, information security and trust, human factors of security, insider threats, and services security.



Arnau Erola is a post doctoral researcher at the Cyber Security Centre at the University of Oxford. His research interests are related but not limited to cryptography, semantics, security and privacy. In 2006 he received a B.Sc. in Computer Science from the Rovira i Virgili University of Tarragona (URV), Spain. After that he worked as software developer for two years. He obtained a Ms.C. in Computer Security and Computer Science in 2009 and a Ph.D. in Computer Science in 2013, both from the URV. He is author of several international journal articles on online privacy and anonymity protocols.



Michael Goldsmith is a Senior Research Fellow at the Department of Computer Science and Worcester College, Oxford. With a background in Formal Methods and Concurrency Theory, Goldsmith was one of the pioneers of automated cryptoprotocol analysis. He has led research on a range of Technology Strategy Board and industrial or government-funded projects ranging from highly mathematical semantic models to multidisciplinary research at the social-technical interface. He is an Associate Director of the Cyber Security Centre, Co-Director of Oxford's Centre for Doctoral Training in Cybersecurity and one of the leaders of the Global Centre for Cyber Security Capacity-Building hosted at the Oxford Martin School, where he is an Oxford Martin Fellow.

Sadie Creese is Professor of Cybersecurity in the Department of Computer Science at the University of Oxford. She is Director of Oxford's Cyber Security Centre, Director of the Global Centre for Cyber Security Capacity Building at the Oxford Martin School, and a co-Director of the Institute for the Future of Computing at the Oxford Martin School. Her research experience spans time in academia, industry and government. She is engaged in a broad portfolio of cyber security research spanning situational awareness, visual analytics, risk propagation and communication, threat modelling and detection, network defence, dependability and resilience, and formal analysis. She has numerous research collaborations with other disciplines and has been leading inter-disciplinary research projects since 2003. Prior to joining Oxford in October 2011 Creese was Professor and Director of e-Security at the University of Warwick's International Digital Laboratory. Creese joined Warwick in 2007 from QinetiQ where she most recently served as Director of Strategic Programmes for QinetiQ's Trusted Information Management Division.