X.509 Forensics: Detecting and Localising the SSL/TLS Men-in-the-Middle

Ralph Holz, Thomas Riedmaier, Nils Kammenhuber, Georg Carle

Network Architectures and Services
Fakultät für Informatik
Technische Universität München
{holz,riedmaie,kammenhuber,carle}@net.in.tum.de

Abstract Although recent compromises and admissions have given new credibility to claimed encounters of Man-in-the-middle (MitM) attacks on SSL/TLS, very little proof exists in the public realm. In this paper, we report on the development and deployment of Crossbear, a tool to detect MitM attacks on SSL/TLS and localise their position in the network with a fair degree of confidence. MitM attacks are detected using a notary approach. For the localisation, we use a large number of traceroutes, conducted from so-called hunters from many positions on the Internet. Crossbear collects this data, orchestrates the hunting from a central point and provides the data for analysis. We outline the design of Crossbear and analyse the degree of effectivity that Crossbear achieves against attackers of different kinds and strengths. We also explain how analysis can make use of out-of-band sources like lookups of Autonomous Systems and geo-IP-mapping. Crossbear is already available, and 150 hunters have been deployed on the global PlanetLab testbed.

Keywords: Man-in-the-middle attack, detection, localisation, X.509, SSL/TLS

1 Introduction

The Secure Socket Layer/Transport Layer Security protocol suite (SSL/TLS) is commonly used on the Internet, and especially the WWW, to provide confidentiality, authentication and data integrity. A key feature is its use of the X.509 PKI to address the key distribution problem. In X.509, Certification Authorities (CAs) issue certificates to entities, with each certificate asserting a binding of entity name (e. g., a WWW domain) and the corresponding public key. The X.509 PKI forms a hierarchy where CAs at the root may issue certificates directly to an entity or delegate this process to (one or more) subordinate CAs. The result is a chain of certificates. Verifiers must trust the CAs at the root and, transitively, the subordinate CAs along the chain to verify a end-host certificate. Thus, Web browsers commonly ship with a list of root CAs deemed trustworthy (the 'root store'). A remarkable property of X.509 implementations in browsers is that all CAs in the root store *and* thus also all subordinate CAs are equally

capable of issuing certificates to *any* domain. This was always perceived to be problematic as it reduces the strength of the whole PKI to the weakest CA. An attacker with control over just one CA is able to stage MitM attacks against any domain. This happened to the DigiNotar CA in 2011 when an attacker was able to issue more than 500 forged certificates [11]. As the revocation infrastructure was also deemed compromised, all major browsers reacted by blacklisting forged certificates directly in the browser. The forged certificates were allegedly used in a Man-in-the-middle attack (MitM) staged against citizens of Iran.

It is this latter kind of attack that this paper is concerned with. While we may suspect from [11] that a MitM attack happened, and may speculate who the victims were, it remains curiously unknown how many MitM attacks really happen in the wild. Most reports seem to exist only in the form of blog posts or forum entries, e.g., [6,4,3], with claims ranging from attackers in hotel networks to state-level attacks against citizens of a country. In these cases, all attacks were actually easily detectable because the MitM did not bother to forge certificates but used invalid certificates and relied on users to ignore browser warnings. Unfortunately, affected users seem unlikely to store the MitM's certificate; nor do they record how they have connected to the Internet or to which WWW server (only [4] provides a copy of the fake certificate). Without proper evidence, however, we as a security community cannot know how pressing the problem of MitM attackers really is.

Our tool, Crossbear, has been developed as a response to this lack of hard data. It aims to make a first step towards gathering data and providing proof of the existence of MitM attacks. With the on-going deployment of Crossbear, we invite the interested community in our quest to give answers to questions like how many SSL/TLS MitM attackers exist on the Internet, which certificates do they use, and where are they located in the network. We are fully aware that, in particular, localisation is difficult to perform in the face of an adaptive attacker attempting to counter our methods; however, we are certainly going to raise the bars for evil-doers and hard evidence will likely help to increase public awareness. We also emphasise that Crossbear is intended as a tool for the savvy user or travelling hacktivist who wishes to contribute in the investigation of this important attack on one of the backbone protocols of the Internet, and not as a reinforcement or replacement for the current PKI (like, e.g., Perspectives and Convergence, see Section 2).

Crossbear builds on the well-understood notary concept, but with a twist: it employs a large number of so-called *hunters* distributed over the Internet that compare certificates they receive in a SSL/TLS handshake and record the IP route they have to that SSL/TLS server. This is reported to a central server where certificates and routes are analysed and further hunting initiated, i.e., more hunters asked to connect to the potential victim server and to report certificates and IP routes. A comparison of the IP routes from hunters that are affected by the MitM and by those who are not yields an approximation of the MitM's location in the network. The accuracy increases with the number of hunters.

Contributions and organisation The remainder of this paper is organised as follows. Section 2 presents related work and positions Crossbear as a tool to combine detection, localisation, and reporting. Section 3 outlines the design of the Crossbear ecosystem and highlights relevant design decisions. Section 4 analyses to which degree Crossbear can be an effective tool against different kinds of attackers of varying strength. It also gives an estimate of the needed numbers of hunters and shows how out-of-band information can help where pure tracerouting will fail. We conclude with a summary and invitation to participate.

2 Background and related work

The weaknesses of the X.509 PKI for SSL/TLS have been described in several research papers as well as at hacking symposia. Vratonjic et al. presented a study of the Alexa Top 1 Million list [20]. Holz et al. presented an extensive study [9] that covers 1.5 years and includes observation points from around the globe, as well as data from traffic monitoring. Eckersley and Burns presented a survey based on scans of the IPv4 space [12]. These efforts showed that certification practices are not very stringent at best, and authentication errors common. As Sunshine et al. [17] found that users are likely unable to decide whether a browser warning indicates a threat or can be safely ignored, this constitutes a serious weakness. Regardless of user abilities, Soghoian and Stamm warned that governments can compel CAs to issue forged certificates for state-level MitM attacks [15]. In such a case, browsers would not even show a warning.

The number of proposals to strengthen or replace the X.509 PKI suggests how little confidence is placed in it. Two replacements, EFF's Sovereign Keys [5] and Google's Certificate Transparency [10], are based on public logs, i.e., public append-only timelines with certificate information. Both are still in the design phase. It is unclear if they will be successful. A different idea is to employ a notary approach, which is also a key idea in Crossbear. This concept is based on the observation that a MitM is unlikely to control all network paths between a server and its clients. Thus, SSL/TLS clients can ask third-party observers (the notaries) whether they observe the same certificate from a given server. As long as the route to at least one notary remains outside of the attacker's control, this results in a mismatch between the certificates reported by the notaries and the one observed by the client.

To our knowledge, Perspectives [21] was the first notary-based project. The idea is to make initial contact to an unknown server robust against a MitM. The project periodically scans WWW hosts to generate a database of public keys. Browsers with the Perspectives add-on can compare keys from the database with those they are observing. Since 2011, Convergence [19] provides a similar service on the basis of observed certificates. The Convergence notaries do not conduct pro-active scanning and connect to a WWW server only when a client reports a yet unknown certificate. The browser-side add-on relies entirely on the notaries and essentially disables the use of the normal X.509 PKI. Convergence emphasises so-called 'trust-agility': users can choose to use different notaries

when their current ones have been compromised. This, however, requires that users understand the involved technologies and consequences.

All notary concepts share the problem of lack of privacy: notary operators know which sites users access. Convergence employs a kind of onion-routing to mediate this. Crossbear does not address privacy issues: as its purpose is to collect and report data about real attacks, privacy could not be a design goal.

3 Crossbear design and ecosystem

Crossbear's purpose is the collection of data that is likely to contain proof of a MitM attack. One particular working hypothesis that we would like to see either verified or falsified is that there are primarily two kinds of attackers. The first kind consists of MitM attacks close to the victim client, e.g., on wireless access points. The other kind are 'state-level' attackers, i.e., such attackers that can control whole ISPs and plant the MitM software on network border routers with the goal of monitoring all SSL/TLS traffic from within their country to one or several external services (e.g., Web mailers or social networks).

3.1 Principle of operation

Key idea Crossbear deploys a large number of so-called hunters on the Internet, distributed over as many Autonomous Systems and networks as possible. We implemented two kinds of hunters. The first is an add-on for the Mozilla Firefox Web browser; the other is a stand-alone application. The add-on is used for both detection and localisation, the stand-alone applications only for localisation.

The Crossbear server holds a list of servers that are reportedly attacked by a MitM. This list is pulled at regular intervals by the hunters, which will then connect via SSL/TLS to the reportedly attacked servers. They extract the certificate chain the server sends and record the IP route to the server by doing a traceroute. This information is then sent to back to the central server, where it can be analysed.

Detection Possibly attacked servers are reported automatically with the help of the Firefox add-on. We elaborate on this in Section 3.2. Naturally, a user of the browser add-on is warned if an ongoing MitM is detected.

Localisation The position of the attacker can be approximated by *cross-bearing*, i.e., comparing the routes that hunters recorded and determining the intersection points for routes that have been found to be poisoned and those that have been found to be clean. This is illustrated in Fig. 1, where the Crossbear server receives different certificates and traceroutes from the victim Alice and from the hunters Bob, Charlie, and Dave. This allows it to guess that the attacker is located in the vicinity of Alice because the intersection between her traceroute and Bob's traceroute is already at router R4, and Bob reports a clean connection. We discuss the effectivity of cross-bearing against attackers of various strengths and positions in Section 4.

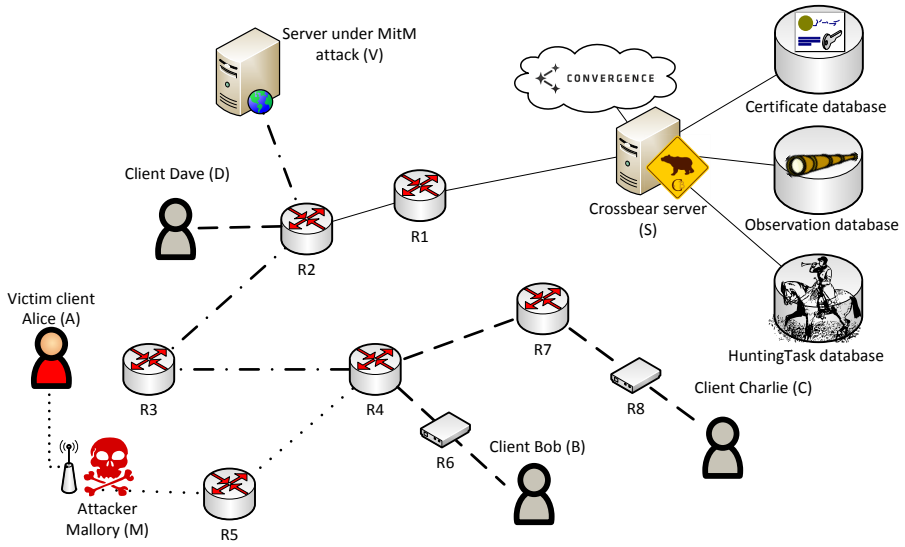

Figure 1: Components of the Crossbear system.

Further vantage points The Crossbear server uses Convergence [19] as a source of independent observations from other vantage points on the Internet. It queries Convergence notaries and stores certificate information plus additional independent temporal information, e.g., for how long a given certificate has been observed.

Out-of-band information sources We also store additional information we obtain from sources other than hunters.

- CAs used: We store which CAs a domain uses. Domains like, e.g., Google have always remained customers of the same CAs for longer periods of time[1].
- WHOIS information: We retrieve the AS number of all hosts in a traceroute. For a reported MitM attack, we also take into account how many reports from the ASes in the country in question have reached us, and whether the reported forged certificates share properties. The latter is motivated by the observation that an attacker is less likely to compel or compromise more than one CA.
- Geo-IP-mapping: Hosts in a traceroute are also looked up in geo-IP databases. Although imperfect, this allows to guess which countries SSL/TLS traffic has traversed.

We elaborate on the use of this information in Section 4.

[1] For Google, this has been confirmed to the authors in private e-mail.

3.2 Details of the detection process

MitM attacks are detected with the add-on for the Web browser.

Protecting the communication with the server All Crossbear clients (add-ons and hunters) communicate with the Crossbear server via TLS. To protect this channel against MitM attacks, the server certificate is hard-coded into clients. If a client finds that the received server certificate does not match the hard-coded one, its current behaviour is to refuse to operate and offer the user to send an automatic mail or fax to the Crossbear team that contains all details about the incident (including the forged certificate). While not yet implemented, the next major version of Crossbear may support the server signing its messages. This would allow to bypass the attacker even over a poisoned connection. Note that clients never sign messages: they do not have IDs and can thus not be authenticated.

Certificate verification Fig. 2 shows how certificate verification works. If user Alice (A) connects to a Web server V via SSL/TLS and the connection is under attack by a MitM, she will receive a forged certificate. A thus always sends a `CertVerifyRequest` to the Crossbear server S. The message includes the observed certificate chain, V's domain name and A's IP address. S stores this together with a timestamp. It then connects to V itself and stores the corresponding data. S also queries Convergence notaries for known certificates for V and stores the results with the observation period that Convergence reports. This result is sent to A (`CertVerifyResult` message). There are three optional messages. If the certificate comparison suggests a MitM, S includes a hunting task for A, i. e., a request to conduct a traceroute to V. S also includes a reference timestamp and a `PublicIPNotification`. We explain the latter in Section 3.3.

Score over certificate properties The Crossbear server also computes a score that is reported to client add-ons. The score is a weighted sum over a number of properties. The motivation here is to reduce the number of false positive warnings for human users of the add-on, i. e., occasions where S detects a certificate mismatch, but V's certificate is actually valid. This can occur for server farms with multiple different certificates deployed, or for sites that change their certificates very frequently.

The primary criterion in the score is the comparison of the certificates that client and S have encountered. However, S also takes the *last continuous observation period* (LCOP) into account, which expresses for how long *only* the certificate in question has been observed. Further criteria are the number of previous observations and the certificates that Convergence reports (together with their LCOP). Our weights are chosen thus that 'critical' combinations of properties yield a score of less than 100. When a score is below this (user-adjustable) threshold, our add-on displays a warning. Where several factors (completely or almost) counter-balance a certificate mismatch, the warning will either be

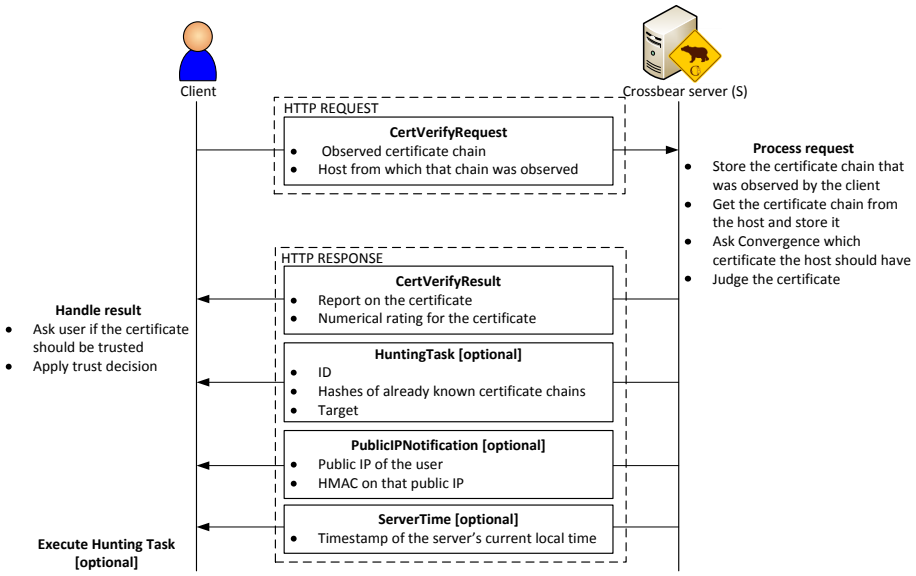

Figure 2: Protocol flow including the certificate verification request and hunting tasks.

suppressed or a user is at least given the server's score indicating the factors that make the certificate likely valid (recall that Crossbear is intended for savvy users). We list the most relevant factors in Table 1.

When the certificate score is above the threshold, the browser add-on caches the observed (host, certificate) combinations. When the threshold is not reached and a warning is displayed, the user is asked if the combination should be exempted and cached. We have experimented with the default settings over the course of several months and found that false positives occur only rarely. For popular sites that use several certificates (like Facebook), they have become rare (due to the server observations) and are not annoying due to the caching. They can still happen in the small time window when a site that frequently changes its certificates (like Google) does so and neither hunters nor Convergence have yet observed this. The more popular a site is, the less frequently this happens.

3.3 Details of the hunting process

Hunting is the process of determining a suspected attacker's network location, i.e., his position in an AS, sub-network or (with the help of Geo-IP databases) approximate geographic position.

Every hunter pulls the list of active hunting tasks from the Crossbear server at regular intervals. A hunting task can also be sent to a client together with a certificate judgement indicating a possible MitM attack. A `ServerTime` message

Property and score	**Rationale**
Certificate comparison:	
80 if $C_c = C_s$	S observes same certificate
0 if $C_c \neq C_s$	Potential MitM
-100 if S cannot get certificate from V	S likely blocked
LCOP:	
$\frac{days \cdot 2}{3}$ if LCOP ongoing	C_c still observed
$\frac{days}{3}$ if LCOP ended in the past	C_c observed in the past only
Observations: $\frac{count}{30}$	Number of observations
Convergence:	
$\frac{days \cdot 2}{3}$ if certificates match	Confirmation
$\frac{days}{3}$ matched in past, but not now	Outdated confirmation
-20 if never observed	Weak indication of MitM
0 if no reply from Convergence	Inconclusive

Table 1: Parameters used in computing a score for a reported certificate. C_c is the certificate observed by the client, C_s the certificate observed by the Crossbear server (S). V is the victim server.

is used to give a reference time to use when results are reported back to the Crossbear server. The hunter then starts to execute the tasks (see Fig. 3). The hunting starts with conducting a full SSL/TLS handshake and extracting the certificate chain. The next step is to record the route that IP packets take towards the destination. This is done with a standard ICMP traceroute. Certificate chain and route are sent to the Crossbear server.

We require hunters to send a `PublicIPNotification` request to the Crossbear server before they can conduct a traceroute and submit results. The reply to a `PublicIPNotification` contains the public IP address *that the Crossbear server observes* plus a HMAC of it, keyed with a secret key that only the Crossbear server knows and that is replaced every 30 minutes. Hunters must prepend their thus determined public IP address to the traceroute and include the HMAC. This serves several purposes. First, it is necessary to inform hunters with multiple IP addresses (e.g., IPv4 and IPv6), or which are located behind a NAT, of the IP address to conduct the traceroute from. Second, it acts as a protection against completely deliberate forgery of traceroute results: it forces a hunter to be reachable from the IP address it claims to have. While a powerful attacker is still able to spoof IP addresses of the system it controls, or of attached systems, this prevents him from submitting results allegedly from networks that are not under his control. Finally, knowledge of the claimed source of the traceroute enables us to draw on publicly available BGP dumps (e.g., Route Views [1]). This can aid in testing the plausibility of a route during the analysis process. We discuss this in Section 4.

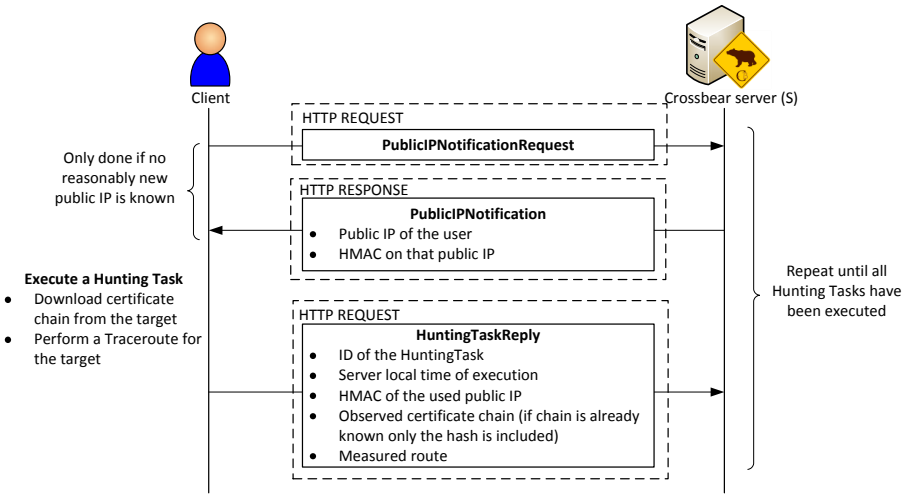

Figure 3: Crossbear's protocol for the execution of Hunting Tasks.

3.4 Status of deployment

Crossbear is available in version 1.5 [14]. Stand-alone hunters have been implemented and are already deployed on the PlanetLab testbed in 150 different locations on the globe. The Crossbear server is hosted at Technische Universität München. At the time of writing, Crossbear has just finished its beta phase; our database contains about 4,000 certificate observations conducted by our server plus another 2,000 retrieved from Convergence notaries. Results have been reported from more than 150 unique IP or /24-sub-networks. We have not found indications of MitM attacks so far, however. Current work includes providing the hunting functionality as a module for OONI, a distributed framework to monitor network interferences on the Internet [7].

4 Analysis and discussion of effectivity

We analyse the degree to which Crossbear can be an effective tool. We also discuss counter-attacks against Crossbear.

4.1 Attacker model

The attacker is assumed to have the full control over a 'system' on the path from the client to the victim server. A system can be either a router or an entire Autonomous System (AS) through which traffic is forwarded. The attacker does not control any other path in the network. In particular, he can only 'impersonate' IP addresses (i.e., spoof them *and* intercept replies addressed to them) from the system he controls or systems that are attached to it, and whose upstream

and downstream traffic is routed through it. An attacker controlling several systems can be modelled as separate attacks, with the addition that the attacker can use impersonated IP addresses from the other attacking systems as well.

We structure our discussion along two dimensions. Firstly, we distinguish attacker types by their selectivity against clients:

Non-selective attacker: The non-selective attacker stages his MitM attack against all clients attached to his system. There are two sub-types: attackers that MitM only the connections to some SSL/TLS server(s) and attackers that MitM every SSL/TLS connection.

Selective attacker: The selective attacker stages his MitM attack against a *sub-set* of clients attached to the system he controls. The same two sub-types as above exist.

Secondly, we distinguish by the position of the attacker in the network. We give special focus to locations that give an attacker most impact: towards the periphery of the Internet and close to the client; towards the periphery and close to the victim server; and in a central location of the network topology (i.e., an important router or well-connected transit AS).

This is motivated by the suspected kinds of attacks in the MitM reports like [6,4,3], which Crossbear was designed to address primarily. These attackers are depicted in Fig. 4a and 4b. The first kind is a non-selective attacker who operates close to the client, e.g., a poisoned wireless access point. The second kind is a much more powerful but also non-selective attacker who controls an entire system to which several sub-systems are attached. An example is a state-level attacker who stages a MitM attack against the population of his own country by controlling traffic passing his border system(s).

Fig. 4c and 4d show attackers that Crossbear (and indeed any tracing system) is less effective against. Fig. 4c is a selective attacker that is located close to the client but acts only against a sub-set of the clients attached to the system he controls. Fig. 4d is the most powerful and cunning attacker we consider: He is in control of an important system in the Internet core (e.g., an important transit AS) and stages his attack against just a sub-set of client systems at the periphery. A possible example is state-condoned industrial espionage where a government agency stages a MitM attack on traffic passing through their AS. Note that MitM attacks become more difficult the more the attacker moves towards the core of the network: the attacker needs to modify both directions of the traffic; but phenomena such as hot-potato routing [18] and BGP peering policies like valley-free routing [13] often cause IP packets to take different return paths.

We discuss now how effective Crossbear is for each scenario and which additional steps can be taken to aid detection and localisation.

4.2 Detection

In general, ongoing MitM attacks can be reliably detected by a Crossbear client/hunter because the queried Crossbear server observes a different certificate

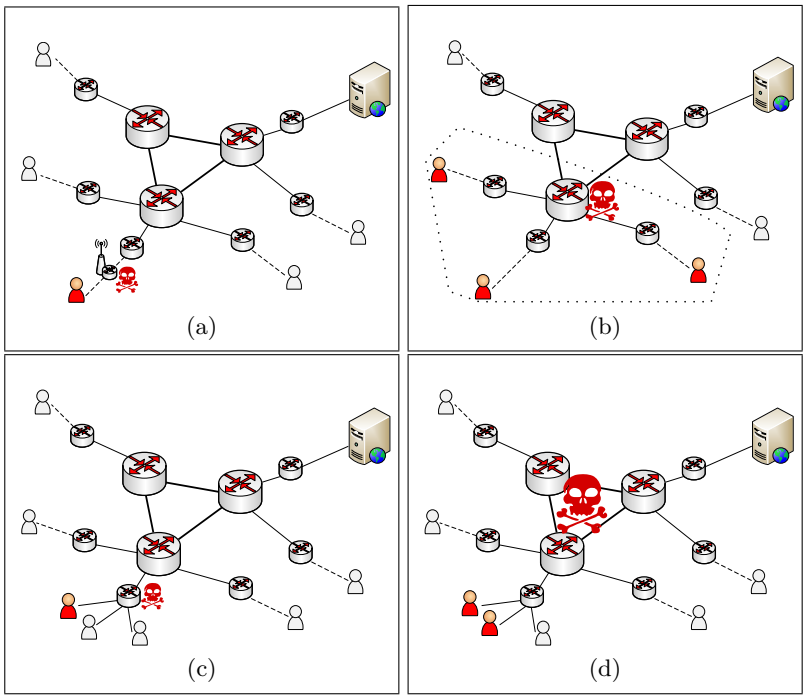

Figure 4: (a) Non-selective attacker in vicinity of client. (b) Non-selective state-level attacker. (c) Selective attacker in vicinity of client. (d) Selective (super-) attacker in core of network.

for the victim server. This is true for all attacker types in Fig. 4. Note that if the attacker chooses to MitM the connection to the Crossbear server, this is detected and the add-on will react to it (see Section 3.2).

The only attack that cannot be reliably detected by certificate comparison is when the attacker is on all paths from the vantage points to the victim server. This is a weakness all notary systems share. Such an attacker would either have to hijack BGP routes (as proposed in [8]) or position himself on a point in the network where all paths to the destination have already converged, i.e., close to the victim server. If the victim server has been observed previously, however, Crossbear can still profit from information available at the server. For example, when important certificate properties like the issuing CA change, this will flag a client report for manual verification. However, if the victim server has never been observed before, the attack is not detectable by any notary system.

4.3 Localisation

The ability to accurately trace the attacker's position in the network depends entirely on the attacker acting selectively or non-selectively.

The non-selective attacker The non-selective attacker lends itself well to localisation. In order for this to work, Crossbear needs a traceroute from the victim client and from at least one hunter that is attached to an upstream system (from the attacker's point of view) and which reports a clean connection. The accuracy increases the closer that upstream system is towards the attacker's own position and if that view is corroborated by other hunters either downstream (reporting poisoned connections) or upstream (reporting clean connections).

We present a rough estimate of how many hunters are required in order to locate a non-selective MitM attacker. To this end, we derive a closed-form model to estimate the average number of hunters needed to detect a MitM with a certain probability. We fully acknowledge that we require a number of simplifying assumptions to make the model suitable for analysis.

Our model only requires the distribution of path length between victim client and victim server and the distribution of node degrees as input. We have derived such topological data from publicly available router-level maps and from measurements from our own university network.

Our analysis is based on an observation that holds for most Internet traffic: Once two traffic flows with the same destination converge at a point in the network, they will not separate again until they reach their target. This is a characteristic of standard IP routing which is based on the destination but not on the source address. Exceptions exist (e. g., ECMP, CoS differentiation) but are rare; thus our model will hold for most cases. Given a path from victim client to victim server via an attacker, traceroutes from hunters will join the path at some point. Due to the genericity of our model, we can apply it at router level (i. e., to find the router conducting the MitM attack) as well as at an AS level (i. e., to find the AS conducting the attack).

In the following, we use the generic term *node* to denote a router or an AS.

Closed-form model for estimating the number of hunters We assume that initially there is only one victim client C and one victim server V. Let the path $C \leftrightsquigarrow V$ consist of intermediate nodes X_j, where X_2 is connected to $X_1 = V$ and X_ℓ is connected to C. Traffic $C \leftrightsquigarrow V$ is subject to a MitM attack at $M =: X_m$. After C suspects that its traffic exchanged with V is being attacked, hunters $H_1, \ldots, H_n$ conduct SSL/TLS handshakes and traceroutes to V.

In order to accurately locate the attacker at $X_m := M$, we require that traffic from some hunter H_1 joins the path $C \rightsquigarrow V$ exactly at X_m, and we need another hunter H_2 whose traffic joins the path on the last undisturbed hop X_{m-1}. See Fig. 5 for a graphical explanation. Calculating the probability for these placements yields the probability for attack localisation.

Our model makes the following assumptions: (1) The attacker intercepts the traffic exchanged between C and V at a single node M. (2) The MitM attacker M is not selective, i. e., all traffic for V passing through M is attacked. (3) For simplicity, we assume that the traffic path $V \leftrightsquigarrow C$ is symmetric, as well as any section $V \leftrightsquigarrow X_j$ of any path $V \leftrightsquigarrow X_k \leftrightsquigarrow H_i$. Note that real-word routing often results in asymmetric paths. However, our model remains usable as long as the path lengths do not differ significantly. (4) In all nodes along the path $C \leftrightsquigarrow V$,

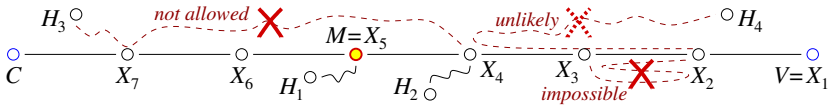

Figure 5: Notation for the model

traffic is routed purely according to the destination address (an exception is M, who may choose to divert the attacked traffic). This means that any traffic sent to V follows a tree with root V: once two traffic flows $H_1 \rightsquigarrow V, H_2 \rightsquigarrow V$ have converged at some intermediate node X, they will not separate until they reach V. The same considerations apply for the opposite traffic originated by V: once its flows separate, they cannot converge again. However, note that due to hot potato routing, assumption 4 may not hold on an AS level if the node under consideration is a large-scale AS and the ingress points of H_1's and H_2's traffic are very different, e. g., on different continents. (5) The probability for a node to be selected as the location for a hunter H_i is the same across all nodes.

Probability that a hunter covers a node We estimate the probability that traffic from a randomly placed hunter H_1 traverses a given intermediate node X_j. For simplicity, we only analyse the traffic direction $\{C, H_i\} \rightsquigarrow V$. Call X_{j-1} the *successor* of X_j. Assume we already know the probability $\Pr[X_{j-1}]$ that H_1's traffic passes successor X_{j-1}. We now make the further assumption that (6) the probability that traffic is forwarded to a specific neighbour of X_j (e. g., X_{j-1}) is evenly distributed among all neighbours of X_j. The only exception is its successor X_{j-2}, which cannot be chosen: any path $H_1 \rightsquigarrow X_{j-2} \to X_{j-1} \to X_{j-2}$ implies a routing loop (Fig. 5, line labelled 'impossible'). If X_j has d_j neighbours (i. e., has degree d_j), then the probability that H_1's traffic comes from X_j is $1/(d_j - 1)$. Hence the overall probability that H_1's traffic passes X_j is $\Pr[X_j] = \prod_{k=1}^{j} 1/(d_k - 1)$. Note that this assumption is actually overly conservative: In reality, certain neighbours can be ruled out due to hot potato routing, valley-free routing, topological position etc.; e. g., in Fig. 5 a direct path $X_2 \leftrightsquigarrow H_4$ is much more likely than the dotted path labelled 'unlikely'. Hence d_j is effectively reduced and the probability that H_1's traffic crosses X_j is thus higher than our estimate.

Probability for correct placement To locate the attacker at X_m, we need (*requirement I*) one hunter H_1 who also experiences the attack and whose traffic separates right at X_m, and (*requirement II*) another hunter H_2 whose traffic separates nearer to V at X_{m-1}, i. e., it just escapes the MitM attack. To meet requirement I, the traffic must not come via the predecessor X_{m+1}. Neither can it come from the successor X_{m-1} (routing loop argument). Under assumption (6), the probability that a hunter H_1 meets our requirement I is thus $\Pr[\text{req I}] := (d_m - 2)/(d_m - 1) \cdot \Pr[X_m]$. Similarly, the probability that a hunter H_2 meets our requirement II is $\Pr[\text{req II}] := (d_{m-1} - 2)/(d_{m-1} - 1) \cdot \Pr[X_{m-1}]$.

As the placement of n hunters can be viewed as a Bernoulli trial, the probability that the traffic of at least one of n hunters satisfies requirement II is $1-(1-\Pr[\text{req II}])^n$, and the probability that at least one of the remaining $(n-1)$ hunters satisfies requirement I is $1-(1-\Pr[\text{req I}])^{n-1}$. Hence the probability that both requirements are satisfied and that the attacker can be located at node X_m is $\Pr[\text{locate}(X_m)] := 1-(1-\Pr[\text{req II}])^n \cdot 1-(1-\Pr[\text{req I}])^n$. If we allow a defined uncertainty of M's position, we have to consider cases where H_1's and H_2's traffic flows separate one or two hops further towards C or V.

Arbitrary positions of C, V and M In reality, the hop distance ℓ between C and V is not fixed but can be seen as a discrete random variable that, on a realistic router-level graph, can assume integer values between 2 and about 30 (for AS graphs, this number is naturally smaller). Note that ℓ has not played a role in our calculations so far, as our probability is only affected by m, i.e., the number of hops between attacker and victim server. We can calculate an aggregate detection probability for the attacker by summing over all possible values of ℓ, and summing over all possible locations m of the attacker. Our final closed-form model is thus:

$$\Pr[\text{locate}] := \sum_{k=1}^{max\ path\ length} \left(\Pr[\ell = k] \cdot \sum_{m=1}^{k} \Pr[\text{locate}(X_m)] \right) \tag{1}$$

Topological data for the closed-form model To fill in the necessary distributions for the number of hops ℓ and the node degrees d_j, we collected data on both IP router as well as AS topologies. Rocketfuel [16] provides router-level maps gained from sophisticated measurements. As networks mainly change in size but not significantly in fundamental structure, we conjecture that these somewhat dated maps are still usable for our application. Since they do not reveal the positions of clients or servers, we only calculated an average degree $\bar{d}$ and use $d_j = \bar{d}$ in our model. Across all Rocketfuel topologies, the average degree is 3.98. We obtained the typical number of IP hops by issuing traceroutes from our university network to about 30,000 randomly chosen hosts from the Alexa list [2] of the top 1 million most popular Web sites. The distribution of the path lengths (range 5–28, mean 15.28, median 15 hops) loosely resembles a bell curve, suggesting that data collected from other vantage points will not be fundamentally different.

For constructing a (partial) AS graph, we used the RouteViews archive [1] and combined the 07/07/2011, 12:00h MRT-formatted full-table RIBs from Oregon IX, Equinix Ashburn, ISC/PAIX, KIXP, LINX, DIXIE/WIDE, RouteViews-4, Sydney, and São Paulo. From this graph, we determined an average degree of 3.51, as well as the distribution of path lengths (range 1–17, mean 3.25, median 3 hops). Again, the data does not reveal locations of clients and servers, so these calculations have to be taken with a grain of salt.

Results using the model Fig. 6 summarises the information we can gain from our model. The model suggests that we only need a very small number of hunters

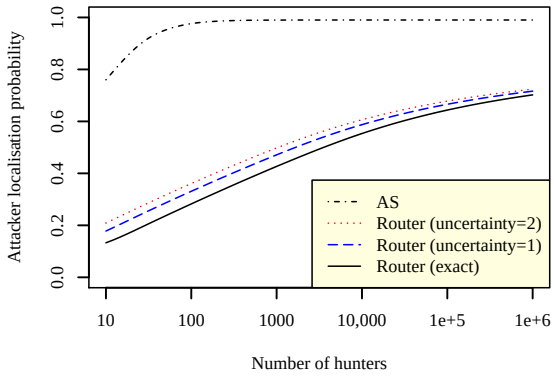

Figure 6: Estimating the number of hunters required to pinpoint an attacker.

to localise the AS in which an attacker resides (dash-dotted line). With as little as 100 hunters randomly distributed across the Internet, chances to pinpoint the attacker to an AS approach 100 %. At the router level, however, the picture looks different. The model suggests that just 10 hunters are needed to pinpoint the attacker to a single router (solid line) or to a set of two (dashed line) or three routers (dotted line) on the path to the attacked server with a probability between 10 % and 20 %. This probability rises to roughly 50 % if about 5,000 hunters can be employed. However, even with one million hunters, we only have about a 70 % chance to pinpoint a malicious router. Our conclusion here is that Crossbear works well in tracing an attacking AS, but much less so in tracing the exact router. However, this is quite acceptable: Crossbear works well against our state-level attacker from Fig. 4b. As for the attacker on the wireless access point, (Fig. 4a), successful localisation needs exact placement of hunters in the same ISP network anyway, rather than a large global number of hunters.

The challenge of selective attackers Selective attackers can neither be localised directly nor on-the-fly. Indeed, the possibility of selective attackers requires that every reported attack is carefully analysed manually.

Consider Fig. 4c and 4d: no hunter, not even downstream, experiences the attack. As far as tracerouting is concerned, these attackers become indistinguishable from the one in Fig. 4a. A major challenge thus lies in telling them apart. However, the attacker may still leave clues that, using the out-of-band information described in Section 3.1, point to the nature of the attack.

Assume that we are in possession of traceroutes from all clients that are affected by the MitM. Ideally, we also have traceroutes from seemingly non-affected hunters in the same AS, and ASes in the same country, and ASes that are attached to an AS that is further upstream. Recall that the attacker cannot deliberately forge traceroutes as the `PublicIPNotification` mechanism forces him to be able to intercept replies to his IP address. Thus, he can only choose his source IP from the system he controls or one that is attached to it. Also recall

that a traceroute can be tested for plausibility to some degree with available BGP data (e.g., [1]). The hints we are looking for are poisoned routes from different stub ASes, i.e., ASes on the network periphery. If we find such routes in our data, we can conjecture that the MitM is located either where traffic from these AS converges (the earliest possible location), or further upstream. The only plausible alternative would be to assume simultaneous attackers against multiple AS. This is possible, but one way to tell them apart is to investigate if the forged certificates share properties (like issuer, key lengths, X.509v3 extensions). If they do, this points to a common rule set for creation, thus two separate MitM attacks are less likely. The next step to execute is now to look up the AS and countries of all hops in the traceroutes. A hint that a selective state-level attacker is indeed at work is then if we find that the source IPs in the traceroutes belong to an AS/country which we associate with radical monitoring of their own population. If the earliest possible location is in that country, that is another hint.

If we do not find anything of the kind, however, our chances become slimmer. One pattern that is still worthwhile to look for is the one that the selective super-attacker in the core of the network (Fig. 4d) should show. If the purpose of the attacker is indeed industrial espionage, one may expect that the MitM reports and traceroutes are primarily from companies within a select few countries.

Naturally, all of the above is a mere test of plausibility, and we acknowledge that the proposed methods require (comparatively) intensive manual labour. However, we wish to point out that until now the research community has practically no data at all about MitM attacks occuring in the wild. Any report providing such data will advance current research. The MitM attack in [4], for example, became known thanks to external reports and because someone made the effort to try and inform the outside world. Receiving automated reports is thus useful even where automatic localisation is not possible. This is why we advertise Crossbear as a tool to record as much data as possible about attacks, but not as a silver bullet in exposing attackers.

4.4 Attacks against Crossbear

Due to Crossbear's open nature, there are several options for particularly aggressive attackers. Many of these cannot be entirely avoided and have to be dealt with in a reactive way.

Hunters do not need to register nor do they have IDs. This was a conscious choice to encourage user participation. As is true for all such systems, however, one consequence is that attackers can freely send forged data to the Crossbear server. Such injections are particularly hard to detect if the attacker employs 'malicious hunters'. Here, the attacker first drops the connections of all honest hunters in the system he controls or that are attached to it (note that this may lead to out-of-band reports). Then, his malicious hunters send forged reports stating that the connection via the attacker is fine and no MitM is detected. The Crossbear server will thus have received only one report of a possible MitM (from the client victim) and a large number of forged reports. The only defence that Crossbear has here is that the attacker's source IP is ascertained. As long

as the attacker is not in the core of the network, this will result in a suspicious cluster of reports from the same AS or country. The attacker can again offset this by renting and using other computers, e.g., in the cloud. This might be revealed by implausible traceroutes, but blacklisting such attackers becomes much harder (and effectively an arms race).

The Crossbear server is a single point of failure. The usual (pro-active and reactive) DoS defences on the IP level can be taken. However, attackers can inflict more serious damage with the above attack or by, e.g., flooding the server with alleged MitM reports (which lead to hunting processes being initiated). These attacks can only be detected by continuously monitoring requests and reports, with special focus on reports from recurring systems or countries.

5 Conclusion

We have described how our tool, Crossbear, can be employed to detect and localise MitM attacks on SSL/TLS, and we have analysed against which attacker types it is particularly effective.

Crossbear can reliably detect and report MitM attacks by most attacker types. Crossbear's effectivity in localising the attacker's position in the network depends strongly on the kind of attacker it faces. Best results can be expected against an attacker who stages a non-selective MitM attack, like attackers close to the victim client or state-level attackers monitoring all SSL/TLS traffic to some WWW servers. Selective attackers cannot be accurately localised. However, they do leave hints in the reported data that a careful analysis can use to reveal or assess the nature of the attack. We have also analysed active measures that an attacker can take against Crossbear. Like all open systems, Crossbear shows a certain vulnerability here. However, such counter attacks leave hints, too.

We thus advertise Crossbear as a tool to make a step forward in the reporting and possibly also in the localisation of MitM attacks in the wild, but we expressively do not market it as a silver bullet to expose all kinds of attackers. We wish to invite the research community to participate. Naturally, our data will be shared.

Acknowledgements We wish to thank Christian Grothoff and Johann Schlamp for their valuable input. We also wish to thank OONI, and in particular Jacob Appelbaum, for their offer to accept Crossbear as a module.

References

1. Advanced Network Technology Center, University of Oregon: Route views project. `http://www.routeviews.org/` (2012), [last retrieved in April 2012]
2. Alexa Internet Inc.: Top 1,000,000 sites (updated daily). `http://s3.amazonaws.com/alexa-static/top-1m.csv.zip` (2009–2011), [last retrieved in April 2012]

3. Borhani, A.: Is This MITM Attack to Gmail's SSL? Forum post: https://www.google.com/support/forum/p/gmail/thread?tid=2da6158b094b225a&hl=en (Aug 2011), [last retrieved in April 2012]
4. Eckersley, P.: A Syrian man-in-the-middle attack against Facebook. https://www.eff.org/deeplinks/2011/05/syrian-man-middle-against-facebook (May 2011), [last retrieved in April 2012]
5. Electronic Frontier Foundation: The Sovereign Keys project. https://www.eff.org/sovereign-keys (2011), [last retrieved in April 2012]
6. Engert, K.: Man-In-The-Middle experience in Warsaw. Blog entry: https://kuix.de/blog/comments.php?y=11&m=06&entry=entry110616-171707 (Jun 2011), [last retrieved in April 2012]
7. Filastò, A., Appelbaum, J.: OONI: Open observatory of network interference. In: Proc. 2nd USENIX Workshop on Free and Open Communications on the Internet (FOCI 2012) (Aug 2012)
8. Hepner, Clint and Earl Zmijewski: Defending against BGP man-in-the-middle attacks. Talk at BlackHat 2009. https://www.renesys.com/tech/presentations/pdf/blackhat-09.pdf (2009), [last retrieved in April 2012]
9. Holz, R., Braun, L., Kammenhuber, N., Carle, G.: The SSL landscape – a thorough analysis of the X.509 PKI using active and passive measurements. In: Proc. 11th Annual Internet Measurement Conference (IMC '11), Berlin, Germany. ACM, Sheridan (Nov 2011)
10. Laurie, B., Langley, A.: Certificate transparency. http://www.certificate-transparency.org/ (2012), [last retrieved in April 2012]
11. Mozilla Security Blog: DigiNotar removal follow up. https://blog.mozilla.com/security/2011/09/02/diginotar-removal-follow-up/ (2011), [last retrieved in April 2012]
12. P. Eckersley and J. Burns: Is the SSLiverse a safe place? Talk at 27C3. https://www.eff.org/files/ccc2010.pdf (2010), [last retrieved in April 2012]
13. Qiu, S., McDaniel, P., Monrose, F.: Toward valley-free inter-domain routing. In: Proc. IEEE Int. Conf. on Communications (ICC). pp. 2009–2016 (Jun 2007)
14. Riedmaier, T., Holz, R.: Crossbear repository. https://github.com/crossbear/Crossbear, [last retrieved in April 2012]
15. Soghoian, C., Stamm, S.: Certified lies: Detecting and defeating government interception attacks against SSL. In: Proc. 15th. Int. Conf. Financial Cryptography and Data Security (FC'11) (Mar 2011)
16. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with Rocketfuel. In: Proc. ACM SIGCOMM. pp. 133–145. ACM, Pittsburgh, PA, USA (Aug 2002)
17. Sunshine, J., Egelman, S., Almuhimedi, H., Atri, N., Cranor, L.F.: Crying wolf: an empirical study of SSL warning effectiveness. In: Proc. 18th USENIX Security Symposium. pp. 399–416 (2009)
18. Teixeira, R., Shaikh, A., Griffin, T., Rexford, J.: Dynamics of hot-potato routing in IP networks. In: Proc. Joint Int. Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS). pp. 307–319. ACM, New York, NY, USA (2004)
19. Thoughtcrime Labs/IDS: Convergence. http://convergence.io (2011), [last retrieved in April 2012]
20. Vratonjic, N., Freudiger, J., Bindschaedler, V., Hubaux, J.P.: The inconvenient truth about Web certificates. In: 10th Workshop on Economics of Information Security (WEIS 2011) (June 2011)
21. Wendlandt, D., Andersen, D.G., Perrig, A.: Perspectives: Improving SSH-style host authentication with multi-path probing. In: Proc. USENIX 2008 Ann. Techn. Conf. (ATC) (2008)